Universidad de Cantabria
Departamento de Ingeniería
Informática y Electrónica

Minería de datos aplicada a la enseñanza virtual: nuevas propuestas para la construcción de modelos y su integración en un entorno amigable para el usuario no experto

Tesis Doctoral

Diego García Saiz

Dirigida por: Marta Elena Zorrilla Pantaleón

5 de abril de 2016

Universidad de Cantabria
Departamento de Ingeniería
Informática y Electrónica

Minería de datos aplicada a la enseñanza virtual: nuevas propuestas para la construcción de modelos y su integración en un entorno amigable para el usuario no experto

Memoria

Presentada para optar al grado de DOCTOR
por la Universidad de Cantabria
por

Diego García Saiz

"En el fondo, los científicos somos gente con suerte: podemos jugar a lo que queramos durante toda la vida."

Lee Smolin

Resumen

La implantación de la Web 2.0 ha permitido el diseño de distintas tecnologías en el campo educativo, como son las plataformas e-learning o los sistemas tutores inteligentes, que facilitan el desarrollo de la enseñanza a distancia. Actualmente, estos sistemas se utilizan de forma generalizada en el sistema educativo español y, sobre todo, en las universidades, con objeto tanto de impartir enseñanza completamente a distancia así como para complementar y dinamizar la enseñanza tradicional en las aulas. Gracias a que estos sistemas almacenan la actividad que los estudiantes realizan en ellos, se facilita la aplicación de técnicas de minería de datos para extraer patrones y modelos que ayuden a los profesores a dar respuestas a preguntas como, por ejemplo, "¿cuáles son los perfiles de los estudiantes?", "¿qué herramientas utilizan frecuentemente en sus sesiones de aprendizaje?", o "¿qué características determinan el rendimiento final de un estudiante?". Disponer de esta información permite a los profesores analizar cómo se lleva a cabo el proceso de enseñanza-aprendizaje y proponer acciones dirigidas a su mejora. Entre las muchas y variadas cuestiones que se plantean los educadores, una de las más estudiadas en la literatura es la aplicación de técnicas de clasificación para modelar y predecir el rendimiento de los estudiantes. Lamentablemente, la aplicación de estas técnicas no está al alcance de todo el profesorado, y es por tanto necesario proveer de herramientas a los profesores para que puedan extraer esa información sin necesidad de tener conocimientos sobre minería de datos. La herramienta *E-learning WebMiner*, desarrollada como Proyecto de Fin de Carrera del autor de la presente tesis, ofrecía la posibilidad de obtener modelos de minería de datos sobre cursos virtuales, pero no incluía la posibilidad de predecir el rendimiento de los estudiantes. La contribución de estas tesis, por tanto, se centra en dos aspectos: por un lado, se ofrecen modelos de predicción del rendimiento de los estudiantes en cursos virtuales atendiendo a distintas métricas y utilizando distintos procesos de minería para mejorar su calidad y precisión y, por otra parte, se propone un sistema que construye el mejor modelo de clasificación para un conjunto de datos dado sin interacción de un experto en minería de datos, esto es, se ofrece una alternativa para democratizar el uso de la minería de datos para usuarios no expertos. Para la consecución de estos objetivos, se han seguido varias líneas de trabajo cuyos estudios y conclusiones principales y más relevantes se incluyen en el presente documento: (1) revisión de la bibliografía relativa a la predicción con técnicas de minería de datos del rendimiento de los estudiantes en cursos, (2) estudio de aplicación de estas técnicas sobre nuevos conjuntos de datos, (3) aplicación de procesos de meta-learning para desarrollar un proceso de selección automática de clasificadores, (4) detección y eliminación de anomalías en los conjuntos de entrenamiento para mejorar los modelos de predicción, (5) aplicación de técnicas de análisis de redes sociales para la obtención de nuevos atributos predictores basados en la interacción social de los estudiantes, y

(6) extensión del diseño de la herramienta *E-learning WebMiner* para incluir, con los resultados obtenidos en las anteriores líneas de trabajo, la posibilidad de obtener modelos de predicción del rendimiento de los estudiantes.

Agradecimientos

No existe una sola palabra en el diccionario de la Real Academia Española con la que realmente pueda hacer honor a la magnitud del agradecimiento que siento ante todos los que me han apoyado y han estado a mi lado en el desarrollo de este trabajo de tesis, que bien parece llega a su justo final. Ha habido momentos buenos, muy buenos, y también momentos peores. Y en todos ellos habéis estado ahí, escogiendo acertadamente las palabras oportunas, y los silencios necesarios. Espero que estas limitadas pero sentidas palabras manuscritas reflejen al menos una pequeña porción de esta gratitud con todos vosotros.

El primer lugar de los agradecimientos es sin duda para mi directora de tesis, Marta Zorrilla, quien me tendió la mano y me dio la oportunidad de iniciarme en el mundo de la investigación, y quien ha guiado el desarrollo de este trabajo, guía sin la cual ahora mismo no estaría escribiendo estas líneas.

Otro agradecimiento que se impone por justicia es para Marta Pascual, durante todos estos años no ha habido persona que más me haya oído hablar sobre cualquier eventualidad en el desarrollo de la tesis.

Pero sin duda, quienes más merecen este agradecimiento son mis padres, Maria Luisa y José Aurelio, piedra angular en mi desarrollo como persona y, ahora, como investigador. Sin vosotros, nada de esto hubiese sido posible. Este trabajo es tan mío como vuestro, y espero que podáis sentir orgullo de lo que habéis ayudado a construir.

Índice general

Contenidos xi

Índice de Figuras xv

Índice de Tablas xix

Abreviaturas xxiii

1. Introducción 1
- 1.1. Motivación: la minería de datos en el área educativa . . . 3
- 1.2. E-learning WebMiner: minería de datos al alcance de todos los profesores de cursos virtuales . . . 4
- 1.3. Objetivos de la tesis y líneas de trabajo . . . 6
- 1.4. Modelo de procesos seguido para la aplicación de las técnicas de minería de datos . . . 11
- 1.5. Publicaciones sobre los contenidos de la tesis . . . 12
- 1.6. Organización del documento . . . 12

2. Descripción del contexto, fuentes de datos, procesos y técnicas utilizadas 17
- 2.1. Los cursos virtuales en la Universidad . . . 18
- 2.2. La actividad almacenada en Moodle y Blackboard . . . 19
- 2.3. Características de los cursos . . . 20
- 2.4. Extracción y pre-procesado de las medidas de actividad de los estudiantes 23
- 2.5. Técnicas utilizadas para la obtención y evaluación de los modelos de minería de datos . . . 29
- 2.6. Despliegue de los resultados: mejora y extensión de ElWM . . . 30

3. Educational Data Mining: estado del arte 31
- 3.1. La minería de datos aplicada al campo educativo . . . 31
- 3.2. Prediciendo el rendimiento de los estudiantes . . . 35
- 3.3. Conclusiones sobre el estado del arte . . . 39

4. Meta-learning: en busca del mejor clasificador 47
- 4.1. Estado del arte: meta-learning y la predicción de rendimiento . . . 48
- 4.2. Hipótesis de partida, organización y resumen de los estudios . . . 55
- 4.3. Meta-características de los conjuntos de datos utilizadas en los estudios . 58
- 4.4. Configuración, proceso, resultados y conclusiones de los estudios . . . 63

4.4.1. Estudio 1. Primeros pasos: comparativa de tipos de clasificadores para la predicción del rendimiento . . . 64
4.4.2. Estudio 2. Las meta-características simples como predictores . . . 67
4.4.3. Estudio 3. Construcción de recomendadores con las meta-características simples . . . 74
4.4.4. Estudio 4. Más allá de las meta-características simples: las meta-características de complejidad y de contexto . . . 79
4.4.5. Estudio 5. Nuevo enfoque del proceso de meta-learning: ranking con regresión . . . 83
4.5. Conclusiones y trabajo futuro . . . 96

5. Detección y eliminación de comportamientos anómalos para la mejora de los modelos de predicción 99
5.1. Estado del arte: detección de outliers y aplicación al entorno educativo . . 101
5.2. Visualizando los comportamientos anómalos . . . 103
5.3. Hipótesis de partida, organización y resumen de los estudios . . . 105
5.4. Configuración, proceso y resultados de los estudios . . . 108
5.4.1. Estudio 1. Aplicación de técnicas de detección de outliers para eliminar comportamientos anómalos . . . 110
5.4.2. Estudio 2. Aplicación de boosting y bagging para la mejora de los modelos de predicción . . . 116
5.4.3. Estudio 3. Ensamblado de clasificadores con Voto Mayoritario para detectar y eliminar comportamientos anómalos . . . 121
5.4.4. Estudio 4. DARIM: una nueva técnica de detección y eliminación de outliers . . . 130
5.4.5. Estudio 5. DARIM: otra aproximación basada en densidad . . . 150
5.4.6. Estudio 6. Detección de comportamientos anómalos para evitar el bajo rendimiento o el abandono . . . 153
5.5. Conclusiones y trabajo futuro . . . 162

6. Estudio y aplicación de otro tipo de técnicas para mejorar los modelos ofrecidos por ElWM 165
6.1. El poder predictivo del Análisis de Redes Sociales en la educación . . . 166
6.1.1. Estado del arte: Análisis de Redes Sociales en entornos e-learning . 167
6.1.2. Modelado de las redes sociales basadas en foros de cursos e-learning 170
6.1.3. Hipótesis de partida, organización y resumen de los estudios . . . 171
6.1.4. Medidas de SNA utilizadas en los estudios: las medidas centralidad 172
6.1.5. Configuración, proceso, resultados y conclusiones de los estudios . 173
6.1.5.1. Estudio 1. Redes sociales centralizadas en los foros de cursos *e-learning* . . . 173
6.1.5.2. Estudio 2. Redes sociales distribuidas en los foros de cursos *e-learning* . . . 179
6.1.6. Conclusiones y trabajo futuro . . . 190
6.2. Eliminando redundancia para facilitar la visualización de las reglas de asociación . . . 191
6.2.1. Comparativa de algoritmos de reglas de asociación . . . 193
6.2.2. Conclusiones y trabajo futuro . . . 201

7. Extensión de E-learning WebMiner: definición de nuevas plantillas 203
7.1. Estado del Arte: herramientas de minería de datos para usuarios no expertos en el campo educativo . . . 204
7.2. Arquitectura y nuevos servicios en ElWM . . . 206
7.3. Flujo de trabajo en ElWM: uso de los servicios . . . 207
7.4. Prediciendo el rendimiento de los estudiantes con ElWM: incorporación del proceso de meta-learning . . . 210
7.5. ¿Cómo ElWM muestra los modelos al usuario?: ejemplos . . . 210
7.6. Colaboraciones en curso . . . 213
7.6.1. E-learning WebMiner como Línea de Productos Software . . . 215
7.6.2. Extendiendo la funcionalidad de ElWM a otras áreas . . . 220

8. Conclusiones 223

9. Trabajo futuro 227

A. Anexos 231
A.1. Ejemplos de uso de ElWM previos al desarrollo de esta tesis . . . 231
A.2. Características de los cursos utilizados en la tesis . . . 234

Bibliografía 239

Índice de figuras

1.1. Arquitectura Orientada a Servicios Web de E-learning WebMiner 5

2.1. Relación del rendimiento de los estudiantes 21
2.2. Frecuencia de uso de las herramientas en los cursos 22

4.1. Recomendador J48 del estudio 2 72
4.2. Recomendador J48 para el conjunto md1 (Mejor accuracy) del estudio 4 . 82
4.3. Recomendador J48 para el conjunto md2 (Mejor TPrate) del estudio 4 . . 82
4.4. Propuesta de proceso de meta-learning con modelos de regresión 85
4.5. RMSE utilizando meta-características simples con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5 90
4.6. RMSE utilizando meta-características estadísticas con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5 90
4.7. RMSE utilizando landmarkers con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5 91
4.8. RMSE utilizando meta-características de complejidad con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5 91
4.9. RMSE utilizando todas las meta-características con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5 92
4.10. RMSE medio con (SA) y sin (no SA) selección de atributos en el estudio 5 92
4.11. Comparativa del RMSE utilizando todas las meta-características vs utilizando sólo landmakers con selección de atributos en el estudio 5 93

5.1. Distribución de los estudiantes de acuerdo al tiempo total y número de sesiones 104
5.2. Distribución de los estudiantes que suspendieron de acuerdo al tiempo total 104
5.3. Distribución de los estudiantes de acuerdo al tiempo total dedicado a los tests y número de tests realizados en los cursos con identificador del seis al once 105
5.4. Detección de comportamientos anómalos con LOF en el estudio 1 112
5.5. Variación de las mejoras obtenidas por ECODB al aumentar el % de instancias eliminadas con k=7 en el estudio 1 114
5.6. Porcentaje de veces que los algoritmos de boosting y bagging mejoran en términos de accuracy en el estudio 2 120
5.7. Variación de las mejoras obtenidas por el proceso de Voto Mayoritario al aumentar el umbral en el estudio 3 122
5.8. Porcentajes de mejora por medida aplicando el proceso Voto Mayoritario en el estudio 3 126
5.9. Mejoras por rango de accuracy inicial (Voto Mayoritario) en el estudio 3 . 130

5.10. Clusters C1 y C2 agrupando instancias de la clase positiva que han sido mal clasificadas . . . 133
5.11. Proceso seguido por DARIM . . . 134
5.12. Mejoras obtenidas por DARIM utilizando de 2 a 5 clústers (y eliminando N-1) en el estudio 4 . . . 135
5.13. Comparativa de mejora media entre DARIM y el proceso Voto Mayoritario en el estudio 4 . . . 137
5.14. Comparativa de mejora en la desviación estándar del proceso CV entre DARIM y el proceso Voto Mayoritario en el estudio 4 . . . 138
5.15. Comparativa de tiempos de ejecución entre DARIM y el proceso Voto Mayoritario en el estudio 4 . . . 138
5.16. Porcentajes de mejora por medida usando DARIM en el estudio 4 . . . 140
5.17. Comparativa de mejoras entre DARIM y el proceso Voto Mayoritario por conjunto en el estudio 4 . . . 141
5.18. Comparativa de mejoras mínimas alcanzadas por DARIM y por el proceso de Voto Mayoritario, por conjunto, en el estudio 4 . . . 142
5.19. Comparativa de mejoras máximas alcanzadas por DARIM y por el proceso de Voto Mayoritario, por conjunto, en el estudio 4 . . . 143
5.20. Comparativa de mejora por clasificador entre DARIM y Voto Mayoritario en el estudio 4 . . . 145
5.21. Mejoras por rango de accuracy inicial (DARIM) del estudio 4 . . . 146
5.22. Comparativa de mejoras entre DARIM y el proceso de Voto Mayoritario, según rango de accuracy inicial, en el estudio 4 . . . 147
5.23. Comportamientos anómalos detectados por ECODB . . . 149
5.24. Comportamientos anómalos detectados por DARIM . . . 150
5.25. Comparativa de mejoras obtenidas entre DARIM original y DARIM con DBScan . . . 154
5.26. Comparativa de las veces que mejoran DARIM original y DARIM con DBScan . . . 154
5.27. Estudiantes objetivo del estudio 6 . . . 156

6.1. Ejemplo de red social con las interacciones entre estudiantes de un foro en un curso e-learning . . . 171
6.2. Red centralizada obtenida en el curso 1 del estudio 1 . . . 177
6.3. Árbol de decisión J48 con el conjunto "mixed.dat", utilizando medidas SNA en el estudio 1 . . . 180
6.4. Red social del foro para el dataset1 del estudio 2 . . . 183
6.5. Red social del foro para el dataset2 del estudio 2 . . . 184
6.6. Árbol de decisión J48 para el dataset1, utilizando medidas SNA en el estudio 2 . . . 188
6.7. Árbol de decisión J48 para el dataset2, utilizando medidas SNA en el estudio 2 . . . 190

7.1. Arquitectura SOA de ElWM . . . 207
7.2. Flujo de trabajo de ElWM . . . 208
7.3. Información que el usuario puede obtener con ElWM . . . 208
7.4. Opciones que ElWM ofrece al usuario para generar modelos de predicción del rendmiento de los estudiantes . . . 209

7.5. Despliegue de los meta-modelos para la selección del clasificador 211
7.6. Predicción del rendimiento de los estudiantes mostrada al usuario por ElWM 212
7.7. Análisis de la interacción en el foro mostrado al usuario por ElWM 214
7.8. Proceso SPL 216
7.9. Costes *Software Product Lines* vs *Single-System Engineering* 217
7.10. Modelo de características 218
7.11. Reglas con restricciones al modelo de características 218
7.12. Ejemplo de arquitectura *Software Product Lines* 218
7.13. Plantilla de generación de código con Epsilon 219
7.14. Ejemplo de configuración 220
7.15. Arquitectura del servicio de minería propuesto 221

A.1. Resultados para la pregunta "¿Cuál es el perfil de estudiantes de mi curso?" 232
A.2. Resultados para la pregunta "¿Cuál es el perfil de sesiones que existen en mi curso?" 233
A.3. Resultados para la pregunta "¿Qué herramientas se suelen utilizar habitualmente juntas por los estudiantes en una sesión de aprendizaje?" . . . 234

Índice de tablas

1.1. Publicaciones surgidas durante el desarrollo de esta tesis 13

2.1. Resumen de las principales medidas generales de actividad 24
2.2. Resumen de las principales medidas de actividad en el foro 25
2.3. Resumen de las principales medidas de actividad en las herramientas Blog, Glosario y Wiki 25
2.4. Resumen de las principales medidas de actividad en las herramientas de Test, Correo y Entregables 26

3.1. Encuadramiento de los 4 grandes problemas en EDM en los que se aplican técnicas de clasificación 34
3.2. Trabajos relacionados en el área de la predicción del rendimiento de estudiantes en cursos (1) 40
3.3. Trabajos relacionados en el área de la predicción del rendimiento de estudiantes en cursos (2) 41
3.4. Trabajos relacionados en el área de la predicción del rendimiento de estudiantes en cursos (3) 42
3.5. Trabajos relacionados en el área de la predicción del rendimiento de estudiantes en cursos (4) 43

4.1. Trabajos relacionados sobre la selección de clasificadores con meta-learning (1) 53
4.2. Trabajos relacionados sobre la selección de clasificadores con meta-learning (2) 54
4.3. Meta-características simples utilizadas en los estudios 60
4.4. Meta-características estadísticas utilizadas en los estudios 61
4.5. Meta-características de complejidad utilizadas en los estudios 62
4.6. Meta-características landmarkers utilizadas en los estudios 63
4.7. Resultados obtenidos en los 3 conjuntos del estudio 1 66
4.8. Rango de las meta-caraterísticas de los conjuntos de datos utilizados en el estudio 2 68
4.9. Relevancia de las meta-características simples en el estudio 2 71
4.10. Caracterización de los conjuntos de datos del estudio 3 75
4.11. Descripción de los conjuntos de prueba del estudio 3 77
4.12. Ranking de los 5 clasificadores con un mayor accuracy para los conjuntos de prueba del estudio 3 77
4.13. Recomendaciones de los meta-modelos construidos con J48 y NaïveBayes para los conjuntos de prueba del estudio 3 78

4.14. Recomendaciones de clasificadores para los conjuntos de entrenamiento del estudio 3 . . . 79
4.15. Descripción de los conjuntos de datos del estudio 4 . . . 80
4.16. Características de los 4 conjuntos de prueba del estudio 4 . . . 81
4.17. Clasificador recomendado para cada conjunto de prueba del estudio 4 . . . 83
4.18. Rango de valores de las meta-características de complejidad en el estudio 5 86
4.19. Rango de valores de las características simples en el estudio 5 . . . 86
4.20. Rango de valores de las características estadísticas en el estudio 5 . . . 86
4.21. Rango de valores de los landmarkers en el estudio 5 . . . 86
4.22. RMSE de los meta-regresores del estudio 5 . . . 88
4.23. Número y porcentaje de veces que el mejor clasificador es recomendado por el sistema del estudio 5. Se incluye comparativa por cuartiles . . . 94
4.24. Ranking basado en el accuracy predicho (Acc. Pred.) y real (Acc. Real) para un conjunto de datos del estudio 5 . . . 95

5.1. Encuadramiento de los procesos y técnicas de clasificación en presencia de ruido, según el Survey de Frenay et al. [1] . . . 102
5.2. Mejoras obtenidas en los modelos de predicción al aplicar LOF, por ejecución, en el estudio 1 . . . 111
5.3. Mejoras obtenidas en los modelos de predicción al aplicar ECODB, por ejecución, en el estudio 1 . . . 113
5.4. Mejoras obtenidas en los modelos de predicción al aplicar ECODB con k=7 y 10 % de instancias eliminadas, por conjunto de datos, en el estudio 1 115
5.5. Estadísticas de las mejoras obtenidas en los modelos de predicción con RandomForest, por conjunto de datos, en el estudio 2 . . . 117
5.6. Estadísticas de las mejoras obtenidas en los modelos de predicción con MultiBoost, por conjunto de datos, en el estudio 2 . . . 118
5.7. Estadísticas de las mejoras obtenidas en los modelos de predicción con AdaBoost, por conjunto de datos, en el estudio 2 . . . 118
5.8. Mejoras obtenidas en los modelos de predicción con Bagging, por conjunto de datos, en el estudio 2 . . . 119
5.9. Mejoras obtenidas en los modelos de predicción al aplicar el proceso de Voto Mayoritario por ejecución en el estudio 3 . . . 123
5.10. Mejoras obtenidas en los modelos de predicción al aplicar el proceso de Voto Mayoritario, por conjunto de datos, en el estudio 3 . . . 125
5.11. Mejoras obtenidas en los modelos de predicción al aplicar el proceso de Voto Mayoritario, por clasificador, en el estudio 3 . . . 128
5.12. Mejoras obtenidas en los modelos de predicción al aplicar el proceso de Voto Mayoritario, por accuracy inicial, en el estudio 3 . . . 130
5.13. Mejoras obtenidas en los modelos de predicción al aplicar DARIM, por ejecución, en el estudio 4 . . . 136
5.14. Mejoras obtenidas en los modelos de predicción al aplicar DARIM, por conjunto de datos, en el estudio 4 . . . 139
5.15. Mejoras obtenidas en los modelos de predicción al aplicar DARIM, por clasificador, en el estudio 4 . . . 144
5.16. Mejoras obtenidas en los modelos de predicción al aplicar DARIM, por accuracy inicial, en el estudio 4 . . . 146

5.17. Mejoras obtenidas en los modelos de predicción al aplicar DARIM con DBSCAN, por ejecución (número mínimo de puntos = 1/10 del número de instancias), en el estudio 5 . . . 152
5.18. Mejoras obtenidas en los modelos de predicción al aplicar DARIM con DBSCAN, por conjunto de datos, en el estudio 5 . . . 153
5.19. Características principales de los dos conjuntos de datos utilizados, conteniendo la actividad de los estudiantes en el periodo comprendido entre el inicio del curso y la fecha del primer entregable para los cursos 23 y 26 157
5.20. Proceso de clustering y peso de los atributos en el conjunto "d1"del estudio 6 . . . 160
5.21. Proceso de clustering y peso de los atributos en el conjunto "d2"del estudio 6 . . . 160
5.22. Actividad de los estudiantes y calificaciones en los 2 primeros entregables de los cursos ("q1" y "q2" respectivamente) del estudio 6 . . . 162

6.1. Medidas de centralidad . . . 174
6.2. Valor de cada medida de centralidad para los 3 nodos con mayor valor del estudio 1 . . . 177
6.3. Accuracy obtenido por cada dataset y clasificador, utilizando sólo como predictoras las medidas SNA en el estudio 1 . . . 179
6.4. Accuracy obtenido con el dataset "mixed.dat" utilizando sólo medidas de actividad ("No SNA") y añadiendo las medidas de centralidad ("SNA") en el estudio 1 . . . 179
6.5. Top 5 de los estudiantes con los valores más altos en cada medida de centralidad en el estudio 2 . . . 183
6.6. Atributos seleccionados con la técnica CfsSubSetEval para el "curso 4"del estudio 2 . . . 185
6.7. Atributos seleccionados con la técnica ClassifierSubSetEval para el dataset1, utilizando NaïveBayes como clasificador base en el estudio 2 . . . 186
6.8. Atributos seleccionados con la técnica ClassifierSubSetEval para el dataset1, utilizando J48 como clasificador base en el estudio 2 . . . 186
6.9. Accuracy, TPrate y TNrate obtenidos con J48 y NaïveBayes en el dataset1 del estudio 2 . . . 187
6.10. Atributos seleccionados con la técnica CfsSubSetEval para el dataset2 del estudio 2 . . . 188
6.11. Atributos seleccionados con la técnica ClassifierSubSetEval para el dataset2, utilizando NaïveBayes como clasificador base en el estudio 2 . . . 189
6.12. Atributos seleccionados con la técnica ClassifierSubSetEval para el dataset2, utilizando J48 como clasificador base en el estudio 2 . . . 189
6.13. Accuracy, TPrate y TNrate obtenidos con J48 y NaïveBayes en el dataset2 del estudio 2 . . . 190
6.14. Ejemplo de rendundancia entre reglas . . . 192
6.15. Descripción de los conjuntos de datos . . . 193
6.16. Ejemplo de instancias en el dataset2 . . . 194
6.17. Número de reglas generadas al aplicar cada algoritmo sobre los datasets . 195
6.18. Subconjunto de reglas con confianza de 100 % obtenidas al aplicar Apriori de Weka sobre el dataset3 . . . 196
6.19. Subconjunto de reglas con confianza menor de 100 % obtenidas al aplicar Apriori de Weka sobre el dataset3 . . . 196

6.20. Subconjunto de reglas obtenidas al aplicar PredictiveApriori sobre el dataset3 . . . 196
6.21. Subconjunto de reglas obtenidas al aplicar Apriori de Borgelt sobre el dataset3 . . . 197
6.22. Subconjunto de reglas obtenidas al aplicar ChARM sobre el dataset3 . . . 197
6.23. Subconjunto de reglas obtenidas al aplicar ChARM sobre el dataset5 . . . 197
6.24. Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset3 . . . 198
6.25. Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset5 . . . 199
6.26. Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset2 . . . 200
6.27. Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset4 . . . 200
6.28. Subconjunto de reglas obtenidas al aplicar ChARM sobre el dataset1 . . . 200
6.29. Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset1 . . . 201

A.1. Características de los cursos . . . 235
A.2. Herramientas más frecuentemente utilizadas en los cursos . . . 237

Abreviaturas

CRISP-DM	**C**ross **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining
ElWM	**E**-l**e**arning **W**eb **M**iner
KDD	**K**nowledge **D**iscovery on **D**atabases
SNA	**S**ocial **N**etwork **A**nalisys
10-CV	10-folds **C**ross **V**alidation
EDM	**E**ducational **D**ata **M**ining
MOOC	**M**assive **O**pen **O**nline **C**ourse
LCMS	**L**earning **C**ontent **M**anagement **S**ystem
TPrate	**T**rue **P**ositive rate
TNrate	**T**rue **N**egative rate
FPrate	**F**alse **P**ositive rate
FNrate	**F**alse **N**egative rate
SPL	**S**oftware **P**roduct **L**ine
AUC	**A**rea **U**nder the **C**urve
BI	**B**usiness **I**ntelligence
SOA	**S**ervice **O**riented **A**rchitecture
WSDL	**W**eb **S**ervices **D**escription **L**anguage

Capítulo 1

Introducción

Las actuales técnicas y herramientas de almacenamiento y gestión de los datos han permitido a instituciones, empresas y organizaciones la acumulación de grandes cantidades de datos, relativos a sus procesos de negocio [2]. Se pueden citar, como ejemplos prácticos, a las grandes superficies, que guardan en sus sistemas miles, o incluso millones, de transacciones a la semana relativas a las compras de los clientes; a las empresas de telecomunicaciones, que almacenan cientos de petabytes con datos acerca del tráfico en sus redes; a las redes sociales, que soportan millones de interacciones entre usuarios al día; a los bancos, que generan diariamente gran cantidad de datos sobre transacciones monetarias; y, a otra escala más baja, pero también muy importante, tenemos ejemplos como los censos que almacenan las administraciones públicas; o los cursos virtuales impartidos en plataformas MOOC o en plataformas internas de las instituciones educativas, que almacenan gran cantidad de datos relacionados con los contenidos impartidos, con las actividades que desarrollan los estudiantes en dichas plataformas, y con la interacción que surge entre los diferentes actores del curso, como son los estudiantes y los profesores.

Una de las ventajas de almacenar estos datos es que resultan ser muy útiles de cara a extraer información que ayude a la toma de decisiones dentro de la propia organización, con objeto de mejorar sus procesos [3]. Dado que el volumen de datos que se maneja es ingente, resulta imposible para un humano la extracción de información y patrones ocultos contenidos en los datos, si no es mediante el uso de técnicas que permitan obtener y sintetizar esta información de forma que sea humanamente interpretable y accionable, es

decir, que permita tomar acciones en consecuencia. Sin este tipo de técnicas, por ejemplo, una empresa de venta de productos podría conocer, observando los datos, cuánto han vendido en un periodo o cuál es la tendencia de venta, pero difícilmente podrían entender el por qué. Estas técnicas se engloban en el área que actualmente se conoce como Inteligencia de Negocio (*Business Intelligence* en inglés, BI por sus siglas) [4], técnicas que en los últimos años vienen siendo clave como parte del proceso *Big Data* [5]. Y entre los tipos de técnicas existentes para la extracción de la información, están las técnicas de lo que se conoce como **Minería de Datos** (*Data Mining* en inglés). De acuerdo a Witten et al. [6], *"Data mining is the extraction of implicity, previously unknown, and potentially useful information from data"*.

Actualmente, y con este objetivo, las técnicas de minería de datos están siendo aplicadas, de forma muy notable, en diversas áreas, como pueden ser el marketing [7], las finanzas [8], y la medicina [9], entre muchas otras. Y una de las áreas en las que, en los últimos años, la minería de datos está teniendo un gran impacto y desarrollo, es en el área educativa [10].

Lamentablemente, el uso de estas técnicas no está al alcance de todo el mundo. El proceso de extracción de conocimiento (*Knowledge Discovery in Databases* [3] en inglés, KDD por sus siglas) consta de varias fases, que requieren tomar decisiones cuya ejecución no está al alcance de los usuarios no expertos en este área. Más aún, debido a la complejidad de las propias técnicas de minería de datos, su uso requiere de expertos en la materia capaces de tomar las decisiones pertinentes acerca de, por ejemplo, qué técnica o conjunto de técnicas se adaptan mejor a los datos sobre los que se está trabajando, la configuración y parametrización de estas técnicas, o la interpretación de los modelos extraídos.

En los últimos años han surgido algunos trabajos [11] cuyo objetivo es desarrollar y adaptar el proceso de minería de datos de forma que usuarios no expertos, sin conocimientos en el área, puedan obtener, por ellos mismos, los patrones y modelos que ofrecen estas técnicas. De esta forma, al usuario sólo se le requiere que exprese los requisitos sobre la información que se desea obtener. Por supuesto, en este proceso también se ha de prestar atención al hecho de que los modelos y patrones retornados al usuario han de ser sencillos de interpretar.

1.1. Motivación: la minería de datos en el área educativa

Las técnicas de minería de datos son ampliamente utilizadas en el área de la educación, una tendencia que ha ido creciendo hasta la actualidad [10, 12]. La aplicación de estas técnicas sobre datos del entorno educativo ha adquirido aún mayor relevancia gracias a la aparición de los entornos de aprendizaje virtual, como Moodle [13] o Blackboard [14], cuyo uso está muy extendido en las instituciones de ámbito educativo, y especialmente en el nivel superior de educación, tanto para complementar con nuevos contenidos y actividades las clases presenciales, conocido como educación semi-presencial o *blended learning*, como para impartir cursos íntegramente virtuales, lo que comúnmente se denomina *e-learning*. Para facilitar la lectura, a partir de ahora en el texto se hará referencia a ambos tipos de cursos como cursos virtuales.

La falta de contacto existente entre profesores y estudiantes en cursos virtuales, entre muchos otros problemas, puede repercutir negativamente en el proceso de aprendizaje de estos últimos, y por tanto también en su rendimiento. Es por ello que existe la necesidad de ofrecer realimentación, tanto a profesores como estudiantes, sobre la actividad realizada en los cursos impartidos en estas plataformas, con objeto de extraer información que ayude a comprender como desarrollan su labor los estudiantes y poder así mejorar el proceso de enseñanza-aprendizaje. Esta información puede ser extraída y ofrecida mediante el uso de técnicas de minería de datos [15].

En trabajos como [10] y [12] se pueden constatar las diferentes utilidades y objetivos con los que se puede utilizar la minería de datos en el entorno educativo: entre otros, y por ejemplo, descubrir los diferentes perfiles de comportamiento y actividad que existen en los estudiantes, o caracterizar a los estudiantes con alto riesgo de abandono o con bajo rendimiento en las pruebas evaluables.

Gracias a que los mencionados entornos de aprendizaje virtual tienen bases de datos en las que almacenan toda la actividad que los estudiantes desarrollan en los cursos virtuales, es posible modelar el comportamiento de estos estudiantes mediante aplicación de técnicas de minería de datos. Con estos modelos, los profesores pueden tener una mejor idea de cómo adaptar sus cursos para favorecer el rendimiento, pudiendo detectar, entre muchos otros, problemas como la falta de motivación, el bajo rendimiento, o el poco uso de los recursos más importantes para superar la asignatura [15].

En la actualidad, existe una disciplina específica para la aplicación de algoritmos y técnicas de minería de datos en el área educativa, conocida, en inglés, como *Educational Data Mining* (EDM) [16]. Precisamente, uno de los problemas abiertos en esta disciplina es el desarrollo y aplicación de técnicas de minería de datos sobre datos de cursos impartidos en plataformas *e-learning*, esto es, sobre cursos virtuales. En ese área de la disciplina EDM, uno de los retos que se plantean es el acercamiento de estas técnicas a usuarios no expertos, como son, por ejemplo, profesores de cursos virtuales ajenos al campo de la minería de datos, de forma que puedan obtener la respuesta dada por estas técnicas sobre la actividad de sus estudiantes sin que sea necesaria la intermediación de un experto en minería de datos [17]. Por ello, y con la intención de que, por ejemplo, profesores de áreas como derecho o económicas, puedan conocer y extraer sobre sus cursos estos modelos que les ayuden a mejorar el proceso de enseñanza-aprendizaje, es necesario ofrecer herramientas y técnicas de minería de datos que no requieran de una configuración compleja ni de un entendimiento profundo de su implementación y que, por supuesto, les muestren los patrones y modelos que contienen la respuesta que desean de forma que sea sencilla de entender. Aunque en la literatura ya existen propuestas de herramientas en este sentido, tanto en entornos educativos [18–20] como en otras áreas [21, 22], la mayoría no están orientadas al usuario no experto. Con objeto de dar respuesta a esta necesidad, el autor de esta tesis, en su trabajo de Fin de Carrera en Ingeniería Informática, diseñó e implementó la aplicación E-learning WebMiner [23].

1.2. E-learning WebMiner: minería de datos al alcance de todos los profesores de cursos virtuales

E-learning WebMiner (ElWM) es una aplicación que, siguiendo una Arquitectura Orientada a Servicios Web [24] (*Service-Oriented Architecture*, SOA por sus siglas, ver Figura 1.1), ofrece a los profesores de cursos virtuales la posibilidad de extraer modelos de minería de datos sobre sus cursos a partir de la actividad realizada por sus estudiantes, de forma sencilla y sin necesidad de tener conocimientos sobre las técnicas que se utilizan, esto es, ocultándolas al usuario.

Para ello, ElWM tiene implementadas una serie de plantillas, cada una de las cuáles responde a una pregunta concreta que el profesor puede hacer sobre su curso. Estas

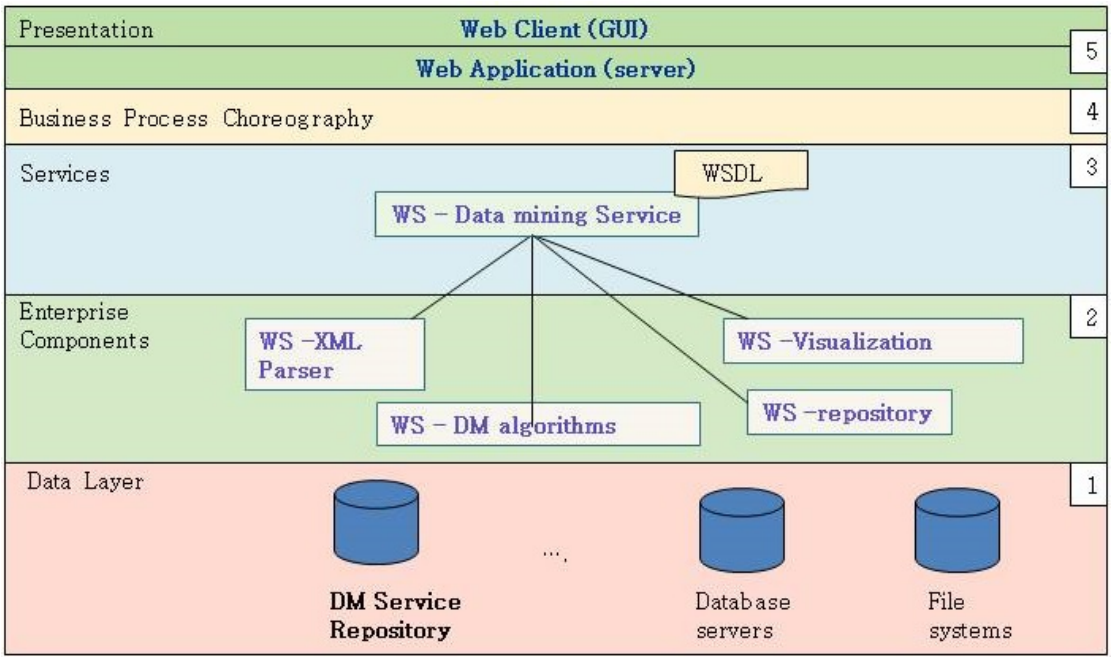


FIGURA 1.1: Arquitectura Orientada a Servicios Web de E-learning WebMiner

plantillas configuran un proceso secuencial, en el cuál se hace uso de una técnica predefinida para dar respuesta a cada pregunta. El profesor, por tanto, sólo ha de proveer el fichero con los datos de actividad de los estudiantes de su curso y seleccionar la pregunta sobre la que desea respuesta. De forma adicional, en vez de proveer el fichero con los datos, el profesor puede optar por requerir que ElWM genere automáticamente el conjunto de datos con la actividad de los estudiantes accediendo a la base de datos del LCMS en donde se hospeda su curso.

Una vez seleccionada la pregunta, el profesor tan solo ha de pulsar en el botón "Ejecutar", y ElWM se encargará de seguir el proceso definido por la plantilla para extraer y mostrar el modelo de minería de datos. En el momento en el que se inicia esta tesis, ElWM tiene definidas tres plantillas diferentes, que responden a tres preguntas que el profesor puede hacer sobre su curso: "¿Cuál es el perfil de estudiantes de mi curso?", "¿Cuál es el perfil de sesiones que existen en mi curso?" y "¿Qué herramientas se suelen utilizar habitualmente juntas por los estudiantes en una sesión de trabajo?". Las dos primeras preguntas utilizan algoritmos de agrupamiento o *clustering* para mostrar la información al usuario, mientras que la tercera hace uso de técnicas de reglas de asociación. En el anexo A.1 se muestran ejemplos de cómo ElWM muestra al usuario las respuestas a algunas de estas preguntas.

En base a posteriores estudios en los que se contó con la realimentación y opiniones de varios profesores de cursos virtuales, se puede afirmar que ElWM mostró ser una herramienta muy útil de cara a ofrecer a estos profesores modelos de minería de datos que les ayudasen a mejorar el proceso de enseñanza y aprendizaje [25].

No obstante, en el momento de inicio de esta tesis, la herramienta presentaba algunas carencias que debían de ser corregidas. La primera de ellas tiene que ver con los algoritmos utilizados para responder a cada pregunta. Aunque estos algoritmos habían sido seleccionados en base a una exhaustiva experimentación previa, hecho que garantiza que la información mostrada al profesor sea de calidad y fácilmente interpretable, no era siempre el mismo algoritmo el que podía conseguir el mejor modelo para cada caso particular, como así postula el *no free lunch theorem* [26]. Dependiendo de las características de los datos o del contexto del que provienen, los algoritmos que retornen los mejores modelos pueden variar. En segundo lugar, ElWM no da respuesta a uno de los problemas más ampliamente estudiados y que más preocupan a la comunidad EDM: la predicción del rendimiento de los estudiantes, esto es, la generación de modelos predictivos que, utilizando técnicas de minería supervisadas, nos informen de la relación que existe entre la actividad desarrollada por los estudiantes y su rendimiento en el curso.

1.3. Objetivos de la tesis y líneas de trabajo

La investigación desarrollada en esta tesis se ha orientado al estudio, desarrollo y aplicación de técnicas que permitan obtener modelos de predicción del rendimiento de los estudiantes de cursos virtuales impartidos en niveles de educación universitaria, con la intención final de que los profesores, sin conocimientos en minería de datos, puedan obtener y utilizar estos modelos para mejorar el proceso de enseñanza y aprendizaje, lo que a su vez redundará en una mejora del rendimiento de sus estudiantes. Existen, por tanto, dos objetivos específicos en esta tesis:

- **Objetivo 1:** Aplicación del proceso de minería de datos, con especial atención en las técnicas de clasificación, con objeto de obtener patrones precisos y fiables que modelen y predigan el rendimiento de los estudiantes en relación a la actividad que éstos desarrollan en cursos virtuales de educación universitaria, utilizando técnicas ya existentes y proponiendo otras nuevas que mejoren los modelos obtenidos.

- **Objetivo 2:** Desarrollo de procesos de automatización y propuesta de diseño de una herramienta software, partiendo de la experiencia de ElWM, que permitan a un profesor de cursos virtuales sin conocimientos sobre minería de datos obtener, interpretar y tomar decisiones en base a los modelos del Objetivo 1, sin necesidad de la intervención de un experto en el área.

Para la consecución de estos objetivos, se estableció un plan de actuación con diferentes líneas de trabajo, dando lugar a las distintas aportaciones de esta tesis que se exponen a continuación:

I. Aplicación y estudio de las técnicas de clasificación sobre datos provenientes del entorno educativo

Dependiendo de la fuente de los datos, la calidad de los modelos obtenidos por las técnicas de minería datos puede variar enormemente, siendo unas técnicas más útiles que otras según el propio contexto del que los datos utilizados provengan. Por ejemplo, no es lo mismo realizar un estudio sobre los datos de actividad de los estudiantes de un curso de filología en el que sólo se puedan matricular estudiantes con un perfil académico cercano a los contenidos del curso, que realizar el mismo estudio sobre datos de actividad provenientes de un curso de carácter transversal, en el que puedan matricularse estudiantes de muy diversos perfiles académicos. En este ejemplo, la aplicación de una misma técnica de minería de datos, con la misma configuración en ambos casos, podría dar resultados muy diferentes, y probablemente además estos resultados serían más útiles e informativos en un caso que en el otro.

Esta línea de trabajo, por tanto, tuvo por objeto estudiar la utilidad de las distintas técnicas de clasificación más frecuentemente usadas en la actualidad, con la intención final de determinar cuáles eran las más útiles según el contexto, la información requerida por el usuario, y la propia idiosincrasia y características de los cursos.

La presente línea de trabajo se compuso de dos tareas. Por un lado, la búsqueda y estudio de trabajos publicados en el área de la predicción del rendimiento de los estudiantes para conocer las conclusiones de otros autores, cuyos resumen y discusión se recogen en el apartado 3.2 del capítulo 3. Por otro lado, la aplicación, estudio y comparativa de algoritmos de clasificación sobre los conjuntos de datos disponibles para la realización de esta tesis. En general, los resultados de esta segunda tarea no se exponen extensamente

en este documento de tesis, ya que no son un fin en sí mismos, sino el punto de partida sobre el que se desarrollan las siguientes tareas. Para comprender precisamente este punto de partida, un ejemplo de los resultados obtenidos al comparar diferentes técnicas sobre conjuntos de datos de diferentes características se muestra en el apartado 4.4.1 del capítulo 4.

II. Selección automática de las técnicas de clasificación para la predicción del rendimiento de los estudiantes

Dada la premisa de que los profesores de cursos virtuales no tienen, por norma, conocimientos sobre minería de datos, puede ser sumamente complejo, y en ocasiones casi imposible, que este tipo de usuarios puedan hacer uso y aplicar, por sí mismos, las técnicas de clasificación necesarias para obtener un modelo de predicción sobre sus estudiantes. Es por esto que, de cara a cumplir con los objetivos de esta tesis, era imperativo desarrollar un sistema que ofreciese estos modelos sin necesidad de que los profesores tengan que intervenir en el proceso de selección y uso de los clasificadores. Para ello, se estableció la necesidad de desarrollar un recomendador de algoritmos que determinase, a partir de las características de los datos, cuál era el clasificador más adecuado para construir el modelo de predicción del rendimiento de los estudiantes. También se determinó que los clasificadores utilizados y recomendados por este sistema habían de retornar modelos de predicción sencillos de interpretar, con objeto de que el profesor pueda, por sí mismo, extraer conclusiones.

Los estudios y resultados más relevantes así como las conclusiones de esta línea de trabajo se recogen en el capítulo 4.

III. Mejora de los modelos de clasificación que predicen el rendimiento de los estudiantes

Con el desarrollo de esta línea de trabajo lo que se pretendió fue ofrecer a los profesores modelos de predicción con la mayor calidad posible. Esta línea se abordó en dos tareas diferentes, que se exponen a continuación:

III.1 Diseño, implementación y aplicación de técnicas que ayuden a mejorar la calidad de los modelos de clasificación

La idea en esta tarea fue desarrollar algoritmos o técnicas de minería de datos que sirviesen para mejorar la calidad de los patrones y modelos de predicción que pueden

extraerse con las técnicas de clasificación actualmente existentes. En este sentido, los esfuerzos se centraron en la detección y tratamiento de comportamientos irregulares entre los estudiantes de cursos virtuales que provocasen un empeoramiento en la calidad de los modelos de predicción. Como consecución de esta tarea, se implementó una nueva técnica que, basándose en la detección de estos comportamientos irregulares, fue capaz de mejorar la precisión y calidad de los modelos extraídos con las técnicas de clasificación.

Los estudios y resultados más relevantes así como las conclusiones de esta tarea se recogen en el capítulo 5.

III.2. Generación y estudio de nuevos atributos predictores que modelen la actividad de los estudiantes

Aunque las propias plataformas *e-learning* almacenan gran cantidad de datos sobre la actividad desarrollada por los estudiantes de cursos virtuales, como pueden ser las visitas realizadas a una determinada sección del curso o los mensajes enviados a un foro, hay datos que, pudiendo ser útiles para el proceso de extracción del conocimiento, no se encuentran directamente almacenados en sus sistemas, pero pueden ser inferidos de los que sí se almacenan. En este tarea se buscó la obtención de nuevos atributos predictores que puedan ayudar a mejorar los modelos de clasificación del rendimiento de los estudiantes, y que no puedan ser obtenidos mediante lectura directa de las tablas de *log* de los sistemas LCMS.

Para su consecución, se aplicaron técnicas de Análisis de Redes Sociales (*Social Network Analysis*, SNA por sus siglas) para analizar la interacción de los estudiantes en los foros de cursos virtuales, extrayendo así atributos que caracterizasen el comportamiento social de los estudiantes y que pudiesen ser utilizados para predecir su rendimiento.

Los estudios y resultados más relevantes así como las conclusiones de esta tarea se recogen en el apartado 6.1.

IV. Ampliación de E-learning WebMiner con nuevas plantillas que permitan a los profesores obtener modelos de predicción del rendimiento

En base a los resultados obtenidos en las líneas de trabajo I, II y III, y con objeto de que los profesores puedan obtener modelos de predicción del rendimiento de sus estudiantes, se desarrolló una extensión del diseño de la herramienta ElWM con nuevas plantillas que ofreciesen esta posibilidad. Para ello, se incluyeron en el nuevo diseño de ElWM

el sistema de selección automática de clasificadores descrito en la tarea II, así como la técnica de mejora de modelos de predicción de la tarea III.1, y la posibilidad de obtener modelos de predicción basados en la interacción social de los estudiantes, tal y como se describe en la tarea III.2. Igualmente, se realizó un diseño mejorado de ElWM, haciendo uso de nuevas tecnologías que fueron surgiendo en el periodo de desarrollo de esta tesis.

Este nuevo diseño de ElWM se incluye en el capítulo 7.

V. Otras líneas de actuación desarrolladas en la tesis

Además de las líneas descritas en el anterior apartado, y fruto de la investigación realizada a lo largo de las diferentes fases de la tesis, se han desarrollado otras líneas de trabajo de menor entidad pero que se relacionan con los objetivos de esta tesis, y que se describen a continuación:

V.A. Mejora de las plantillas que utilizan reglas de asociación

ElWM, tal y como se describe en el anexo A.1, utiliza el algoritmo *Apriori* para responder a las preguntas que necesitan de algoritmos de reglas de asociación. No obstante, este algoritmo genera, en ocasiones, una cantidad ingente de reglas redundantes, lo que convierte sus modelos en inmanejables para cualquier persona, y más aún para un usuario no experto. En el apartado 6.2 se incluyen los resultados de un estudio cuyo aporte consistió en la mejora de esta plantilla de ElWM, utilizando técnicas que eliminasen patrones redundantes y redujesen el número de reglas mostradas al usuario, de forma que el modelo obtenido fuese fácilmente interpretable y accionable por cualquier profesor de cursos virtuales.

V.B. Detección de estudiantes en riesgo de abandono

El proceso de detección de comportamientos anómalos propuesto en la tarea II.A mostró ser útil no solamente para mejorar los modelos de predicción, sino para ayudar a la detección de estudiantes en riesgo de suspender o incluso abandonar un curso antes de que suceda, alertando al profesor para que pueda actuar en consecuencia. El estudio relacionado con esta aportación se incluye en el apartado 5.4.6 del capítulo 5.

V.C. Exploración de otras posibilidades de diseño de ElWM como Línea de Productos Software

ElWM está diseñado como un conjunto de servicios web independientes, orquestados por una aplicación web. Este tipo de arquitectura es adecuada cuando se tiene como finalidad construir un solo producto software. No obstante, se estableció la necesidad de modelizar y ofrecer variantes del producto software para satisfacer las demandas únicas de cada potencial cliente o usuario de la herramienta. En el apartado 7.6.1 se explora la posibilidad de diseñar ElWM como Línea de Productos Software (*Software Product Line* en inglés, SPL por sus siglas) [27].

V.D. Extensión del estudio de minería de datos para usuarios no expertos a otros campos diferentes del educativo

El campo educativo no es el único en el que se pueden encontrar usuarios no expertos que necesiten de herramientas que les muestren modelos de minería de datos fácilmente interpretables, y sin que ellos tengan que intervenir en la configuración o selección de estos algoritmos. En el desarrollo de esta tarea se propuso una primera aproximación al diseño y desarrollo de un sistema, siguiendo la idea de ElWM, que ofrezca esta posibilidad con datos provenientes de cualquier área, negocio u organización. Esta propuesta está desarrollada en el apartado 7.6.2 del capítulo 7.

1.4. Modelo de procesos seguido para la aplicación de las técnicas de minería de datos

El proceso de minería de datos no consiste, simplemente, en la aplicación de algoritmos de minería de datos. Antes de ello, e incluso después, es necesario aplicar una serie de tareas que garanticen que los modelos y patrones obtenidos sean fiables y muestren con precisión la información oculta en los datos y requerida por el usuario. Estas acciones pueden ir desde la necesidad, por parte del usuario, de entender los datos con los que se está trabajando, hasta la evaluación rigurosa del propio modelo obtenido, tratando de garantizar su calidad.

Con objeto de seguir un orden que garantice la correcta consecución de la experimentación en minería de datos, existen diversos modelos de procesos que definen las tareas a seguir para construir y evaluar modelos, así como realizar su adecuado despliegue. En los experimentos de minería de datos llevados a cabo en esta tesis, se ha utilizado uno

de estos modelos de procesos cuyo uso está más extendido [28], CRISP-DM (siglas de *Cross Industry Standard Process for Data Mining*) [29].

1.5. Publicaciones sobre los contenidos de la tesis

El trabajo contenido en esta tesis ha dado lugar a la publicación y presentación de 20 artículos de congresos nacionales e internacionales y extensiones de los mismos, 2 capítulos de libro y 3 revistas, cuya relación puede encontrarse en la Tabla 1.1. La cuarta columna de esta tabla indica en qué líneas de trabajo, de las expuestas en el apartado 1.3, se encuadran estas publicaciones. En la quinta columna, se indica el medio de publicación: congresos y versiones extendidas de los mismos (C), revistas (R) y capítulos de libro (L).

De los 20 artículos de congreso, 6 de ellos se han publicado en el congreso de referencia en el campo del EDM, el *International Conference of Educational Data Mining*, durante todas sus ediciones entre 2011 y 2015. También se han publicado artículos en otros congresos con temáticas afines a la investigación de esta tesis, como son el *Computational Collective Intelligence* o el *International Symposium on Computers in Education*, así como en congresos de corte más generalista como el *International Conference of Format Concept Analysis*, el *International Conference on Rough Sets and Intelligent Systems Paradigms* o el *Simposio de Teoría y Aplicaciones de Minería de Datos*. En cuanto a las revistas, uno de los artículos fue publicado en *Decision Support Systems*, con JCR del primer cuartil. Los otros dos artículos han sido publicados en revistas indexadas en *Scopus* y que actualmente también se encuentran indexadas en el *Emerging Source Citation Index*, e incorporadas por tanto a la *Web of Science* en espera de obtener índice JCR próximamente.

1.6. Organización del documento

El presente documento de tesis se organiza de la siguiente forma:

En el capítulo 2 se expone el proceso seguido en esta tesis, incluyendo una descripción resumida del contexto en el que se desarrollan los estudios, de las características de los

Tabla 1.1: Publicaciones surgidas durante el desarrollo de esta tesis

Ref.	Título	Año	Tarea	Medio
[30]	Comparing classification methods for predicting distance students' performance	2011	I,II	C
[31]	E-learning Web Miner: A data mining application to help profesors involved in virtual courses	2011	IV	C
[32]	Towards Parameter-free Data Mining: Mining educational data with yacaree	2011	V.A	C
[25]	A Data Mining Service to Assist Instructors Involved in Virtual Education	2011	IV	L
[33]	Iterator-based Algorithms in Self-Tuning Discovery of Partial Implications	2012	V.A	C
[34]	Closures and Partial Implications in Educational Data Mining	2012	V.A	C
[35]	A promising classification method for predicting distance students' performance	2012	III.1	C
[36]	Software Product Line Engineering for e-Learning Applications: A Case Study	2012	IV	C
[37]	Towards the Development of a Knowledge Base for Realizing User-Friendly Data Mining	2012	II	C
[38]	Towards the development of a classification service for predicting students' performance	2013	III.1	C
[39]	Development of a Knowledge Base for Enabling Non-expert Users to Apply Data Mining Algorithms	2013	II	C
[40]	Social Network Analysis and Data Mining: An Application to the E-Learning Context	2013	III.2	C
[41]	A service oriented architecture to provide data mining services for non-expert data miners	2013	III.2,IV	R
[42]	Data Mining and Social Network Analysis in the Educational Field: An Application for Non-Expert Users	2013	IV	C
[43]	Building Families of Software Products for e-Learning Platforms: A Case Study	2014	V.B	R
[44]	Meta-learning: Can It Be Suitable to Automatise the KDD Process for the Educational Domain?	2014	II	C
[45]	The predictive power of the SNA metrics for education	2014	III.2	L
[46]	Domain Specific Languages for Data Mining: A Case Study for Educational Data Mining	2015	V.C	C
[47]	Towards a DSL for Educational Data Mining	2015	V.C	C
[48]	Detection of Learners with a Performance Inconsistent with Their Effort	2015	III.2	C
[49]	Meta-Learning Based Framework for Helping Non-expert Miners to Choice a Suitable Classification	2015	II	C
[50]	Un marco para democratizar la minería de datos: propuesta inicial y retos	2015	II, IV, V.D	C
[51]	A parametrisable method for measuring online attendance in e-learning tools	2015	I	R
[52]	Enabling Non-expert Users to Apply Data Mining for Bridging the Big Data Divide	2015	II, V.D	C
[53]	Metalearning-based recommenders: towards automatic classification algorithm selection	2015	II	C

cursos y de los conjuntos de datos, así como de los procesos y técnicas de minería de datos utilizadas en esta tesis.

En el capítulo 3 se expone un resumen de los trabajos más relevantes en el área del EDM, entrando en detalle en los estudios de aplicación de técnicas de clasificación para la predicción del rendimiento de los estudiantes.

Los capítulos del 4 al 7 contienen los estudios y conclusiones de las aportaciones y líneas de trabajo desarrolladas en esta tesis, mencionadas en el apartado 1.3:

- Capítulo 4: estudio correspondiente a la línea de trabajo II de los objetivos de esta tesis, en el que se aplican, estudian y proponen procesos de selección automática de clasificadores. También contiene un primer estudio realizado para constatar el diferente comportamiento que tienen los clasificadores al predecir el rendimiento de los estudiantes con cada conjunto de datos (línea de trabajo I).
- Capítulo 5: se corresponde con la tarea III.1 de la línea de trabajo III, en la cual se utilizan y proponen técnicas de detección y tratamiento de comportamientos anómalos por parte de los estudiantes de cursos virtuales.
- Capítulo 6: se divide en dos grandes apartados, 6.1 y 6.2, cada uno de los cuales contiene respectivamente las aportaciones de la tarea III.2, dentro de la línea de trabajo III, y relativa a la mejora de los modelos de predicción del rendimiento aplicando técnicas de Análisis de Redes Sociales; y de la tarea V.C, respecto a la mejora de la plantilla de reglas de asociación de ElWM.
- Capítulo 7: en este capítulo, el diseño de ElWM es extendido y mejorado al incorporar los resultados obtenidos en la investigación de los capítulos anteriores (línea de trabajo IV). En este capítulo se incluyen dos apartados en los que se exponen respectivamente las tareas V.B y V.C, proponiendo un nuevo diseño de ElWM como una Línea de Productos Software así como una primera aproximación a un sistema similar a ElWM de propósito general, de forma que no esté enfocado exclusivamente a datos del área educativa.

Los capítulos 4 y 5 así como el apartado sobre Análisis de Redes Sociales del capítulo 6, que contienen estudios de minería de datos, se estructuran, al menos, en los siguientes apartados:

- Estado del arte: se resumen los trabajos más relevantes relacionados con el área de estudio concreta del capítulo.
- Hipótesis de partida, organización y resumen de los estudios: se explican las hipótesis de partida y objetivos concretos de los estudios incluidos en el capítulo, y se expone un resumen de los mismos.
- Configuración, proceso y resultados de los estudios: se detallan la configuración, proceso seguido y resultados obtenidos en los estudios llevados a cabo para dar respuesta a los objetivos y validar o refutar las hipótesis de partida planteadas.
- Conclusiones y líneas de trabajo futuro: se resumen las conclusiones de los diferentes estudios mostrados y se exponen los retos y líneas de trabajo futuras.

Además de estos apartados, cada uno de los capítulos mencionados pueden incluir otros apartados adicionales cuando procedan, que serán introducidos al inicio del capítulo. El capítulo 7, dado que no es un estudio del área de minería de datos, sino más cercano a la Ingeniería de Software, tiene su propia estructura, que se detalla en el propio capítulo.

En el capítulo 8 se discute sobre los resultados obtenidos en las diferentes líneas de trabajo de esta tesis y se extraen conclusiones generales sobre todas ellas.

Finalmente, en el capítulo 9 se exponen las diferentes líneas de investigación y trabajos futuros que podrían realizarse a partir de lo desarrollado en esta tesis.

El presente documento contiene también dos anexos, que pueden servir de ayuda al lector para complementar la información que se ofrece en la tesis:

- Anexo A.1: Ejemplos de uso de ElWM previos al desarrollo de esta tesis.
- Anexo A.2: Información sobre los cursos utilizados en los estudios.

Capítulo 2

Descripción del contexto, fuentes de datos, procesos y técnicas utilizadas

En este capítulo, se describen el contexto, las fuentes y conjuntos de datos, así como los procesos y técnicas de minería de datos utilizadas para todos los estudios de minería de datos desarrollados en esta tesis.

El capítulo sigue la siguiente estructura: en el apartado 2.1 se describe un resumen del estudio desarrollado para comprender la estructura y desarrollo de los cursos virtuales impartidos en la Universidad, así como de los procesos de enseñanza y aprendizaje seguidos por profesores y estudiantes en estos cursos. En el siguiente apartado 2.2, se presenta un resumen de las características de las bases de datos de los dos sistemas LCMS utilizados en los estudios, Moodle y Blackboard. Seguidamente, en el apartado 2.3 se incluye un resumen estadístico de los cursos de los que se ha hecho uso en esta tesis. En el apartado 2.4 se detallan los datos extraídos de los repositorios de los LCMS y utilizados en esta tesis, que se corresponden con los datos de actividad y rendimiento de los estudiantes. En el apartado 2.5 se explica de forma breve qué técnicas de minería de datos han sido utilizadas en esta tesis, así como la forma en la que han sido configuradas, además de las técnicas y medidas de validación y evaluación elegidas. Finalmente, el apartado 2.6 resume el despliegue de los resultados obtenidos en la herramienta ElWM.

2.1. Los cursos virtuales en la Universidad

Antes de aplicar técnicas de minería de datos, se debe analizar consecuentemente el contexto en el que se pretenden aplicar y qué fines se persiguen. Por tanto, se realizó un exhaustivo análisis de la estructura, desarrollo y proceso de enseñanza de los cursos que, a lo largo del desarrollo de esta tesis, estuvieron disponibles. La mayor parte de estos cursos provinieron de la Universidad de Cantabria, estando alojados en dos plataformas LCMS diferentes, BlackBoard y Moodle. También se realizó la tarea de consultar a distintos profesores sobre el proceso de enseñanza que llevan a cabo en los cursos virtuales, determinado así sus necesidades y planteando cómo los modelos obtenidos por las técnicas de minería de datos podrían ayudar a mejorar sus cursos.

Respecto a las características de los cursos, éstos pueden clasificarse en base a diferentes criterios. Atendiendo a la metodología de enseñanza, se pueden categorizar en dos grupos:

- **E-Learning**: cursos en los que la docencia es completamente virtual, esto es, son impartidos 100 % *on-line* a través de las plataformas LCMS. Existe por tanto una gran cantidad de actividad de los estudiantes sobre la que aplicar técnicas de minería de datos.
- **Blended learning**: son cursos semipresenciales, en los que parte del proceso de enseñanza se realiza de forma virtual a través de los LCMS, mientras que otra parte de la enseñanza se realiza de forma presencial.

Como ya fue mencionado en el capítulo 1, salvo que sea necesario hacer distinción entre estos dos tipos de cursos, en la memoria se hará referencia a todos los cursos, tanto si desarrollan su actividad de forma completamente virtual o sólo parcialmente, con la denominación de cursos virtuales.

Los cursos también pueden ser divididos en dos grandes grupos, según el perfil de los estudiantes matriculados:

- **Transversales**: son cursos con contenidos de formación transversal, en los que se pueden matricular estudiantes de cualquier área de conocimiento. En estos cursos, por tanto, es muy probable encontrarse con estudiantes de muy variados perfiles, como las ciencias sociales, las ciencias puras o las ingenierías. La mayor parte de

estos cursos son *e-learning*, si bien existen algunos cursos semipresenciales en los que la carga de actividad virtual es considerablemente alta.

- **Específicos**: estos cursos tienen contenidos de formación específicos para un área de conocimiento o incluso carrera universitaria específica, por lo que el perfil de los estudiantes matriculados está limitado al área o carrera en el que se imparten. Los cursos de tipo específico utilizados en esta tesis son tanto *e-learning* como semipresenciales.

Existen además otros criterios con los que podrían clasificarse los cursos, de los cuales algunos de los más relevantes son mencionados a continuación:

- **Método de evaluación**: según la forma de evaluar de los profesores, los cursos pueden clasificarse, a alto nivel, como **evaluación continua**, si la evaluación se realiza mediante calificación de tareas entregables; como **evaluación por exámenes**, si se evalúa con exámenes parciales; como **evaluación final**, si se utiliza un único examen final para calificar a los estudiantes; o como **evaluación mixta**, si se combinan las calificaciones de entregables con exámenes parciales o finales.
- **Utilización de herramientas en el proceso de enseñanza**: existen cursos, por ejemplo, en los que la interacción de los estudiantes a través del **foro** tiene un gran peso en el proceso de aprendizaje, bien sea para resolver dudas sobre los contenidos o para realizar trabajos en grupo. Otros cursos, sin embargo, dan un peso más alto a la realización de **auto-test** que permitan a los estudiantes asentar sus conocimientos. En otros casos, por ejemplo, los cursos se estructuran en torno a los **ficheros de contenidos**, dando más peso a su lectura que al uso de otras funcionalidades. La utilización que se les dé a las herramientas del LCMS, por tanto, puede ser otra forma de clasificar a los cursos.

2.2. La actividad almacenada en Moodle y Blackboard

Tanto Moodle como BlackBoard proveen de una base de datos relacional en donde almacenan todos los datos necesarios para identificar los elementos que componen los cursos, su organización y las acciones realizadas por los diferentes actores.

En el caso de Blackboard, existen además un conjunto de vistas, de las cuáles la que más interesa para los objetivos de esta tesis es la vista llamada *rpt_tracking*, en dónde se almacena toda la actividad realizada por los estudiantes.

En la plataforma Moodle, la base de datos también contiene una tabla que almacena cada una de las acciones ejecutadas por los éstos, llamada *mdl_log*. No obstante, existen diferencias con respecto a la vista de *log* ofrecida por Blackboard. Una de ellas, y que como se verá en el apartado 2.4 afecta al proceso de extracción de los datos, es que la tabla de Moodle no apunta el momento de finalización de cada acción, algo que en Blackboard si se puede obtener.

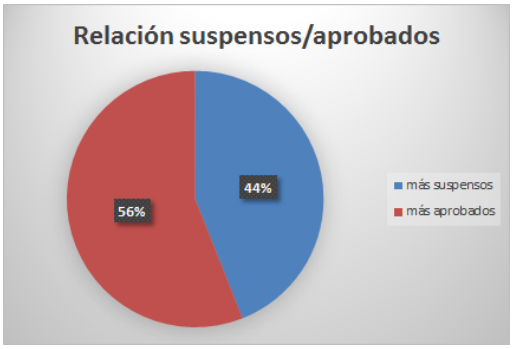
Las tablas y vistas de *log* de Moodle y Blackboard, no obstante, no son las únicas que resultaron necesarias para el desarrollo de esta tesis. A lo largo de la investigación, se ha tenido que hacer uso de otras tablas que almacenaban información necesaria para los estudios, como por ejemplo la tabla *mdl_quiz_attemps* de Moodle, que contiene, entre otra información, la calificación obtenida por los estudiantes en cada de uno de los intentos de realización de los test de un curso.

2.3. Características de los cursos

En el anexo A.2 se recogen las características de los 25 cursos que se han utilizado a lo largo del desarrollo de esta tesis. La mayor parte de los cursos, 21, se alojaban en la plataforma Moodle, mientras que sólo 4 de ellos estaban desplegados en Blackboard. En la Figura 2.1 se muestran un conjunto de gráficas con datos estadísticos sobre las características de estos cursos respecto del rendimiento de los estudiantes.

Puede constatarse al observar la Gráfica 2.1a que, en la mayoría de los cursos, existe un equilibro entre el porcentaje de estudiantes suspensos y el porcentaje de estudiantes aprobados. Solamente en un 16 % de los cursos existe un porcentaje de suspensos superior al 75 %, y en el 24 % de los cursos, este porcentaje de suspensos es menor del 25 %. La mayor parte de los cursos, un 60 %, tienen un porcentaje de suspensos que oscilan entre el 25 % y el 75 %.

(A) Relación de estudiantes suspensos según porcentaje



(B) Relación de estudiantes según su rendimiento en todos los cursos

(C) Relación de estudiantes según su rendimiento en cursos específicos

(D) Relación de estudiantes según su rendimiento en cursos transversales

FIGURA 2.1: Relación del rendimiento de los estudiantes

En cuanto al porcentaje de cursos en los que la mayor parte de los estudiantes suspenden, este asciende al 44 %, habiendo por tanto un 56 % de cursos en los que la mayor parte de los estudiantes aprueban, tal como muestra la Gráfica 2.1b.

Si realizamos la misma comparativa separando los cursos según su carácter, vemos que en el caso de los transversales, Gráfica 2.1c, la mayoría de los estudiantes suele aprobar los cursos. En el caso de los específicos, Gráfica 2.1d, la mitad de ellos son aprobados por la mayoría de los estudiantes, y la otra mitad no.

En el mismo anexo A.2 se muestran las herramientas con mayor relevancia en el desarrollo del proceso de enseñanza y aprendizaje en cada uno de los cursos. Mientras que en el caso de 18 de los cursos, con identificadores del 1 al 5 y del 13 al 25, se dispuso de los *logs* de los LCMS para extraer la información de actividad de los estudiantes en todas las herramientas, no fue el caso de los cursos con identificadores del 6 al 12, cuyos conjuntos de datos fueron proporcionados directamente por los profesores. Por ello, de estos cursos solo se dispuso de la actividad de los estudiantes en las herramientas seleccionadas por estos profesores.

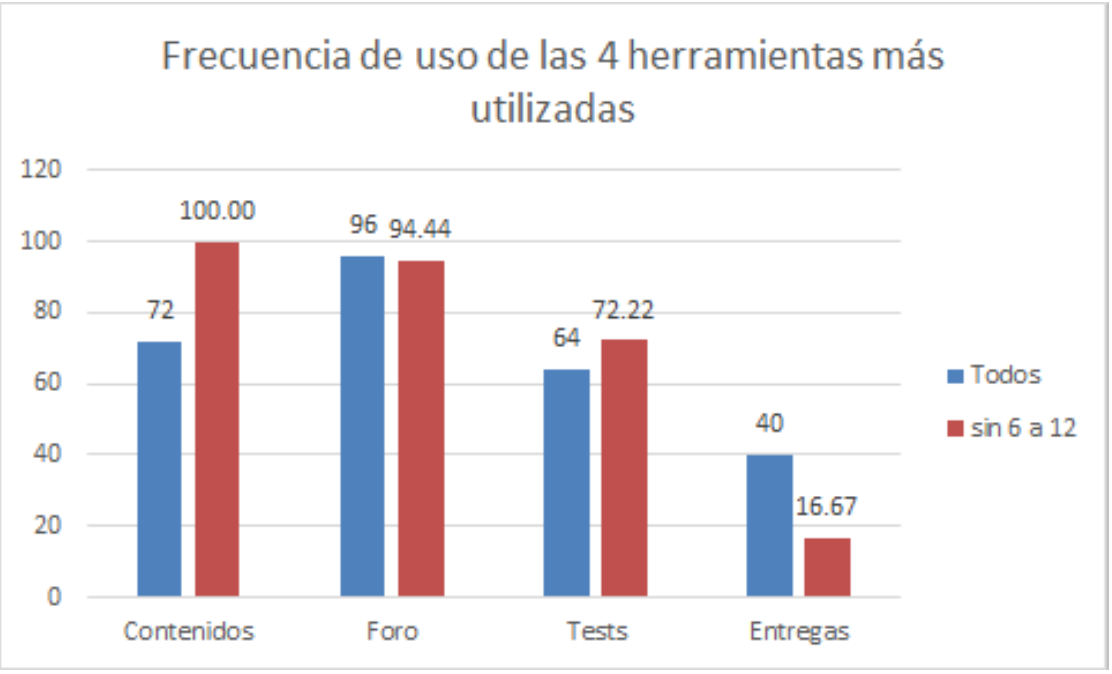


FIGURA 2.2: Frecuencia de uso de las herramientas en los cursos

En la Figura 2.2 se muestra la frecuencia de uso de las herramientas más destacadas, esto es, que tienen una mayor relevancia en el proceso de enseñanza-aprendizaje de cada uno de los cursos. Puede verse como el **foro** es una de las herramientas más utilizadas, denotando el interés que los profesores tienen, al diseñar los cursos, en fomentar la interacción con y entre los estudiantes. Otra de las herramientas más utilizadas en los cursos es la de **recursos**, esto es, la funcionalidad que permite ofrecer los contenidos del curso. De hecho, si no se tienen en cuenta a los cursos del 6 al 12, de los que no se tiene información de actividad sobre esta herramienta, alcanza una frecuencia de uso del 100 %. Los **tests** también tienen especial importancia en la estructura de los cursos, siendo ampliamente utilizados en un 64 %, mientras que la herramienta **entregables**, en la que se pueden consultar y subir los ejercicios y trabajos del curso, tiene una frecuencia de uso también bastante alta, del 40 %.

Al margen de estas cuatro herramientas, existen otras que también tienen un considerable peso en algunos de los cursos, como son los **blogs**, los **wikis**, el **calendario**, el **glosario** o el **correo interno**. Todas estas herramientas, además de otras de las que en determinadas situaciones se estableciese que pudieran tener relevancia para medir la actividad de los estudiantes, fueron utilizadas en el desarrollo de esta tesis.

Un hecho importante a destacar es que no todos los cursos estuvieron disponibles desde

el inicio de la investigación de esta tesis, sino que a medida que se ha ido avanzando en ella, se han ido obteniendo más cursos disponibles de cara a añadirlos y completar la experimentación. Es por este hecho que, a medida que el lector avance en la lectura de este documento, podrá observar como en sucesivos experimentos, el número de conjuntos de datos utilizados va en aumento.

2.4. Extracción y pre-procesado de las medidas de actividad de los estudiantes

En las Tablas 2.1, 2.2, 2.3 y 2.4 se muestra un resumen de las principales medidas de actividad que se extrajeron de los cursos, y que fueron utilizadas como atributos predictores del rendimiento de los estudiantes en esta tesis:

Tabla 2.1: Resumen de las principales medidas generales de actividad

Nombre	Descripción	Plataforma
Tiempo total dedicado	Tiempo total que el estudiantes ha permanecido conectado al curso	Blackboard
N# de sesiones totales	Número de sesiones de conexión totales del estudiante al curso	Blackboard
N# de acciones totales	Número de acciones en todas las herramientas	Ambos
Tiempo dedicado por herramienta	Tiempo total que el estudiantes ha permanecido conectado a una herramienta concreta	Blackboard
N# de sesiones por herramienta	Número de sesiones en los que el estudiante ha visitado una herramienta concreta	Blackboard
N# de acciones por herramienta	Número de acciones en una herramienta concreta	Ambos
Tiempo dedicado por periodo de tiempo	Tiempo total que el estudiantes ha permanecido conectado en un periodo de tiempo concreto (día de la semana, semana concreta, mes, bimestre, otros)	Blackboard
N# de sesiones por periodo de tiempo	Número de sesiones en un periodo de tiempo concreto (día de la semana, semana concreta, mes, bimestre, otros)	Blackboard
N# de acciones por periodo de tiempo	Número de acciones en un periodo de tiempo concreto (día de la semana, semana concreta, mes, bimestre, otros)	Ambos
Tiempo dedicado por periodo de tiempo y herramienta	Tiempo total que el estudiantes ha permanecido conectado en un periodo de tiempo concreto en una herramienta concreta	Blackboard
N# de sesiones por periodo de tiempo y herramienta	Número de sesiones en un periodo de tiempo concreto en una herramienta concreta	Blackboard
N# de acciones por periodo de tiempo y herramienta	Número de acciones en un periodo de tiempo concreto en una herramienta concreta	Ambos

Tabla 2.2: Resumen de las principales medidas de actividad en el foro

Nombre	Descripción	Plataforma
N# de hilos iniciados en el foro	Cantidad de hilos que un estudiante ha iniciado en el foro de la asignatura	Ambos
N# de hilos respondidos en el foro	Cantidad de veces que un estudiante ha respondido a un hilo en el foro de la asignatura	Ambos
N# de hilos o respuestas modificadas en el foro	Cantidad de veces que un estudiante ha modificado el texto de un hilo o respuesta en el foro de la asignatura	Ambos
N# de hilos y respuestas leídas	Número de hilos y respuestas en el foro que un estudiante ha leído	Ambos
N# de foros subscritos	Número de foros a los que un estudiante se ha subscrito para recibir notificaciones	Moodle

Tabla 2.3: Resumen de las principales medidas de actividad en las herramientas Blog, Glosario y Wiki

Nombre	Descripción	Plataforma
# entradas en el blog	Cantidad de entradas escritas en el blog	Moodle
# modificaciones en el blog	Cantidad de veces que el estudiante ha modificado una entrada de un blog	Moodle
# entradas en glosario	Cantidad de entradas escritas en el glosario	Moodle
# modificaciones en glosario	Cantidad de veces que el estudiante ha modificado una entrada del glosario	Moodle
# entradas en wiki	Cantidad de entradas escritas en la wiki	Moodle
# modificaciones en wiki	Cantidad de veces que el estudiante ha modificado una entrada de una wiki	Moodle

Tabla 2.4: Resumen de las principales medidas de actividad en las herramientas de Test, Correo y Entregables

Nombre	Descripción	Plataforma
N# de auto-test realizados	Cantidad de veces que un estudiante ha realizado un auto-test	Ambos
N# de auto-test superados	Porcentaje de auto-test superados por el estudiante, con respecto al total realizados	Ambos
N# de auto-test suspendidos	Porcentaje de auto-test suspendidos por el estudiante, con respecto al total realizados	Ambos
N# de correos internos enviados	Cantidad de correos internos enviados a otros usuarios por el estudiante	Blackboard
N# de correos internos respondidos	Cantidad de correos internos respondidos a otros usuarios por el estudiante	Blackboard
N# de correos internos leídos	Cantidad de correos internos leídos por el estudiante	Blackboard
% de entregables realizados	Porcentaje de tareas entregadas con respecto al total por parte del estudiante	Ambos
% de entregables no realizados	Porcentaje de tareas no entregadas con respecto al total por parte del estudiante	Ambos

Como puede ser observado, dadas las diferencias entre Moodle y Blackboard, tanto al respecto de la actividad almacenada como de las herramientas ofrecidas, las medidas de actividad, si bien en su mayoría son las mismas, difieren entre unos cursos y otros según se alojen en una u otra plataforma. Así, por ejemplo, la vista de *log* de Blackboard permite extraer de la misma durante cuánto tiempo el estudiante permaneció conectado en una herramienta concreta, dado que existen dos campos indicando los momentos de inicio y fin de la acción. En el caso de Moodle, sólo almacena el momento en el que se realizó la acción, por lo que a priori no se puede conocer cuánto tiempo duró. Aunque existen propuestas para extrapolar ese valor, en el desarrollo de esta tesis no se han incorporado, trabajando únicamente con datos de actividad cuyos valores estén garantizados.

Con los datos de actividad extraídos, se construyeron varios conjuntos de datos, cuyo número se ha ido incrementando a lo largo del desarrollo de la tesis a medida que se

fue obteniendo el acceso a más cursos. A los conjuntos de datos para la predicción del rendimiento de los estudiantes se les ha añadido un campo indicando el rendimiento que los estudiantes tuvieron en el curso. Por tanto, en estos conjuntos de datos las filas o instancias representan la actividad y rendimiento de cada uno de los estudiantes en un curso alojado en Moodle o Blackboard, siendo el rendimiento el atributo de clase a predecir, y teniendo las medidas de actividad como atributos predictores.

Las principales características de los conjuntos de datos son las siguientes:

- **Atributos predictores numéricos:** todas las medidas de actividad, ya mencionadas, son de tipo numérico, por lo que los modelos de clasificación que se vayan a construir, inicialmente, sólo tienen como predictores a atributos de este tipo. No obstante, en determinados momentos del estudio, se utilizarán técnicas de discretización sobre determinadas medidas de actividad para completar los experimentos. Así también, a lo largo del desarrollo de la tesis, se han utilizado otros atributos como predictores. Por ejemplo, durante los primeros pasos de la investigación se incorporaron datos demográficos y estilos de aprendizaje de los estudiantes junto con los datos de actividad. En el estudio de SNA, además, se utilizan medidas que informan de la interacción social de los estudiantes, también junto a las de actividad, para tratar de predecir el rendimiento de los estudiantes. Por otro lado, en el estudio del capítulo 4, se extraen un conjunto de meta-datos que caracterizan los conjuntos de datos y a los que se les aplican diferentes técnicas de minería de datos. También en el estudio con algoritmos de reglas de asociación del apartado 6.2 se utilizan otros atributos que, al contrario que los de actividad, son todos discretos. Como norma, si en la experimentación no se menciona de forma explícita, se asumirá que los atributos predictores son todos numéricos y en base a las medidas actividad mencionadas en esta sección.

- **Conjuntos de datos pequeños:** dado que cada instancia en los conjuntos de datos representa la actividad y rendimiento de un estudiante en un curso, y debido a que el número de estudiantes matriculados en los cursos utilizados en los estudios de esta tesis varía entre 9 y 502, los conjuntos de datos tienen una cantidad baja de instancias, algo que, como el lector podrá comprobar en el apartado 3, es habitual en el campo de estudio del EDM. Por ello, en la experimentación, no sólo se han utilizado conjuntos de datos de un solo curso, sino que se han generado otros

conjuntos fruto de combinar la actividad y rendimiento de los estudiantes de dos o varios de estos cursos cuando comparten características similares.

- **Dos valores de clase:** el principal objetivo de esta tesis se enfoca en la predicción del rendimiento de los estudiantes, el cual es medido en base a la calificación que éstos obtienen al finalizar los cursos. Esta calificación, que en la mayor parte de los cursos varía en un rango del 0 al 10, se podría discretizar, para ser utilizada como clase, de diversas formas. Inicialmente, se valoró la utilización de conjuntos de datos con hasta 5 valores de clase, con las siguientes correspondencias según la calificación obtenida: del 0 al 4.9, clase **suspenso**; del 5 al 6.9, clase **aprobado**,; del 7 al 8.9, clase **notable**; y del 9 al 10, **sobresaliente**, y clase **abandono** si el estudiante no llegó a completar el curso, pudiendo todos estos rangos variar según las observaciones hechas al analizar el curso. No obstante, y debido al ya mencionado reducido tamaño de los conjuntos de datos, tener tantos valores de clase no permite obtener modelos de predicción precisos. Por ello, aunque en determinados experimentos incluidos en este documento se han utilizado conjuntos de datos con hasta 5 valores de clase, en general, los conjuntos utilizados tienen únicamente dos clases: **suspenso**, si el estudiante obtuvo una calificación de entre 0 y 4.9 sobre 10, y **aprobado**, si la calificación estaba entre 5 y 10. Esta clasificación del rendimiento de los estudiantes se asumirá por defecto a lo largo de este documento, a menos que explícitamente se indique otra clasificación.

- **Limpieza de los datos de actividad:** dado que los datos de actividad son extraídos de una tabla de *log* de los LCMS, y previa comprobación del proceso de extracción de estos datos, se puede afirmar que los datos utilizados en esta tesis no presentaban ruido por fallo humano, esto es, debido a que alguien hubiese introducido mal algún dato, ya que esta posibilidad no existe. Esto es así para todos los conjuntos de datos obtenidos de los cursos del 1 al 5 y del 13 al 25, con los que se ha tenido un control completo del proceso de extracción de los datos. No obstante, este control no ha sido igualmente efectivo con los datos de los cursos del 6 al 12. Es por ello que con estos conjuntos datos no había garantía de ausencia de fallos humanos, y fueron previamente preprocesados para eliminar cualquier ruido de este tipo. Concretamente, en algunos de estos conjuntos de datos se encontraron valores de actividad negativos, algo imposible con ninguna de las medidas utilizadas. A menos que en el texto, para un estudio concreto, se indique

lo contrario, el lector asumirá que estos valores han sido eliminados convirtiéndolos en nulos, esto es, en valores desconocidos.

En cada estudio mostrado en el presente documento, se mencionarán las características de los conjuntos de datos utilizados. Sobre estos conjuntos, diversas técnicas de pre-procesado fueron aplicadas a lo largo del desarrollo de esta tesis para garantizar la calidad de los mismos y previo paso a la extracción de modelos mediante la aplicación de las técnicas de minería de datos. En la experimentación, a menos que explícitamente se indique que se aplicaron unas técnicas de pre-procesado con un objetivo concreto, se asumirá siempre que los conjuntos de datos utilizados fueron pertinentemente pre-procesados previo paso a la fase de modelado, utilizando técnicas de selección de atributos y de detección y eliminación de datos incongruentes (como pueden ser los valores negativos en atributos de actividad), entre otras.

2.5. Técnicas utilizadas para la obtención y evaluación de los modelos de minería de datos

A lo largo del desarrollo de los diferentes estudios mostrados en esta tesis, se utilizaron distintos conjuntos de técnicas de minería de datos. En cada apartado, por tanto, se informa de las técnicas usadas, así como de la configuración de los parámetros y del software utilizado. Por defecto, si no se mencionase explícitamente, el lector podrá asumir que se ha utilizado la implementación de los algoritmos proporcionada por Weka [54] configurados con los parámetros por defecto. En este caso, y salvo que existan otras indicaciones, los nombres de los algoritmos se corresponderán con el que tienen en las clases Java que los implementan en Weka, y en las cuales se pueden encontrar referencias sobre el funcionamiento de los mismos.

Dado que el objetivo principal de esta tesis es poder mostrar a usuarios no expertos modelos de minería de datos, en gran parte de los estudios mostrados en el presente documento se utilizaron técnicas cuyos modelos fuesen fácilmente interpretables. Así, por ejemplo, se consideran modelos interpretables aquellos basados en reglas, como los ofrecidos por las técnicas OneR, JRip y Ridor; basados en árboles, como los modelos de C4.5 y CART; o ciertos modelos estadísticos que puedan ser manejados por un usuario no experto, como pueden ser los retornados por NaïveBayes y BayesNetwork.

No obstante, en determinados estudios, cuando se consideró oportuno, se incluyeron otros algoritmos que, si bien sus modelos no son fácilmente interpretables, fueron útiles en la experimentación. Así, por ejemplo, en el capítulo 5, se muestran los resultados de un amplio conjunto de clasificadores de cara a comprobar la viabilidad de la propuesta, sin importar a priori la facilidad de interpretar sus modelos.

En cuanto a la evaluación de los modelos de clasificación, en cada experimento se establecieron tanto los procesos y técnicas como las medidas de evaluación a utilizar de cara a comparar su rendimiento. Dado que se pretende poder mostrar estos modelos a usuarios no expertos, y si bien a lo largo del desarrollo de este trabajo se han tenido en cuenta diferentes medidas de rendimiento, como en el caso de la clasificación son la *medida F (f-Measure)* o el *Area bajo la curva (AUC)*, en las comparativas de modelos orientadas a mostrar los resultados a usuarios no expertos, el análisis de resultados se realizó en base a medidas de rendimiento que fuesen sencillas de entender, como el *accuracy*, o el *TPrate* y el *TNrate*, que por defecto, y a menos que en el texto se indique lo contrario, estas dos últimas corresponderán respectivamente con el porcentaje de estudiantes suspensos (clase positiva) y aprobados (clase negativa) correctamente clasificados.

2.6. Despliegue de los resultados: mejora y extensión de ElWM

Una vez finalizada la experimentación relativa al proceso de minería de datos, se realizó un nuevo diseño de la herramienta E-learning WebMiner que incluye nuevas funcionalidades y mejoras. Entre ellas, se encuentran un sistema de recomendación automática de algoritmos de clasificación, un proceso de detección de comportamientos anómalos para la mejora de los modelos de predicción, la posibilidad de extraer modelos de predicción del rendimiento en base al comportamiento social de los estudiantes, y la mejora del proceso de extracción de reglas de asociación. Igualmente, se han dado los primeros pasos en una propuesta de extensión de E-learning WebMiner a otros campos distintos del educativo.

Capítulo 3

Educational Data Mining: estado del arte

Las técnicas y procesos de minería de datos se han venido aplicando extensamente durante los últimos años en el área educativa, con objeto de extraer patrones y modelos que ayuden a mejorar el proceso de enseñanza y aprendizaje. La aplicación de dichas técnicas ha sido reforzada por la aparición de los sistemas computacionales de enseñanza y aprendizaje, como son por ejemplo los sistemas de tutoría inteligente y las plataformas *e-learning*, que almacenan la actividad que los estudiantes realizan en ellas.

En la primera sección 3.1 de este capítulo se realiza un repaso de la minería de datos aplicada al campo educativo y cuáles son los principales objetivos de su aplicación. En el apartado 3.2 se enumeran y analizan los trabajos en los que se aplican técnicas de minería de datos para la predicción del rendimiento de los estudiantes.

3.1. La minería de datos aplicada al campo educativo

Según [10], la disciplina EDM emerge como un paradigma orientado a diseñar modelos, tareas, métodos y algoritmos con el objetivo de explorar los datos del entorno educativo. Por ello, en el área del EDM se trabaja con el objetivo de encontrar patrones y realizar predicciones para caracterizar el comportamiento de los estudiantes y sus logros o sus conocimientos sobre los contenidos del dominio, para detectar estudiantes en riesgo de abandono, o para predecir su rendimiento, entre muchos otros [55].

Actualmente, y debido al auge de los estudios en EDM, tal y como puede constatarse en artículos que han aglutinado los diferentes trabajos realizados en este área durante las dos últimas décadas [56], [10], [12], se ha creado una asociación internacional, llamada *International Educational Data Mining Society (IEDM)* [16], dedicada a organizar eventos y gestionar medios de publicación enfocados exclusivamente al EDM. En 2007, se organizó el primer *WorkShop* sobre EDM dentro del congreso internacional de "Inteligencia Artificial en Educación", y, en 2008, se celebró el primer congreso internacional e independiente sobre EDM, llamado *Internacional Conference on Educational Data Mining*, que desde entonces viene celebrándose de forma anual. En 2009, además, se publicó por parte de esta asociación el primer número de la revista *"Journal of Educational Data Mining"*, que publica artículos referentes al área del EDM, y de la que, hasta la fecha de redacción del presente documento, se han editado 6 volúmenes.

El auge y establecimiento de la disciplina EDM ha dado lugar a un crecimiento en el número de estudios publicados en este campo, así como de los temas u objetivos de dichos estudios, tal y como puede constatarse en [10]. En estos estudios, los autores aplican diferentes tipos de técnicas, tanto supervisadas como no supervisadas, para dar respuesta a las necesidades del entorno educativo.

En el área de las no supervisadas, por ejemplo, las técnicas de *clustering* son ampliamente utilizadas con diferentes objetivos. Así, por ejemplo, encontramos trabajos en los que los autores hacen uso de ellas para tratar de agrupar los materiales de los cursos según su tipo [57–59] con objeto de mejorar la búsqueda de contenidos en plataformas *e-learning*; o para agrupar a los estudiantes según su comportamiento o actividad en el curso, tomando como datos de actividad su comportamiento navegacional [60–62] o su estilo de aprendizaje y su personalidad [63], entre otros, y con objeto de, por ejemplo, proveer de un proceso de enseñanza personalizado a cada perfil de estudiante [64]. Otros autores han mostrado como las técnicas de *clustering* tienen utilidad también para, por ejemplo, mejorar el entorno de aprendizaje [65, 66] y hacer más efectivo el proceso de enseñanza [67, 68]. En cuanto a las técnicas de *clustering* utilizadas, podemos encontrar una gran variedad de ellas en estos trabajos: *Expectation Maximization* [69, 70], *Hierachicals* [59, 71–73], *Self-Organizing Map* (SOM) [74, 75] o *k-Means* [76–78], entre otras. También las reglas de asociación tienen aplicación en el entorno educativo, por ejemplo, para extraer patrones de interés sobre el comportamiento de los estudiantes que informen al profesor de cómo interaccionan con su curso [79], encontrar errores que los estudiantes

cometen frecuentemente de forma conjunta [80], o incluso modelar el rendimiento de los estudiantes en base a su actividad [81].

Pero, indudablemente, son las técnicas supervisadas de clasificación y regresión las que tienen una implantación y uso más extendido en EDM [82]. Los sucesivos resúmenes publicados acerca de las diferentes técnicas utilizadas en EDM y su frecuencia de uso en los diferentes trabajos publicados en este área denotan una creciente tendencia a utilizar técnicas predictivas [12]. En el último de estos resúmenes [10], publicado en 2014, se puede observar claramente como, en términos comparativos, la mayor parte de los trabajos en EDM, publicados en medios de alto impacto, se enfocan precisamente en la aplicación de técnicas de predicción.

En base a lo expuesto en el capítulo dedicado a las tareas de clasificación del *Handbook of Educational Data Mining* de 2010 [82], las técnicas de predicción son utilizadas para resolver 4 grandes problemas en EDM, cuyo encuadramiento se muestra en la Tabla 3.1:

Tabla 3.1: Encuadramiento de los 4 grandes problemas en EDM en los que se aplican técnicas de clasificación

I. Predecir el rendimiento académico: el objetivo es el de clasificar el éxito académico de los estudiantes a nivel universitario, teniendo como objetivos la predicción del abandono al principio de los estudios [83], el tiempo necesario para graduarse [84], el rendimiento general [85], o la necesidad de recibir clases de refuerzo [86].	**II. Predecir el rendimiento en un curso**: el objetivo se centra en tratar de predecir si un estudiante aprobará o suspenderá la asignatura [87, 88], si la abandonará [89], o incluso predecir su calificación numérica [90].
III. Predecir el éxito en la siguiente tarea: se basa en tratar de predecir el rendimiento de un estudiante en una tarea, dado su rendimiento en tareas anteriores [91].	**IV. Predecir o clasificar hábitos, motivación y habilidades**: consiste en descubrir el nivel de motivación del estudiante [92], el estilo de aprendizaje [93], su facilidad para manejarse en un entorno e-learning [94], o recomendar una estrategia de intervención [95].

No obstante, en la literatura de los últimos años podemos encontrar otros trabajos en los que el objetivo de la predicción es diferente a los mencionados. Por ejemplo, en un estudio reciente [96], los autores aplicaron técnicas de clasificación para predecir la posibilidad de que a un estudiante se le conceda una beca de estudios, en base a las notas obtenidas en las diferentes asignaturas de un semestre y la realización de actividades extracurriculares. En otro trabajo [97] se estudió qué factores demográficos pueden afectar a la posibilidad de que un estudiante se matricule o no en una determinada universidad.

En el siguiente apartado 3.2 se exponen los principales trabajos que se han analizado con objeto de aportar soluciones en el área de la predicción del rendimiento de los estudiantes en cursos virtuales.

3.2. Prediciendo el rendimiento de los estudiantes

Tratar de determinar los factores que afectan al rendimiento de los estudiantes, o aquéllos que determinan cuándo un estudiante está en riesgo de abandonar sus estudios, es un problema que viene preocupando desde hace décadas, como puede observarse en la literatura. En un estudio de 1985, Butcher et al. [98] abordaron la problemática de predecir el rendimiento de los estudiantes universitarios de Ciencias de la Computación, mostrando cómo puede afectar a esta tarea el perfil de los estudiantes cuando se matriculan en estos estudios. Ciertos datos sobre los estudiantes, como las notas obtenidas en diversas asignaturas de sus estudios pre-universitarios (matemáticas, lengua, y otros), o si realizaron cursos de introducción a la programación antes de entrar en la carrera, mostraron ser útiles para determinar qué calificación obtendría cada estudiante en el primer semestre de la carrera, así como en el curso de Introducción a las Ciencias Computacionales. No obstante, se basaron únicamente en métodos estadísticos para extraer sus conclusiones. El desarrollo de nuevas técnicas en el área del *Machine Learning* propició la aparición de trabajos en los que eran aplicadas con objeto de modelar el rendimiento de los estudiantes. Así, por ejemplo, en un estudio de 1994 Fausett et al. utilizaron 2 técnicas de redes neuronales, *backpropagation* y *counterpropagation* para predecir el rendimiento de los estudiantes de un curso de Cálculo, en base a su rendimiento parcial en los tests de la asignatura, mostrando la utilidad de estas técnicas en el campo educativo [99].

No obstante, no fue sino hasta los primeros años del siglo XXI cuando comenzaron a aparecer estudios importantes en los que se aplicaban técnicas propias del área de minería de datos con objeto de predecir el rendimiento, o incluso la posibilidad de abandono, de los estudiantes en cursos virtuales, coincidiendo en el tiempo con el desarrollo de sistemas avanzados de tutores inteligentes y los LCMS como Moodle y Blackboard. En las Tablas 3.2, 3.3, 3.4 y 3.5 se muestra un resumen de los principales trabajos de investigación enfocados en la aplicación de técnicas de minería de datos en este ámbito, recogiéndose los detalles de los experimentos que serán posteriormente discutidos.

En 2003, Kotsiantis et. al. publicaron un trabajo pionero [89] en el que aplicaron cinco técnicas de minería de datos de diferente paradigma (bayesianas, árboles de decisión, "lazy", etc.) con objeto de predecir qué estudiantes están en riesgo de abandono y determinar la técnica más adecuada, utilizando los datos demográficos y las calificaciones

obtenidas en las tareas entregables de un curso *e-learning* por parte de los 354 estudiantes matriculados. Los autores observaron que, debido al bajo número de instancias del conjunto de datos, una de las técnicas bayesianas utilizadas en su estudio, NaïveBayes, era la que mejor rendimiento obtenía, en base al *accuracy*. Un año más tarde, los mismos autores publicaron otro trabajo [100] en el cuál se utilizaban los mismos datos de los estudiantes para modelar no ya el riesgo de abandono, sino su rendimiento en el curso, en base a la nota numérica y utilizando 6 algoritmos de regresión diferentes.

El interés por mejorar los modelos de predicción del rendimiento de los estudiantes llevaron a Behrouze et al. [90, 101] a proponer la combinación de los resultados de varios clasificadores, utilizando para ello una técnica basada en algoritmos genéticos. En este estudio se utilizaron como datos la actividad de los estudiantes de un curso alojado en un sistema LCMS conocido como LON-CAPA (acrónimo de *Learning OnLine Network with Computer-Assisted Personalized Approach*), que ofrece la posibilidad de realizar tests *on-line*. También en [102] se utilizaron técnicas genéticas con objeto de mejorar la predicción del rendimiento, en este caso no para combinar los resultados de varios clasificadores, sino para inducir árboles de decisión.

Hämäläinen et al. [87] extendieron aún más la comparativa de modelos de predicción del rendimiento de los estudiantes, utilizando tanto técnicas de regresión para predecir la nota numérica, así como técnicas de clasificación bayesianas que predijesen el valor de una clase discreta que indicaba si los estudiantes aprobarían o suspenderían el curso. En esta comparativa empírica se mostró cómo los clasificadores bayesianos eran capaces de obtener modelos predictivos con un *accuracy* superior al 80 %, concluyendo los autores, al igual que ya hicieran Kotsiantis et. al [100], que este tipo de técnicas tienen un buen rendimiento para conjuntos de datos pequeños. No obstante, este estudio carece de una comparativa con otros tipos de técnicas sobre la que fundamentar estas conclusiones.

En 2006, se publicó de la mano de Calvo-Flores et al. un estudio [103] enfocado a mostrar la utilidad de las medidas de actividad de los estudiantes extraídas de varios cursos alojados en Moodle para predecir por sí solas el rendimiento de los estudiantes, y entre las que se encuentran el número de visitas al curso por mes o las visitas al temario del curso. Los autores se valieron únicamente de una técnica de redes neuronales, con varias configuraciones, para demostrar que efectivamente la actividad de los estudiantes en estas plataformas tiene un gran potencial predictor.

El interés en la aplicación de técnicas de minería de datos para predecir el rendimiento de los estudiantes creció notablemente entre los años 2008 y 2009, coincidiendo con la celebración de la I edición del congreso *Educational Data Mining*. A partir de ese momento, y hasta la fecha, han aparecido numerosos artículos dedicados en exclusiva a este problema, y en los que se utilizan diferentes tipos de atributos para caracterizar a los estudiantes, así como distintos tipos de técnicas con objeto de obtener los modelos de predicción.

En muchos estudios se han utilizado datos demográficos, socioeconómicos y las calificaciones obtenidas en las tareas o pruebas parciales de los propios cursos para tratar de predecir el rendimiento final del estudiante [104–119]. Igualmente, en la literatura se pueden encontrar propuestas de utilización de otros tipos de atributos que caractericen a los estudiantes y sirvan para la predicción, como es el caso de [120], en el que los autores utilizaron un test de personalidad para conocer su interés en los cursos, su talento o su nivel de motivación, o en [121], en donde se usaron como atributos predictores los estilos de aprendizaje de los estudiantes extraídos mediante el test *Solomon-Felder's Index of Learning Styles* [122].

No obstante, cada vez son más los trabajos que hacen uso de medidas de actividad de los estudiantes en los LCMS con este mismo propósito. En [123], los autores utilizaron exclusivamente la actividad en Moodle de los estudiantes para predecir su rendimiento en cursos *e-learning*. En uno de los estudios más completos en cuanto a comparativa de algoritmos de minería de datos para la predicción del rendimiento de los estudiantes [124], Romero et al. [124–126] aplicaron un conjunto de 21 clasificadores sobre la actividad de los estudiantes en 7 cursos alojados en Moodle, con objeto de estudiar el rendimiento de cada una de estas técnicas, y concluyeron que no existe una técnica que sea mejor que el resto en todos los casos. No obstante, destacaron el hecho de que debido al perfil de usuario no experto de los profesores de cursos virtuales, al que van dirigidos estos modelos, es conveniente que en estos estudios se utilicen clasificadores cuyos modelos sean sencillos de interpretar.

Zafra et. al [127, 128] propusieron una nueva aproximación al modelado del rendimiento de los estudiantes, en base a la actividad desarrollada en Moodle, mediante la aplicación de técnicas de clasificación Multi-instancia [129] (*Multiple Instance Learning* en

inglés, MIL por sus siglas), e implementaron un nuevo algoritmo basado en programación genética (G3P [130]) adaptado al paradigma MIL y llamado G3P-ML, obteniendo modelos de predicción con buen rendimiento. Un estudio más reciente de los mismos autores [131] incluyó en la comparativa clasificadores de minería de datos tradicionales (no MIL), extrayéndose del mismo que la clasificación MIL puede ayudar a mejorar el rendimiento de los modelos de predicción. En estos trabajos también se destacó la necesidad de ofrecer modelos sencillos de entender por parte de usuarios no expertos.

Los algoritmos de reglas de asociación también tienen su protagonismo en la generación de modelos de predicción del rendimiento de los estudiantes. En [81] se estudiaron diferentes algoritmos de reglas que permitiesen obtener patrones de comportamiento en la actividad de los estudiantes, incluyendo aquellos que definen el rendimiento que estos obtienen en los tests de un curso alojado en Moodle. Luna et al. [132] aplicaron algoritmos de reglas de asociación que buscan patrones poco frecuentes, como *Aprori-Infrequent*, para encontrar patrones de actividad de los estudiantes inusuales con respecto al rendimiento obtenido, esto es, estudiantes cuya actividad no se corresponde, en términos generales, con la actividad llevada a cabo por la mayor parte de estudiantes que tuvieron un rendimiento similar.

Otros tipos de medidas de actividad de los estudiantes, al margen de las que puedan ser extraídas directamente de las bases de datos de los LCMS como Moodle, han cobrado relevancia en los últimos años. Sorour et al. [133, 134] utilizaron como atributos predictores medidas extraídas mediante la aplicación de técnicas de *Text Mining* y *Latent Semantic Analysis (LSA)* sobre los comentarios que los estudiantes dejan en el curso. En [135] se aplicaron técnicas de Análisis de Redes Sociales (*Social Network Analysis* en inglés, SNA por sus siglas) para extraer medidas que reflejasen el comportamiento social de los estudiantes en los foros de cursos alojados en Moodle. En este estudio, los autores combinaron estas medidas con otras de actividad de los estudiantes en la plataforma, concluyendo que las medidas SNA son considerablemente útiles de cara a predecir el rendimiento de los estudiantes. También en este mismo trabajo, los autores utilizaron no solamente clasificadores tradicionales, sino técnicas de *clustering* con el mismo propósito, concluyendo que, si bien este tipo de técnicas obtenían un buen rendimiento en algunos de los casos estudiados, en general no generaban mejores resultados que los clasificadores tradicionales. Conclusiones similares al respecto de la predicción del rendimiento de los estudiantes fueron extraídas en [136].

En [137], Molina et al. reflejaron la preocupación existente por buscar aquellas técnicas de clasificación que obtuviesen buenos modelos de predicción del rendimiento de los estudiantes, y realizaron un estudio para establecer cómo afectan las características de los conjuntos de datos, como el número de instancias o de atributos, al rendimiento de los clasificadores. Concretamente, en este estudio los autores trataban de determinar la mejor configuración de los parámetros del algoritmo C4.5 en base a la características de los conjuntos de datos que contienen la actividad y el rendimiento de los estudiantes. Concluyeron que una buena configuración del algoritmo, basada en las mencionadas características, permite obtener modelos de predicción con un rendimiento superior al de otros clasificadores, como NaïveBayes.

3.3. Conclusiones sobre el estado del arte

El uso de la minería de datos en el campo educativo está en auge, existiendo un número cada vez más creciente de trabajos en los que se aplican diferentes tipos de técnicas, como agrupamiento, reglas de asociación y clasificación. De entre los muchos tipos de patrones que pueden ser extraídos con minería de datos, la generación de modelos que permitan predecir el rendimiento de los estudiantes es uno de los problemas más estudiados. Las técnicas de *Machine Learning*, y más concretamente las técnicas de minería de datos orientadas a la clasificación, han mostrado su potencial para generar dichos modelos, y el desarrollo de los sistemas de tutores inteligentes y de los LCMS ha propiciado la aparición de un cada vez mayor número de trabajos en los que se aplican estas técnicas y se buscan los mejores modelos que permitan predecir el rendimiento de los estudiantes en base a la actividad que estos desarrollan en cursos virtuales.

Un hecho destacable en todos los trabajos, y que puede observarse en las tablas-resumen, es el bajo número de instancias de los conjuntos de datos con los que se trabaja. Si bien existen unos pocos trabajos en los que el número de estudiantes, y por tanto de instancias, supera los 1000, lo habitual es encontrarse con conjuntos de datos que no superen las 500 instancias, siendo muy común encontrar estudios en los que los conjuntos de datos tienen entre 100 y 300 instancias, e incluso por debajo de 100. Este hecho ya se destacaba en el libro *Handbook of Educational Data Mining* [15], en el que se afirma que, debido a ello, las técnicas bayesianas podrían tener un mejor rendimiento que otros tipos de técnicas, como las basadas en árboles de decisión. Aunque en los primeros trabajos de

Tabla 3.2: Trabajos relacionados en el área de la predicción del rendimiento de estudiantes en cursos (1)

Ref.	Año	Nº Estudiantes	Atributos	Clase	Nº de técnicas	Medidas de evaluación
[89]	2003	354	Demográficos, Calificaciones en entregables, Otros	abandona / no abandona	6	Acc.
[90, 101]	2003	227	Actividad en LON-CAPA [138]	9 clases (de menor a mayor rendimiento)	10	Acc.
[102]	2003	410	Demográficos, rendimiento en pruebas parciales	suspende / aprueba	2	Acc
[139]	2004	354	Demográficos, Calificaciones en entregables, Otros	Nota Numérica	6	MAE, RAE; MRSE, RSE
[87]	2006	88-125	Calificaciones en ejercicios	suspende / aprueba; nota numérica	5	Acc., TP, TN
[103]	2006	240	Actividad en Moodle	supende/ aprueba	1	Acc.
[125, 126]	2008	135-438	Actividad en Moodle	suspenso / aprobado / notable / sobresaliente	21	Acc.
[104]	2008	1407	Demográficos	suspende / aprueba (0,1 para regresión)	1	Acc, MSE, NMSE, MaxAE, MinAE

Tabla 3.3: Trabajos relacionados en el área de la predicción del rendimiento de estudiantes en cursos (2)

Ref.	Año	Nº Estudiantes	Atributos	Clase	Nº de técnicas	Medidas de evaluación
[123]	2008	28-37	Actividad Moodle	nota numérica	2	MAE, R
[127]	2009	419	Actividad en Moodle	aprueba / suspende	16	Acc., TP, TN
[105]	2009	495-516	Demográficos, Rendimiento en otros cursos, Cursos matriculados, Rendimiento pre-universitario	abandona / no abandona	7	Acc., TP, TN, FP, FN
[106]	2009	63-130	Demográficos, Actividad, Calificación en tareas	abandona / no abandona	6	Acc., TP, TN
[121]	2009	71	Estilos de aprendizaje (extraídos de encuesta)	bajo / medio / alto	1	Acc.
[107]	2009	1619	Demográficos, Rendimiento en otros cursos	sin riesgo / riesgo / riesgo alto / abandona	5	Acc.
[108]	2010	1347	Demográficos, Calificaciones en entregables, Otros	suspende / aprueba	15	Acc., T.
[140]	2010	627-720	Actividad en tests	bajo / medio / alto / muy alto	4	RMSE, Acc, TP, TN, FP, FN
[109]	2010	670	Demográficos, Socioeconómicos, Calificaciones en otros cursos	suspende / aprueba	10	Acc., TP, TN
[110]	2011	8–35	Demográficos, Calificaciones en ejercicios, Asistencia a clase	bajo / medio / alto	5	Acc., f-Measure

Tabla 3.4: Trabajos relacionados en el área de la predicción del rendimiento de estudiantes en cursos (3)

Ref.	Año	Nº Estudiantes	Atributos	Clase	Nº de técnicas	Medidas de evaluación
[131]	2011	419	Actividad en Moodle	aprueba / suspende	28	Acc., TP, TN
[141]	2011	628	Actividad en tests de ASSISTments	nota numérica	1	MAE, RMSE
[128]	2012	419	Actividad en Moodle	aprueba / suspende	16	Acc., TP, TN
[111]	2012	298-481	Rendimiento en otros cursos, ABET outcomes	suspende / aprueba	4	Acc.
[137]	2012	88-670	Demográficos, Socioeconómicos, Calificaciones en otros cursos, Actividad en Moodle	Discreta con entre 2 y 6 valores	13	Acc.
[112]	2012	354	Demográficos, Calificaciones en entregables, Otros	Nota Numérica	6	MAE, RAE, MRSE, RSE
[124]	2013	135 -438	Actividad en Moodle	suspenso / aprobado / notable / sobresaliente	21	Acc.
[113]	2013	111	Demográficas, Calificaciones en parciales	nota numérica en examen final	1	Beta Coefficient, Pearson Correlation
[114]	2013	323	Calificaciones en otros cursos y en tareas	nota numérica en examen final	4	APA, PAP

Tabla 3.5: Trabajos relacionados en el área de la predicción del rendimiento de estudiantes en cursos (4)

Ref.	Año	Nº Estudiantes	Atributos	Clase	Nº de técnicas	Medidas de evaluación
[115, 116]	2013	1650	Demográficos, otros	abandona / no abandona	1	Acc.
[117]	2013	261	Demográficos, Rendimiento en pruebas, Otros	bajo / medio / alto	4	Acc., TP, TN
[120]	2013	71	Inteligencia, Talento, Intereses, Motivaciones (medidas por cuestionario)	bajo / medio / alto / muy alto	1	RMSE
[135]	2013	114	Actividad Moodle, Medidas sociales	suspende / aprueba	20	Acc, f-Measure
[118]	2013	146	Rendimiento en otros cursos	bajo / medio / alto / muy alto	1	Acc., TP, TN
[81]	2013	114	Actividad en Moodle	bajo / medio / alto / muy alto	6	sop., conf., lift, cov., t.
[133, 134]	2014	123	Atributos extraídos sobre comentarios usando text mining	bajo / medio / alto / muy alto	2	Acc, f-Measure
[132]	2014	230	Actividad Moodle	no presentado / suspenso / bien / muy bien	5	sop., conf., t, nº reglas
[119]	2015	¿100	Demográficos, otros	suspende / aprueba, nota numérica	varias de regresión y clasificacion	-
[142]	2015	114	Cinco dimensiones de actividad extraídas con LA	bajo / medio / alto / muy alto	6	Acc., TP, TN, fitness

predicción del rendimiento de estudiantes con técnicas de minería de datos se concluía lo mismo [87, 89], los cierto es que estos estudios carecían aún de una extensa comparativa de algoritmos que permitiese afirmar o negar esta hipótesis. De hecho, a medida que han ido apareciendo más trabajos, lo que ha podido observarse es que, independientemente de que se trabaje con conjuntos datos pequeños, no existe un sólo clasificador que sea mejor que el resto en todos los casos [124–126] y que técnicas como, por ejemplo, los árboles de decisión [137] o basadas en Multi-instancia [127, 128, 131] pueden resultar ser las que mejor rendimiento obtengan en determinados casos.

Otra cuestión que ha de ser resaltada es el escaso número de cursos, y por tanto de conjuntos de datos, que se utilizan en los trabajos, debido probablemente a la imposibilidad de los autores de acceder a diferentes fuentes de datos que no sean las de su propio entorno de trabajo, siendo así que en la mayor parte de los estudios los conjuntos de datos utilizados provienen de cursos impartidos en universidades en los que los propios autores realizan su actividad docente. En los estudios con un mayor número de clasificadores utilizados ([124–128, 131, 135] entre otros) los autores sólo utilizan los datos de actividad de estudiantes en 7 cursos diferentes de un grado en Ingeniería Informática de una universidad española. Este hecho puede provocar que los modelos de predicción descritos en los trabajos estén excesivamente vinculados a la estructura, el área de conocimiento o los procesos de enseñanza-aprendizaje seguidos en esos cursos. La ausencia de trabajos en los que se utilicen un amplio número de cursos con diferentes características entre sí hace difícil vislumbrar hasta que punto las conclusiones extraídas en la literatura son extrapolables a otros estudios en los que se utilicen cursos diferentes. Debido a ello, existe la necesidad de realizar nuevos estudios en los que se utilicen conjuntos de datos de cursos de diferente paradigma y características, con objeto de evaluar los modelos de predicción del rendimiento de los estudiantes de una forma más amplia y completa.

Por último, es de destacar la preocupación existente, por parte de muchos autores, por ofrecer modelos de predicción que sean sencillos de entender, de cara a que los usuarios finales, profesores de cursos que no han de tener conocimientos sobre minería de datos, puedan interpretarlos y utilizarlos en su beneficio [124, 142]. Es de resaltar que la complejidad a la hora de obtener modelos de predicción del rendimiento de los estudiantes no se encuentra sólo en la interpretación de los modelos, sino en el proceso KDD al completo y, sobretodo, en el proceso de selección y configuración de los algoritmos de clasificación, que requiere por parte del usuario un conocimiento acerca de

su funcionamiento. Por tanto, no es cuestión solamente de que los profesores obtengan modelos de predicción sencillos de entender, sino de cómo pueden obtener modelos con buen rendimiento automáticamente, sin la necesidad de intervenir en ese proceso.

Capítulo 4

Meta-learning: en busca del mejor clasificador

Una de las tareas principales en la construcción de modelos de predicción consiste en escoger el algoritmo de clasificación que sea capaz de obtener el mejor modelo, esto es, que sea más robusto y preciso que el resto de los clasificadores. En el contexto educativo, el mayor problema se encuentra en que, al utilizar conjuntos de datos de cursos virtuales completamente diferentes, con distintas estructuras, contenidos, perfiles de estudiantes y organización del proceso de enseñanza-aprendizaje, el algoritmo de clasificación para obtener el mejor modelo de predicción puede variar, y de hecho varía enormemente [26].

Tal y como sostienen en [12], "Las técnicas de minería de datos están enfocadas más a la potencia y la flexibilidad que a la simplicidad". Por ello, otro de los grandes problemas en este tema de investigación, también debido a la gran cantidad de algoritmos de clasificación existentes y sus casi infinitas posibilidades de configuración, es que se hace inaccesible para un usuario no experto, como es el caso de los profesores de cursos virtuales, el poder utilizar estos algoritmos para obtener información sobre, por ejemplo, la relación entre la actividad de sus estudiantes y su rendimiento.

Como consecuencia, existe la necesidad en el área EDM de estudiar cómo las meta-características de los diferentes cursos y de los conjuntos de datos influyen en los resultados de los algoritmos de clasificación. De forma más concreta, y en atención a los objetivos mencionados en el apartado 1.3, en el presente capítulo se muestran los resultados y conclusiones de un conjunto de los estudios destinados a establecer la influencia

de estas meta-características en el rendimiento de los algoritmos de clasificación, de cara a poder finalmente desarrollar un sistema automático de recomendación de clasificadores con objeto de que los profesores de cursos virtuales puedan obtener modelos predictivos sin necesidad de seleccionar la técnica por ellos mismos, y tratando de garantizar a su vez que el algoritmo de clasificación utilizado retorna un modelo preciso. Esto es, el presente capítulo se encuadra dentro del área del meta-learning [143] y, más concretamente dentro de este área, la línea de trabajo de los estudios está enfocada en la selección automática de clasificadores.

El presente capítulo se estructura de la siguiente forma: en el apartado 4.1 se introduce el área de meta-learning y se muestra un resumen de los trabajos más relevantes enfocados en la recomendación automática de clasificadores. El apartado 4.2 recoge la organización y un resumen de la configuración de los diferentes estudios mostrados, estableciendo las hipótesis de las que parten y que se pretenden demostrar o refutar con los resultados obtenidos en cada uno de ellos. En el apartado 4.3 se explica el significado de las meta-características utilizadas en los diferentes estudios. El apartado 4.4 se divide en cinco secciones, en cada una de las cuales se muestran la configuración, un resumen de las meta-características de los conjuntos de datos, el proceso seguido, los resultados y las conclusiones de cada uno de los experimentos más relevantes llevados a cabo. Finalmente, en el apartado 4.5 se establecen las conclusiones finales con los resultados de todos los estudios.

4.1. Estado del arte: meta-learning y la predicción de rendimiento

En su trabajo, Vilalta et al. [143], uno de los pioneros en este campo, definieron el área de meta-learning de la siguiente manera: *"Metalearning studies how learning systems can increase in efficiency through experience; the goal is to understand how learning itself can become flexible according to the domain or task under study"*. Existen diferentes procesos englobados dentro del término meta-learning. Lemke et al. [144] publicaron en 2013 un trabajo en el que tratan de definir las distintas ramas que conforman el área del meta-learning, como por ejemplo la aplicación de métodos de ensamblado y combinación de clasificadores base, del que son ejemplo *Stacked Generalisation* [145] y

Cascade Generalisation [146]; o la recomendación de algoritmos de aprendizaje, entre otros. Con respecto a la recomendación, o selección, de algoritmos de aprendizaje, ya en 1975, John R. Rice apuntó a este problema en su trabajo *"The Algorithm Selection Problem"* [147]. En él, el autor estableció "la dificultad que existe para escoger el mejor algoritmo de aprendizaje, dada la gran variedad de situaciones que pueden darse en diferentes contextos".

Los procesos de meta-learning han sido utilizados ampliamente en el área de la minería de datos con objeto de seleccionar el mejor algoritmo para modelar los datos, siendo la selección de clasificadores una de las tareas más estudiadas en este área [143]. Michie et. at, en su libro de 1994 [148], introdujeron algunos conceptos acerca de esta problemática, y propusieron un conjunto de medidas o meta-características que sirviesen para caracterizar los conjuntos de datos de cara a poder determinar, junto con los resultados de aplicar diferentes algoritmos de clasificación sobre estos conjuntos de datos, cuáles son las técnicas de clasificación más recomendadas para un problema concreto. En este trabajo, los autores establecieron 3 grupos de meta-características de utilidad para caracterizar los conjuntos de datos: simples (p.e. número de instancias o de atributos), estadísticas (p.e. kurtosis medio de los atributos) y basadas en teoría de la información (p.e. la entropía de la clase).

Las meta-características propuestas por Michie et al. han servido de base para posteriores trabajos en el campo de meta-learning. En King et al. [149], los autores publicaron los resultados del que fue el primer gran proyecto europeo de investigación en el que se aplicaban procesos de meta-learning para la selección clasificadores, denominado STATLOG. Aplicando diferentes algoritmos de clasificación sobre distintos conjuntos de datos, y utilizando un conjunto de las meta-características propuestas por Michie et al. [148], concluyeron que no existe un mejor algoritmo para todos los casos, y que la selección de uno u otro depende de las meta-características de los conjuntos de datos.

Son varios los procesos de meta-learning empleados para seleccionar o recomendar algoritmos de clasificación. En muchos de los trabajos que se pueden encontrar en la literatura, se utilizaron algoritmos de clasificación como meta-predictores o recomendadores. En este tipo de proceso de meta-learning, se tiene un meta-conjunto de datos, en el que cada fila contiene las meta-características de cada conjunto de datos incluido en el estudio como atributos predictores, y el algoritmo que mejor rendimiento obtuvo

para ese conjunto como clase a predecir [150–154]. Diferentes meta-características han sido utilizadas para determinar cuál es el clasificador con mejor rendimiento aunque, tal como se puede observar en las Tablas resumen 4.1 y 4.2, el más común en todos los trabajos es el *accuracy*. En su trabajo, Linder [150] apuntó a que, si bien el *accuracy* es una medida aceptable en el proceso de selección de algoritmos, han de tenerse en cuenta otros factores, como el costo computacional o la interpretabilidad de los modelos.

En otros trabajos, los meta-modelos construidos no devuelven como salida al mejor algoritmo de clasificación, sino un ranking de clasificadores según el rendimiento esperado de estos en base a las meta-características de los conjuntos de datos. Una de las estrategias consiste en la aplicación del algoritmo *Nearest Neighbourds* [155–158] para determinar qué conjuntos de datos investigados previamente tienen unas meta-características similares a las del conjunto de datos sobre el que queremos una recomendación, y finalmente construir un ranking en base al rendimiento medio de cada uno de los clasificadores sobre estos conjuntos de datos. Otras técnicas de ranking se basan en la construcción de varios meta-modelos de regresión, uno por cada clasificador utilizado en el estudio, utilizando como atributos predictores las meta-características de los conjuntos de datos y, como atributo a predecir el rendimiento del clasificador para cada conjunto en valor numérico (como puede ser, por ejemplo, el *accuracy*) [159]. Al llegar un nuevo conjunto de datos, estos meta-regresores pueden ser utilizados para predecir qué rendimiento tendrá cada clasificador sobre él.

En otros estudios, Kalousis et al. [160–162] construyeron el ranking mediante una comparativa por pares de clasificadores. Por otro lado, en Zeroski et al. [163, 164] propusieron una nueva forma de ranking basada en *Clustering Trees* como alternativa a los rankings generados con *Nearest Neighbourds*. Este método consiste en generar un meta-modelo en forma de árbol de decisión, en el que los atributos predictores son las meta-características de los conjuntos, y en las hojas del árbol se predice no el mejor clasificador, si no el orden de los clasificadores, de mayor a menor rendimiento. Los autores concluyeron que la mayor ventaja de su propuesta radicaba en que, con el árbol de decisión, el usuario puede entender el motivo por el cuál los clasificadores son ordenados en el ranking.

Otra forma en la que el problema de recomendación de algoritmos se ha abordado es haciendo uso de algoritmos multi-etiqueta: en vez de realizar un ranking o recomendar sólo el mejor algoritmo, el problema puede modelarse de forma que, para cada conjunto

de datos, no haya un sólo clasificador recomendado, sino varios, tal y como se propone en un trabajo reciente, publicado en 2014 [165], en el que los autores utilizaron el algoritmo *MLkNN* [166].

Si bien las meta-características simples, estadísticas y de teoría de la información son las más comúnmente empleadas, no son las únicas. En [160, 161] Kalousis et al. propusieron utilizar meta-características a nivel de cada atributo, en vez de extraer únicamente la media global de estas estadísticas en el conjunto de datos. El autor principal de estos trabajos realizó una extensión de los mismos en [162], denotando la utilidad de las técnicas selección de atributos para mejorar la meta-predicción.

En [167] se ahondó en un nuevo concepto, ya mencionado en Michie et al.[148], para la extracción de nuevas meta-características que, desde en entonces, han sido incluidas en numerosos estudios: el *landmarking*. El *landmarking*, en el área de meta-learning, consiste en extraer el rendimiento de clasificadores denominados "con poco coste computacional" y en utilizar este rendimiento como atributo predictor. A estos atributos predictores se les denomina *landmarkers*. Usualmente, estos *landmarkers* representan el rendimiento del clasificador en base a su *accuracy* aunque, no obstante, existen trabajos en los que se proponen otras posibilidades [168]. En Besusan et al. [169], los autores afirmaron, en base a su experimentación, que "los *landmarkers* superan, como meta-características predictoras, a las tradicionales variables estadísticas usadas en el campo del meta-learning".

Recientemente, las meta-características de complejidad han comenzado a cobrar relevancia en los estudios de meta-learning. Como su mismo nombre indica, definen la complejidad de un conjunto de datos en base a meta-características tales como la "no linearidad"de un clasificador lineal. Estas meta-características pueden obtenerse utilizando la herramienta DCol [170]. Existen diversos estudios en los que se analizan estas meta-características de complejidad para determinar la idoneidad de un clasificador sobre un conjunto de datos [171].

Otras meta-características que muestran su utilidad son las basadas en modelos [156]. Estas meta-características se basan en la estructura de un modelo de clasificación construido sobre los conjuntos de datos, usualmente un árbol de decisión, del que se extraen meta-características como la profundidad, el número de hojas, o el número de atributos que utiliza.

Lee et al. [172] adviertieron en su trabajo de la necesidad de estudiar el coste computacional que supone el calcular las meta-características, con objeto de constatar si este es notablemente menor que el coste de ejecutar todos los clasificadores para determinar cuál es el mejor.

El proceso de meta-learning no se ha aplicado con el único objetivo de seleccionar el mejor algoritmo de aprendizaje para una tarea determinada, sino que también se han publicado trabajos en los que el objetivo es seleccionar los parámetros de inicialización adecuados para un determinado algoritmo. Este tipo de estudios tienen una amplia aplicación en torno a la configuración de los clasificadores [173–178]. También el proceso meta-learning tiene su aplicación para tratar de seleccionar el mejor algoritmo de *clustering* [179], si un árbol de decisión debería ser o no podado [180, 181], para predecir el valor de alguna medida de rendimiento de uno o varios clasificadores mediante regresión [151, 172, 182–186], o en presencia de conjuntos de datos de series temporales [182–184].

Algunos autores han llegado a desarrollar componentes o librerías software de meta-learning para la recomendación de clasificadores, con objeto de ponerlos al alcance de los usuarios. Una de estas primeras librerías fue WekaMetal, derivada del proyecto METAL (sitio Web sin mantenimiento), acrónimo de ***Met****a-Learning* ***A****ssistant for Providing User Support in Machine* ***L****earning Mining*. Como módulos de la herramienta de minería de datos llamada RapidMiner [187], caben destacar el sistema propuesto en [185], en el que se desarrolló un módulo de recomendación de clasificadores con *landmarkers*, y el trabajo ya mencionado de Reif. et al. [159] en el que los autores diseñaron un sistema que retorna al usuario rankings de clasificadores.

En conclusión, diversos autores sostienen en sus trabajos que no existe un único algoritmo de clasificación que sea mejor que el resto en todas las tareas o contextos (*No free lunch theorem* [26, 190]). En la literatura, existen diversos trabajos entorno a este problema, en los que se trata de definir qué meta-características de los conjuntos de datos determinan cuándo se debe utilizar un clasificador u otro para obtener los mejores resultados posibles. La selección de estos clasificadores se puede realizar en base a diferentes tipos de meta-características (simples, estadísticas, teoría de la información, de complejidad, *landmarkers*, etc.) y en base a diferentes enfoques (seleccionar el mejor clasificador, crear un ranking, etc.). En el área educativa, la afirmación de que no existe un clasificador mejor que otros para predecir el rendimiento de los estudiantes en todos

TABLA 4.1: Trabajos relacionados sobre la selección de clasificadores con meta-learning (1)

Referencia	Año	Nº*	Filosofía	Medida de rendimiento	Tipos de Características
[149]	1995	17	Mejor técnica	Accuracy, tiempo de ejecución	Simples, Estadísticas
[150]	1999	15	Mejor técnica	Accuracy, tiempo de entrenamiento, tiempo de Test, interpretabilidad	Simples, Estadísticas, Teoría de la información
[151]	2000	3	Mejor técnica y regresión (comparativa)	Ratio de Error	
[155]	2000	6	Ranking	Adjusted Ratio of Ratios	Simples, Estadísticas, Teoría de la información
[167]	2000	4	Mejor técnica comparando por pares de clasificadores	Error predictivo	Simples, Estadísticas, Landmarkers
[169]	2000	11	Utilidad de landmarkers vs teoría de la información	Accuracy	Landmarkers, Teoría de la informacion
[188]	2001		Regresión (comparan con técnicas de ranking)	Accuracy	
[160][161][162]	2001	8	Ranking comparando por pares de clasificadores	Accuracy	Simples, Estadísticas, Teoría de la información
[168]	2001	5	Propuesta de nuevos landmarkers	Accuracy	Landmarkers
[152]	2001	6	Mejor técnica con redes neuronales	Error medio	Simples, Estadísticas, Teoría de la información
[163, 164]	2002	10	Ranking con árboles de decisión y clustering	Adjusted Ratio of Ratios	Simples, Estadísticas, Teoría de la información
[156, 157]	2002	11	Ranking con k-NN	Adjusted Ratio of Ratios	Simples, Estadísticas, Teoría de la información
[189]	2002	6	Clustering para caracterizar las meta-características		Simples, Estadísticas, Teoría de la información, Landmarkers, Basado en modelos

* Número de clasificadores a predecir

Tabla 4.2: Trabajos relacionados sobre la selección de clasificadores con meta-learning (2)

Referencia	Año	Nº*	Filosofía	Medida de rendimiento	Tipos de Características
[158]	2003	10	Ranking con k-NN	Adjusted Ratio of Ratios	Simples, Estadísticas, Teoría de la información
[154]	2006	8	Mejor técnica	TPRate, TNrate, f-Measure, accuracy	Simples, Estadísticas, Teoría de la información
[172]	2008	7	Predicción del accuracy	Accuracy	Simples, Estadísticas, Teoría de la información
[182–184]	2007,2007,2008	1***	Rendimiento de MLP	RMSE	Simples, Teoría de la Información
[180]	2009	1**	Mejor DT según el podado	Accuracy	Teoría de la Información
[153]	2009	7	Ranking con k-NN	Accuracy, Tiempo de ejecución, Ocupación en memoria	Simples, Estadísticas, Teoría de la información, Landmarkers
[185]	2010	7	Regresión y comparativa de tipos de meta-características	Accuracy	Simples, Estadísticas, Teoría de la información
[186]	2011	1	Rendimiento de MLP, Active Learning	RMSE	Simples, Teoría de la Información
[181]	2009,2011	1**	Mejor DT según el podado, Active Learning	Accuracy	Teoría de la Información
[159]	2014	9	Ranking con regresión	Accuracy	Simples, Estadísticas, Teoría de la información, Landmarkers, Basado en modelos
[165]	2014	14	Multilabel-learning	Accuracy, Adjusted Ratio of Ratios	Simples, Estadísticas, Teoría de la información, Landmarks, Basado en modelos

* Número de clasificadores a predecir

** Decission Tree con distinas "podas"

*** Multilayer Perceptron y otros de series temporales

los casos también se muestra acertada, tal y como se concluye en el capítulo 3. No obstante, y pese a la preocupación existente en generar modelos de predicción certeros que puedan ser obtenidos y utilizados por profesores de cursos virtuales, en la actualidad no existen apenas trabajos en este área, al margen de los surgidos del desarrollo de esta tesis, en los que se apliquen técnicas de meta-learning exclusivamente sobre este tipo de datos y con objeto de seleccionar automáticamente estos algoritmos.

4.2. Hipótesis de partida, organización y resumen de los estudios

El trabajo de investigación que componen los estudios incluidos en este capítulo tiene como última finalidad construir un sistema recomendador que, de forma automática, seleccione y recomiende aquellos algoritmos de clasificación que se espera retornen los mejores modelos de predicción del rendimiento de los estudiantes en cursos virtuales. La hipótesis en este capítulo será verificada si: (1) el clasificador seleccionado por este recomendador se encuentra entre los que obtienen los modelos con mejor rendimiento, y (2) el modelo obtenido es sencillo de interpretar para un usuario no experto.

Para validar esta hipótesis, en este capítulo se recogen los resultados de cinco de los estudios más definitorios realizados en esta investigación, que sirvan para comprender el proceso seguido y valorar los resultados y conclusiones del mismo. Estos estudios son los que se describen a continuación:

1. **Estudio 1. Primeros pasos: comparativa de tipos de clasificadores para la predicción del rendimiento**

   En este primer estudio, mostrado en la sección 4.4.1, se realizó una comparativa de 5 algoritmos de clasificación que predijesen el rendimiento de los estudiantes sobre 3 conjuntos de datos diferentes, con objeto de determinar si las meta-características simples de los conjuntos de datos, como el número de instancias, afectan a la calidad de los modelos predictivos. Se concluyó, al igual que ya se constató en la literatura expuesta en los apartados 3.2 y 4.1 de este documento de tesis, la no existencia de un algoritmo que sea mejor que el resto en todos los casos, y que las meta-características de los conjuntos de datos afectan a su rendimiento.

2. **Estudio 2. Las meta-características simples como predictores**

   En este estudio, redactado en la sección 4.4.2, se planteó si las meta-características simples son útiles, por sí solas, para predecir cuál será el clasificador que obtendrá un mejor modelo de predicción. El motivo de utilizar este tipo de meta-características fue el bajo coste computacional de su cálculo, en comparación con otras. Se siguió, por tanto, una estrategia de meta-learning en la que se tiene un conjunto de meta-características simples de los conjuntos de datos utilizados en la experimentación, y cuya clase a predecir es el nombre del clasificador que tuvo un mejor rendimiento en cada conjunto, tomando como medida de rendimiento de los modelos obtenidos el *accuracy*.

   La utilidad de cada una de las meta-características fue evaluada por varios algoritmos de selección de atributos, mostrándose en este apartado los resultados obtenidos por la función *ClassifierSubSetEval* de Weka tomando como base dos algoritmos de diferente paradigma, C4.5 (versión J48 de Weka) y NaïveBayes.

   En este estudio, se constató que las meta-características simples tienen bastante utilidad de cara a realizar la tarea pero que, no obstante, y debido a la escasez de conjuntos de datos con la actividad de los estudiantes en cursos alojados en plataformas LCMS, el recomendador sólo es eficiente cuando se trata de predecir sobre un conjunto de clasificadores reducidos, volviéndose poco eficiente cuando aumentamos su número.

3. **Estudio 3. Construcción de recomendadores a partir de las meta-características simples**

   En este estudio, redactado en la sección 4.4.3, se continuó el estudio anterior, aumentando ligeramente el número de conjuntos de datos y modificando las meta-características de los mismos, creando así conjuntos de datos artificiales a partir de los originales. Para ello, se les aplicaron técnicas de discretizado, rebalanceo de la clase o se les añadieron valores nulos, teniendo así nuevos conjuntos de datos con meta-características diferentes. Se construyeron dos recomendadores, generados con el algoritmo J48, utilizando para ello respectivamente como clase 12 y 4 clasificadores a predecir.

   Los recomendadores fueron probados en base a un conjunto de instancias de prueba previamente seleccionadas y con meta-características diferentes entre sí, con

objeto de comprobar si el conjunto de datos acierta a recomendar el mejor clasificador, o uno cercano al mejor.

Pese a haber aumentado el conjunto de datos y refinado la aplicación de técnicas de pre-procesado para la generación de conjuntos artificiales, las conclusiones extraídas fueron similares a las del experimento anterior, y es que si bien el recomendador construido obtenía un buen rendimiento cuando el conjunto de clasificadores a predecir era reducido, no alcanzaba tan buenos resultados cuando se aumentaba su número.

4. **Estudio 4. Más allá de las meta-características simples: las meta-características de complejidad y de "contexto"**

   Este estudio está contenido en la sección 4.4.4. Se comprobó la utilidad de las meta-características de complejidad para construir los recomendadores, así como un nuevo tipo de meta-características propuestas, las de "contexto", que describen si el curso es *e-learning* o *blended*, y si es específico o transversal. La motivación de utilizar y analizar este tipo de meta-características, previo paso a añadir otros tipos de meta-características, vino dada por el hecho de que, hasta la fecha de finalización de esta tesis, las meta-características de complejidad y de "contexto"han sido poco utilizadas en la literatura, algo que se ha podido constatar en el apartado 4.1.

   Al igual que en los anteriores estudios, se crearon conjuntos de datos artificiales, en este caso únicamente aplicando un proceso de discretización para obtener conjuntos con atributos puramente nominales, ya que los originales únicamente contenían atributos numéricos que medían la actividad de los estudiantes.

   En este estudio, se concluyó que tanto las meta-características de complejidad como de contexto muestran ser bastante útiles de cara a construir recomendadores de clasificadores. En algunos casos, estas meta-características llegaron a tener una relevancia mayor que las simples.

5. **Estudio 5. Cambiando el enfoque del proceso de meta-learning: ranking con regresión.**

   Uno de los principales problemas encontrados en el área del meta-learning es la baja cantidad de conjuntos de datos disponibles para realizar los estudios. Este hecho limitó de forma notable el rendimiento de los recomendadores mencionados

en los estudios anteriores dado que, con tan pocos conjuntos, se hace difícil generar un meta-modelo de clasificación robusto que trate de recomendar el mejor clasificador cuando el conjunto de clasificadores utilizados es alto.

El objetivo en el estudio, recogido en la sección 4.4.5, fue por tanto hallar una forma de generar un sistema recomendador robusto con pocos conjuntos de datos. En vez de tratar de predecir el mejor clasificador, se construyó un sistema recomendador basado en un conjunto de meta-regresores lineales que predicen, para cada clasificador, el *accuracy* que obtendría al aplicarse sobre un conjunto de datos concreto. El sistema genera un ranking, y el clasificador en la cima de ese ranking, esto es, con un *accuracy* predicho más alto, sería el recomendado para realizar la tarea de predicción del rendimiento de los estudiantes. Se añadieron, además de las meta-características simples y de complejidad ya utilizadas en anteriores estudios, meta-características estadísticas y *landmarkers*.

Este nuevo sistema recomendador demostró, en los resultados, ser mucho más eficiente que los recomendadores basados en meta-clasificación. Más aún, aunque obviamente un mayor conjunto de datos podría proporcionar un recomendador más refinado y robusto, el sistema mostró ser eficiente con tan sólo utilizar los 30 conjuntos de datos originales que se tenían al inicio de la experimentación, sin necesidad de crear conjuntos artificiales.

4.3. Meta-características de los conjuntos de datos utilizadas en los estudios

En las Tablas 4.3, 4.4, 4.5, y 4.6 se incluyen las meta-características utilizadas en cada uno de los estudios que se muestran en este capítulo. Las columnas primera y segunda contienen, respectivamente, el nombre y tipo de la meta-característica. En la tercera columna se expone una explicación de su significado. En la cuarta columna se indica el software o librería utilizados para el cálculo de estas meta-características, indicando en su caso con la etiqueta "propio"que la meta-característica se extrajo mediante un método implementado *ad-hoc*. Por último, en la quinta columna, se informa de los estudios en los que la meta-característica fue utilizada. Una descripción detallada de las meta-características de complejidad puede encontrarse en [191] y [192].

Además de los indicados en las tablas, se utilizaron en los estudios 2 y 3 el valor de entropía de Shannon de la clase como meta-característica de teoría de la información, discretizado en 3 valores: clase balanceada, clase ligeramente desbalanceada y clase muy desbalanceada. También, en el estudio 4, se añadieron otras dos meta-características de los conjuntos de datos que definen el contexto de los mismos de dos formas diferentes: una de ellas indica si los datos provienen de un curso semipresencial (*blended*) o completamente virtual (*e-learning*), mientras que la otra contiene información acerca de si el curso es de carácter transversal, esto es, abierto a estudiantes de cualquier tipo de área de conocimiento, o por el contrario, el curso está enfocado a estudiantes de un área de conocimiento en concreto.

Tabla 4.3: Meta-características simples utilizadas en los estudios

Nombre	Tipo	Descripción	Software	Estudios
N# de ins.	Simple	Nº de instancias o filas	Propio	2,3,4,5
N# de att.	Simple	Nº de atributos	Propio	2,3,4,5
N# de att. Nom.	Simple	Nº de atributos Nominales	Propio	2,3
N# de att. Núm.	Simple	Nº de atributos Numéricos	Propio	2,3
Tipo de att.	Simple	Indica si sólo hay att. Núm., att. Nom., o ambos	Propio	2,4
Completitud	Simple	Porcentaje de valores no nulos	Propio	2,3
N# de clases	Simple	N# valores en la clase a predecir	Propio	2,3
Dimensionalidad	Simple	(N# de att./N# de ins.)	Propio	5
Inv. dimensionalidad	Simple	Inverso de la dimensionalidad	Propio	5

Tabla 4.4: Meta-características estadísticas utilizadas en los estudios

Nombre	Tipo	Descripción	Software	Estudios
Skewness medio	Estadística	Skewness medio de los atributos	MATH3-Apache [193]	5
Skewness máx.	Estadística	Skewness máximo de los atributos	MATH3-Apache	5
Skewness min.	Estadística	Skewness mínimo de los atributos	MATH3-Apache	5
Kurtosis medio	Estadística	Kurtosis medio de los atributos	MATH3-Apache	5
Kurtosis máx.	Estadística	Kurtosis máximo de los atributos	MATH3-Apache	5
Kurtosis min.	Estadística	Kurtosis mínimo de los atributos	MATH3-Apache	5

Tabla 4.5: Meta-características de complejidad utilizadas en los estudios

Nombre	Tipo	Descripción	Software	Estudios
F1	Complejidad	*The maximum Fisher's discriminant ratio*	DCol	4,5
F1v	Complejidad	*The directional-vector maximum Fisher's discriminant ratio*	DCol	5
F2	Complejidad	*The overlap of the per-class bounding boxes*	DCol	4,5
F3	Complejidad	*The maximum (individual) feature efficiency*	DCol	4,5
F4	Complejidad	*The collective feature efficiency*	DCol	4,5
L1	Complejidad	*The leave-one-out error rate of the one-nearest neighbor classifier*	DCol	4,5
L2	Complejidad	*The minimized sum of the error distance of a linear classifier*	DCol	4,5
N1	Complejidad	*The fraction of points on the class boundary*	DCol	4,5
N2	Complejidad	*The ratio of average intra/inter class nearest neighbor distance*	DCol	4,5
N3	Complejidad	*The training error of a linear classifier*	DCol	4,5
L3	Complejidad	*The nonlinearity of a linear classifier*	DCol	4,5
T1	Complejidad	*The fraction of maximum covering spheres*	DCol	4,5
T2	Complejidad	*The average number of points per dimension*	DCol	4,5

Tabla 4.6: Meta-características landmarkers utilizadas en los estudios

Nombre	Tipo	Descripción	Software	Estudios
LD	Landmark	Accuracy del clasificador Linear Discriminant	Rapidminer	5
NB	Landmark	Accuracy del clasificador Naïve-Bayes	Weka	5
BN	Landmark	Accuracy del clasificador Best Node (J48 con un sólo nodo para discriminar)	Rapidminer	5
RN	Landmark	Accuracy del clasificador Random Node (Random Tree con un sólo nodo para discriminar)	Weka	5
1NN	Landmark	Accuracy del clasificador Nearest Neighbourds con k=1	Rapidminer	5

4.4. Configuración, proceso, resultados y conclusiones de los estudios

El presente apartado contiene la configuración, resultados y conclusiones de los cinco estudios mencionados en el apartado 4.2. Todos los estudios se estructuran en tres bloques:

- **I. Configuración del estudio:** se detallan los conjuntos de datos, algoritmos, configuraciones y resto de componentes del estudio, así como los procesos llevados a cabo.
- **II. Resultados del estudio:** contiene los resultados del estudio, siguiendo la configuración y procesos indicados en el bloque I.
- **III. Conclusiones del estudio:** se exponen las conclusiones sobre los resultados obtenidos en el bloque II.

4.4.1. Estudio 1. Primeros pasos: comparativa de tipos de clasificadores para la predicción del rendimiento

En este primer estudio, se realizó una comparativa de diferentes clasificadores aplicados sobre tres conjuntos de datos con la actividad y rendimiento de los estudiantes de un curso virtual. El objetivo de este estudio fue determinar si, tal y como indica el teorema *no free lunch*, no existe un sólo clasificador que tenga un mejor rendimiento que el resto en todos los casos.

Configuración del estudio

El proceso seguido en este estudio se estructuró en 3 tareas que se detallan a continuación:

1. Extracción de los datos de los cursos y generación de conjuntos de datos

Se utilizaron 3 conjuntos de datos diferentes, denominados en ese estudio c1, c2 y c3. Los conjuntos de datos c1 y c3 provinieron del curso virtual con identificador 3, mientras que el conjunto c2 almacenaba la información de los estudiantes de los cursos 1, 2 y 3 combinados. Todos ellos contenían 5 atributos que miden la actividad de los estudiantes según el tiempo dedicado total y medio por semana, número total y medio de sesiones dedicadas por semana, y tiempo medio dedicado por sesión. Además, el conjunto c3 contenía cinco atributos discretos adicionales. Cuatro de estos atributos indicaban los estilos de aprendizaje de los estudiantes extraídos mediante una encuesta realizada por el propio profesor. Cada uno de estos estilos de aprendizaje tomaba uno de los siguientes valores: Activo o Reflexivo, Sensitivo o Intuitivo, Visual o Verbal, y Secuencial o Global. El último de estos atributos discretos indicaba el grado o carrera en la que el estudiante estaba matriculado.

2. Aplicación de 5 algoritmos de clasificación sobre los 3 conjuntos de datos

Dado que uno de los objetivos de esta tesis es poder mostrar al usuario no experto modelos de predicción que pueda interpretar, en este estudio se excluyeron métodos como son las Redes Neuronales o SVM debido a que sus modelos predictivos no son sencillos de interpretar para un usuario no experto, optando finalmente por modelos en forma de reglas (con NNge y OneR), en forma de árbol de decisión (J48) o estadísticos (NaïveBayes y BayesNetwork). Aunque NaïveBayes y BayesNetwork pueden mostrar modelos que no siempre sean sencillos de interpretar, se incluyeron en la experimentación

debido a que, como se vio en la sección 3.2, los algoritmos bayesianos suelen obtener un buen rendimiento con este tipo de conjuntos de datos.

3. Evaluación de los modelos obtenidos

Con objeto de evaluar cada modelo de predicción, estos se validaron sobre los conjuntos de datos utilizando 10-CV, repitiendo el proceso 10 veces por cada conjunto con diferentes subconjuntos de entrenamiento y prueba, lo que se traduce en 100 ejecuciones de cada clasificador por cada conjunto de datos, y un total de 1500 ejecuciones (3 conjuntos * 5 clasificadores * 10 CV * 10 ejecuciones).

De cada ejecución, se almacenaron el *accuracy*, *TPrate*, *TNrate*, *FPrate* y *FNrate* obtenidos, que fueron objeto de comparativa en este estudio. El motivo de utilizar el *accuracy* como medida de rendimiento fue, en parte, el mismo por el cual se utilizaron modelos de predicción sencillos de entender, y es que el concepto de *accuracy* es asequible para cualquier usuario no experto, mientras que otras medidas de rendimiento, como la *f-Measure*, son más complejas de entender. Además, tal y como se ha podido comprobar en la literatura, la mayoría de los trabajos en este área contemplan el *accuracy* como la medida de rendimiento con la que comparar sus resultados, por lo que se hace necesario utilizarla en este trabajo para contrastar. Además, se añadieron las otras cuatro medidas debido a que también son sencillas de entender, y para servir de apoyo a las conclusiones al respecto del rendimiento de los clasificadores.

Sobre los resultados obtenidos, se aplicó un test de significancia *two-side paired t-test*, utilizando como base el rendimiento del algoritmo OneR, que es el que obtuvo, en media, peor rendimiento en las pruebas.

Resultados del estudio

Los resultados obtenidos son mostrados en la Tabla 4.7. Como puede observarse, NaïveBayes (NB) es el algoritmo que obtiene un mejor rendimiento con los conjuntos de datos c1 y c3. En el primero de los conjuntos de datos, además este rendimiento es seguido muy de cerca por el obtenido con BayesNetwork (BN). De hecho, aunque NaïveBayes obtiene en este caso un mejor *accuracy* y *TPrate*, 77.29 % y 77 % frente a 76.36 % y 76 % de BayesNetwork, es el algoritmo BayesNetwork el que supera a NaïveBayes en *TNrate*, con un 82 % frente a un 75 % de NaïveBayes. En ambos conjuntos de datos, J48 tiene un buen rendimiento aunque, no obstante, queda lejos del obtenido por NaïveBayes. En el caso del conjunto de datos c2 es BayesNetwork el algoritmo con mejor rendimiento,

con una diferencia notable con respecto al resto de clasificadores. No obstante, en este mismo caso es NaïveBayes el algoritmo que obtiene un peor rendimiento, incluso que OneR. Por otra parte, J48 es el algoritmo con el segundo mejor rendimiento, algo que también sucede con el conjunto de datos c3, en dónde incluso el *t-test* le marca como uno de los algoritmos cuya mejora es más significativa (v).

Por otro lado, también podemos observar que el uso de atributos predictores que contienen las medidas de actividad de los estudiantes es útil para obtener modelos de clasificación fiables, con un rango de *accuracy* de entre 65.79 % y 81.16 %.

Tabla 4.7: Resultados obtenidos en los 3 conjuntos del estudio 1

Conjunto	Clasificador	TPrate	FPrate	TNrate	FNrate	Accuracy
	OneR	0.66	0.47	0.53	0.34	65.79
	J48	0.74v	0.17	0.83v	0.26	74.21v
c1	NB	0.77v	0.25	0.75	0.23	77.29v
	BN	0.76v	0.18	0.82v	0.24	76.36v
	NNge	0.70	0.41	0.59	0.30	70.10
	OneR	0.78	0.20	0.80	0.22	77.86
	J48	0.79	0.17	0.83	0.21	79.36
c2	NB	0.76	0.27v	0.73	0.24	76.40
	BN	0.81v	0.15	0.85v	0.19	81.16v
	NNge	0.78	0.22	0.78	0.22	78.04
	OneR	0.65	0.48	0.52	0.35	65.29
	J48	0.76v	0.23	0.77v	0.24	75.83v
c3	NB	0.81v	0.20	0.80v	0.19	80.90v
	BN	0.73	0.29	0.71	0.27	73.36
	NNge	0.75	0.36	0.64	0.25	74.98

Conclusiones del estudio

De este estudio se pueden extraer las siguientes conclusiones:

1. Al igual que se indica en la literatura, no existe un algoritmo de clasificación que tenga un mejor rendimiento que otros en todas las ocasiones.

2. Aunque es cierto que los algoritmos bayesianos parecen tener un buen rendimiento a la hora de clasificar conjuntos de datos con pocas instancias, tal y como se concluye en otros trabajos, no son constantes en sus resultados: en cada uno de los conjuntos de datos, y dependiendo de la medida de rendimiento utilizada, el mejor algoritmo variaba. De hecho, algoritmos que mostraron un buen rendimiento con unos conjuntos de datos resultaron ser los que obtenían el peor rendimiento en otros.

3. Los árboles de decisión, en este caso obtenidos con J48, son capaces de obtener buenos resultados de clasificación. En dos de los casos, J48 ha sido el segundo mejor algoritmo, llegando a destacar en términos de significancia con t-test.

4. Las medidas de actividad de los estudiantes en cursos virtuales, como el tiempo o el número de sesiones dedicadas, pueden ser suficientes, por sí solas, para obtener buenos modelos de clasificación.

4.4.2. Estudio 2. Las meta-características simples como predictores

Las meta-características de tipo simple son ampliamente utilizadas en la literatura, como se ha podido comprobar en la sección 4.1. Una de sus principales ventajas es que este tipo de meta-características caracterizan al conjunto de datos de forma global, al contrario que otro tipo de meta-características que caracterizan a los atributos, como son las estadísticas o las de teoría de la información.

Además, y en general, el significado de las características simples es sencillo de entender, lo que hace que los meta-modelos obtenidos puedan ser fáciles de interpretar, pudiendo así el usuario final obtener de forma rápida conclusiones acerca de los motivos por los que se le recomendaría un clasificador para un conjunto de datos concreto.

Otra de las ventajas destacables de las características simples es que, en tiempo computacional, el costo de calcular su valor es relativamente bajo si lo comparamos con el de otras meta-características, requiriendo únicamente "contar", por ejemplo, el número de instancias o de atributos que contiene un conjunto de datos. Las meta-características de tipo estadístico, como la media de los coeficientes de kurtosis o de asimetría de los atributos, o las basadas en *landmarkers*, que requieren de la ejecución de clasificadores, sin embargo, pueden ser más costosas en tiempo computacional.

En este estudio se comprobó hasta qué punto las meta-características simples pueden ser útiles, por sí solas, para obtener un meta-modelo capaz de realizar una correcta recomendación del clasificador que habría de utilizarse para obtener un modelo fiable que prediga el rendimiento de los estudiantes en cursos virtuales.

Configuración del estudio

El proceso seguido en este estudio se resume 6 pasos:

1. Extracción de los datos de los cursos y generación de conjuntos de datos

En este estudio se generaron, en un primer paso, 9 conjuntos de datos con la actividad de los cursos con identificadores del 1 al 3 y del 6 al 11, y otros 2 conjuntos de datos más, llegando hasta un total de 11, construidos al combinar la actividad de los estudiantes en los cursos del 1 al 3 y del 6 al 12. Aunque las instancias del curso con identificador 12 se incluyeron en este último conjunto de datos, se descartó su uso de forma individual debido a su bajo número de estudiantes, nueve.

2. Generación de 72 conjuntos de datos adicionales a los 11 originales

Con objeto de tener un número mayor de conjuntos de datos para la experimentación, se generaron 72 conjuntos adicionales, utilizando los conjuntos de datos originales generados en el paso 1, siguiendo diferentes procesos: (1) aplicando discretización con el método *PKIDiscretize* [194] ofrecido por Weka, (2) ampliando el número de clases a cuatro, reflejando si el estudiante suspendió, o si su calificación estaba comprendida entre el 5 y el 6.9 (aprobado), el 7 y el 8.9 (notable), o el 9 y el 10 (sobresaliente); y a cinco valores, añadiendo la posibilidad de que el estudiante sea clasificado como "abandono"(drop-out), y (3) introduciendo valores perdidos.

En la Tabla 4.8 se muestran las características de estos conjuntos de datos:

Tabla 4.8: Rango de las meta-caraterísticas de los conjuntos de datos utilizados en el estudio 2

Meta-carac.	Rango
Nº Ins.	64-438
Nº Att.	11-22
Nº Att. Num.	0-22
Nº Att. Nom.	0-22
Nº Clases	2-5

3. Aplicación de 11 algoritmos de clasificación sobre los conjuntos de datos

Sobre cada uno de los conjuntos de datos, se aplicaron 11 algoritmos de clasificación implementados en Weka, que responden a diferentes paradigmas:

- **Bayesianos:** NaïveBayes y BayesNetwork.
- **"Lazy":** k-NearestNeighbours y NNge (este último basado además en reglas).

- **Reglas:** OneR, JRip y Ridor.
- **Árboles de decisión:** J48 y SimpleCart.
- **Meta:** Adaboost (con J48 como algoritmo base) y RandomForest.

De la aplicación de estos clasificadores se almacenó el *accuracy* obtenido para cada conjunto de datos, que fue la medida de rendimiento utilizada en el proceso de meta-learning.

4. Creación del conjunto de meta-características sobre el que construir los recomendadores

Para cada conjunto de datos, se extrajeron las meta-características indicadas en la sección 4.3 para este estudio, siendo todas ellas de tipo simple, excepto el balanceo de la clase, que se calculó en base a la fórmula de la Entropía de Shannon.

Con estas meta-características, y junto al rendimiento obtenido por los clasificadores mencionados en el paso 2, en base al *accuracy*, se crearon 3 meta-conjuntos de datos diferentes:

- **"md1":** contenía las meta-características de cada conjunto de datos y, como clase a predecir, el clasificador que mejor rendimiento obtuvo de entre los 11 utilizados en la experimentación. En el caso de que existiesen 2 o más clasificadores que tuvieran el mismo *accuracy*, y éste fuese superior al resto, todos ellos fueron incluidos en el meta-conjunto, añadiendo tantas filas como "mejores clasificadores" hubiera.
- **"md2":** contenía la misma información que "md1", pero solamente considerando los resultados obtenidos por los clasificadores J48, JRip, NaïveBayes y BayesNetwork.
- **"md3":** contenía tantas instancias como modelos de clasificación hubiesen obtenido un *accuracy* cuyo nivel de significancia, utilizando *t-test* con OneR como base, fuese menor de un 5 %. El motivo de utilizar OneR como base fue que, como se vio en el estudio anterior, para una gran parte de los conjuntos de datos es el clasificador con un *accuracy* en media más bajo.

5. Evaluación de la relevancia de las características

De cara a evaluar qué meta-características son realmente relevantes para la tarea de

meta-learning, se aplicaron varios algoritmos de selección de atributos, mostrando en este documento los resultados obtenidos por los algoritmos *CfsSubSetEval* [195] y *ClassifierSubSetEval* [196] siguiendo un proceso de validación 10-CV.

El primero de los algoritmos mencionados, *CfsSubSetEval*, no considera la potencial utilidad de las meta-características para un clasificador en concreto, sino que las evalúa en términos generales. En el caso de *ClassifierSubSetEval*, en esta experimentación se han propuesto 2 algoritmos de clasificación como base para el experimento: J48 y NaïveBayes.

Para contrastar y verificar los resultados, se utilizaron dos métodos de búsqueda diferentes en la ejecución de cada algoritmo de selección de atributos: *BestFirst* y *LinearForwardSelection*.

6. Construcción manual y evaluación de los recomendadores de clasificadores

Utilizando J48, uno de los dos algoritmos con los que se determinó la relevancia de las características en el paso 5, se construyeron 3 recomendadores diferentes, utilizando para cada uno de ellos los conjuntos de meta-características "md1", "md2" y "md3" mencionados en el paso 4. Estos recomendadores fueron validados siguiendo un proceso 10-CV.

Resultados del estudio

Los resultados obtenidos de la selección de atributos pueden observarse en la Tabla 4.9. La relevancia como atributos predictores de las meta-características es medida en un rango de 0 (nada relevante) a 10 (muy relevante). Como puede observarse, la completitud es una de las meta-características más importantes, cuya relevancia es elevada en todos los meta-conjuntos de datos y para ambos algoritmos de clasificación. De ello se puede concluir el elevado peso que tienen la cantidad de valores perdidos en el rendimiento de algunos clasificadores. Otras de las meta-características que destacan sobremanera en todos los conjuntos de datos, y para ambos clasificadores, son el número de instancias del conjunto de datos y el número de atributos.

En cuanto al número de atributos numéricos, no parece tener especial relevancia para construir los clasificadores, al igual la meta-característica que indica los tipos de atributos que componen el conjunto de datos ("#Type att."). Algo similar sucede con el número de atributos nominales, que si bien destaca sobremanera en el caso de utilizar NaïveBayes sobre "md1" o en el caso de utilizar J48 sobre "md3", no muestra un rendimiento

constante en el resto de conjuntos de datos, en donde su rendimiento es, en general, bastante bajo. Lo mismo sucede con el número de clases. Por último, el atributo que indica si la clase está balanceada tiene una alta relevancia para el meta-conjunto de datos "md1", y moderada para "md2", siendo muy baja en el caso de "md3".

Tabla 4.9: Relevancia de las meta-características simples en el estudio 2

Característica	Conjunto	NaïveBayes		J48	
		BF	LF	BF	LF
#N Instances	md1	10	9	5	5
	md2	10	8	10	10
	md3	8	8	8	8
#N Attributes	md1	8	7	8	9
	md2	7	9	6	2
	md3	5	6	7	8
#N Numeric att.	md1	1	1	0	3
	md2	2	0	0	4
	md3	0	5	2	5
#N Nominal att.	md1	9	9	6	3
	md2	2	0	5	0
	md3	4	1	10	6
Completeness	md1	10	10	10	10
	md2	7	8	10	10
	md3	7	7	9	9
#Type att.	md1	7	7	4	3
	md2	4	6	2	4
	md3	6	5	3	4
#N Classes.	md1	7	6	2	2
	md2	1	1	8	9
	md3	3	3	5	5
Is_balanced?	md1	8	9	8	8
	md2	6	9	5	7
	md3	3	3	3	3

Ya conocemos las meta-características, de entre las del tipo simple junto con el balanceo de la clase, que parecen más útiles de cara a construir un recomendador (meta-modelo) que indique qué clasificador habría de utilizarse para predecir el rendimiento de los estudiantes en un curso, pero, ¿cómo son los posibles recomendadores que podemos obtener?. En la Figura 4.1 podemos ver el recomendador obtenido al aplicar J48 sobre "md2". Tal y como puede observarse, este recomendador es bastante sencillo de interpretar, incluso para un usuario no experto, ya que este usuario únicamente habría de entender los conceptos de tipo de atributos, número de instancias, número de atributos y completitud para decidir que algoritmo de clasificación utilizar sobre sus datos. Por ejemplo, el árbol recomendaría utilizar el algoritmo JRip si los datos son de tipo numérico, el número de

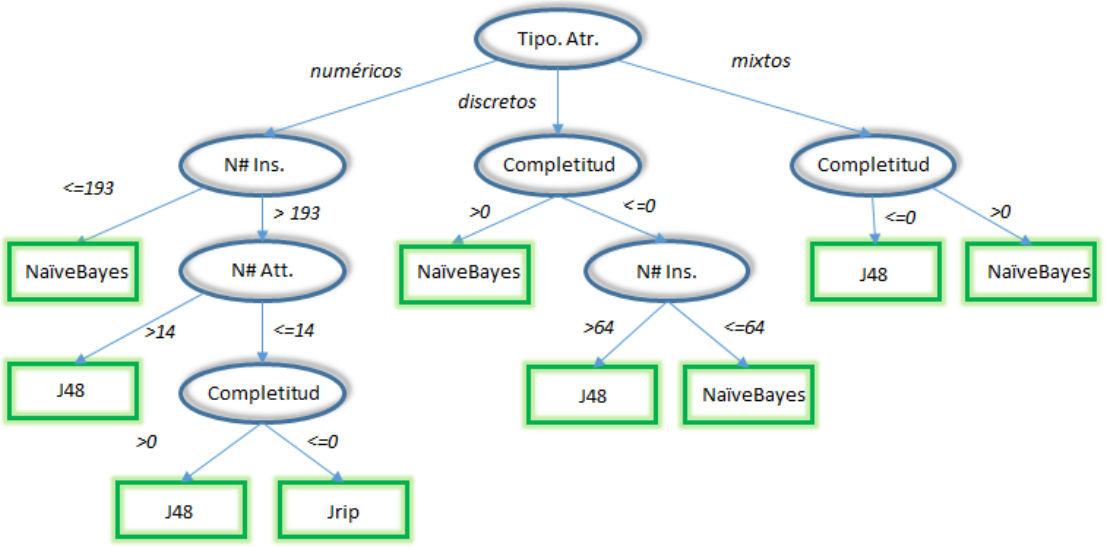


FIGURA 4.1: Recomendador J48 del estudio 2

instancias es mayor de 193, el número de atributos es menor o igual a 14, y si además no existen valores perdidos. O, por ejemplo, si existieran en el conjunto de datos tanto atributos numéricos como nominales, en ese caso dependería de la completitud el que el algoritmo recomendado fuese J48 o NaïveBayes.

Tal y como puede observarse, además, en el árbol aparecen las 3 características que más destacan en la tarea de selección de atributos. No obstante, la característica que se encuentra en la raíz del árbol no es ninguna de ellas, sino que se corresponde con otra que, en la selección de atributos, no parecía tener especial relevancia: el tipo de atributos.

En cuanto al *accuracy* del meta-modelo mostrado, este es de un 68.75 %, siendo el modelo con mayor rendimiento de todos los obtenidos. Este hecho es debido a que en los modelos obtenidos con los meta-conjuntos "md1" y "md3", tenemos 11 clases, correspondientes a 11 posibles clasificadores. Este número de clases es elevado con respecto al conjunto de datos disponibles, lo que provoca que estos meta-conjuntos estén notablemente desbalanceados, con algunos clasificadores siendo "los mejores", en términos de *accuracy*, en pocas de las meta-instancias, mientras que otros lo son en muchas de ellas.

Conclusiones del estudio

En este estudio, se ha podido constatar que las meta-características simples pueden ser moderadamente útiles, por sí solas, para caracterizar a los conjuntos de datos y realizar

recomendaciones sobre los clasificadores que habrían de utilizarse para obtener modelos de predicción de calidad. De entre éstas, el número de instancias, de atributos y la completitud han mostrado ser especialmente relevantes y estar altamente correladas con el rendimiento de los clasificadores. No obstante, en los meta-modelos obtenidos se ha podido observar que otras meta-características, como el tipo de atributos (numéricos o nominales) también pueden ser útiles para su construcción. El balanceo de clase, única meta-característica que no es de tipo simple y que se ha incluido en este primer estudio, también muestra un moderado potencial en la tarea de selección de características. Viendo los resultados obtenidos con el meta-conjunto "md2", podemos concluir que es posible construir un recomendador, basándose en meta-modelos obtenidos con clasificadores y utilizando meta-características simples, que recomiende el algoritmo de clasificación que un usuario debería de utilizar para obtener un modelo de predicción del rendimiento de los estudiantes.

No obstante, y debido principalmente al bajo número de conjuntos de datos disponibles, este estudio tiene una serie de limitaciones que han de ser abordadas y superadas en sucesivos trabajos. Debido a este hecho, como ha podido constatarse, no ha sido posible obtener un buen recomendador de clasificadores utilizando los meta-conjuntos "md1" y "md3", que contenían como clase a predecir 11 posibles clasificadores, por lo que se limita notablemente el número de algoritmos que pueden ser recomendados. Por otro lado, para que el recomendador construido con el meta-conjunto "md2", que tenía 4 clasificadores como valor de clase, fuese fiable, se tuvieron que generar conjuntos de datos artificiales adicionales para incluirlos en la experimentación. Si bien esto es una práctica bastante común en muchos de los trabajos vistos en la literatura del apartado 4.1, el hecho de que existan conjuntos de datos similares entre sí, y que sólo difieran en un reducido número de meta-características, puede llevar a que el meta-modelo construido esté demasiado ajustado a los conjuntos de datos utilizados en la experimentación.

En la siguiente sección 4.4.3, se continúa con un segundo estudio sobre la utilidad de las meta-características simples para la recomendación de clasificadores, extendiendo el experimento actual mediante la creación de nuevos recomendadores, los cuáles fueron probados simulando un hipotético escenario real en el que llegan nuevos conjuntos de datos sobre los que se desea una recomendación del clasificador a utilizar.

4.4.3. Estudio 3. Construcción de recomendadores con las meta-características simples

El objetivo de este nuevo estudio de las meta-características simples fue determinar hasta que punto los recomendadores obtenidos eran capaces de recomendar, utilizando como prueba 3 conjuntos de datos no utilizados para construirlos, un algoritmo de clasificación con mejor rendimiento, en base al *accuracy*, que la mayoría o el resto de los incluidos en el estudio. Se trató de establecer, por tanto, ya no solamente si los recomendadores eran capaces de recomendar al mejor algoritmo, sino si, al menos, era recomendado uno con un buen rendimiento.

Configuración del estudio

Los pasos seguidos han sido los siguientes:

1. Extracción de los datos de cursos e-learning y generación de conjuntos de datos

Además de los cursos utilizados en el estudio anterior, se añadieron los datos de otros nuevos y, además, se generaron nuevos conjuntos de datos mediante la aplicación de los siguientes procesos, con objeto de garantizar una mayor variabilidad en las diferentes características:

- **Discretización:** se aplicó el algoritmo *PKIDiscretize* ofrecido por Weka sobre distintos conjuntos de datos con objeto de tener en la experimentación atributos de tipo discreto.
- **Multiclase:** se han generado conjuntos de datos con diferentes clases: con 2 valores: suspenso y aprobado; con 4 valores: suspenso, aprobado, notable y sobresaliente; y con 5 valores: suspenso, aprobado, notable, sobresaliente y abandono.
- **Valores nulos (desconocidos):** se añadieron en 18 de los conjuntos de datos generados hasta el momento un porcentaje de nulos del 10 %, 20 %, 30 % y 40 %. El porcentaje se basa en la dimensión del conjunto de datos (número de atributos * número de instancias).
- **Re-balanceo de la clase:** se aplicó el algoritmo *SMOTE* [197] (implementación de Weka) para obtener 4 conjuntos de datos bi-clase adicionales con las siguientes proporciones de clase: 80-20 %, 85-15 %, 70-30 % y 90-10 %

Tabla 4.10: Caracterización de los conjuntos de datos del estudio 3

Características	cluster0	cluster1	cluster2	cluster3	cluster4
N# Ins.	438	119.54	512.86	147	401.37
N# Att.	14	16.93	14	19	20.11
% Att. Disc.	85.5	8.68	0	93.29	0
% Att. Núm.	15	91.07	100	6.57	100
% Desconocidos	19.62	16.24	12.64	11.19	16.54
Balanceo	Poco desb.	Balanceado	Poco desb.	Balanceado	Muy desb.

En total, la experimentación contó con 99 conjuntos de datos, de los que 3 fueron utilizados como prueba, y los otros 96 como entrenamiento. En la tabla 4.10 se muestran los diferentes tipos de conjuntos de datos utilizados en este experimento, obtenidos utilizando el algoritmo k-Means con k=5 (implementación SimpleKMeans de Weka). Como puede observarse, los 5 grupos definen a conjuntos de datos con diferentes características:

- **Cluster 0:** agrupa a aquellos conjuntos de datos con un elevado número de instancias y que, en media, tenían un porcentaje de atributos discretos superior al 85 % sobre el total de atributos.
- **Cluster 1:** contiene a los conjuntos de datos cuyo número de instancias es más bien bajo, pero que tenían un considerable número de atributos, siendo además la mayor parte de ellos numéricos.
- **Clusters 2 y 4:** ambos son bastantes similares, y se diferencian fundamentalmente en que el clúster 2 contiene los conjuntos cuyo número de instancias era un poco más elevado que los del 4, mientras que en este último los conjuntos de datos destacaban por tener un gran número de atributos.
- **Cluster 3:** caracteriza a aquellos conjuntos de datos con relativamente pocas instancias, pero un alto número de atributos, que en su gran mayoría son discretos.

2. Aplicación de 12 algoritmos de clasificación sobre los 96 conjuntos de datos de entrenamiento

Los clasificadores utilizados, de diferente paradigma, en la experimentación sobre los 96 conjuntos de datos son los siguientes:

- **Árboles:** J48 y SimpleCart
- **Lazy:** k-Nearest Neighbourds y NNge (este último también basado en reglas)
- **Reglas:** OneR, JRip, Ridor, DecisionTable

- **Meta:** Adaboost (con J48 como algoritmo base) y RandomForest.
- **Bayesianos:** NaïveBayes y BayesNetwork.

Para la ejecución de estos algoritmos, se utilizó la implementación y los parámetros por defecto de Weka.

3. Creación del conjunto de meta-características sobre el que construir los recomendadores

En este paso, se generó un conjunto de meta-características con 111 meta-instancias, que contuviese las características de los 96 conjuntos de entrenamiento utilizadas en este experimento. El motivo de que existan un número más alto de meta-instancias que de conjuntos de datos se explica debido a que, para algunos de estos conjuntos, existían 2 o más algoritmos de clasificación cuyo rendimiento fue equivalente entre ellos y a su vez superior al del resto. En ese supuesto, se añadieron tantas meta-instancias como algoritmos de clasificación tuvieran esas características.

4. Construcción manual de un recomendador de clasificadores

Se utilizaron dos técnicas de clasificación de diferente paradigma, J48 y NaïveBayes, para construir, evaluar y comparar dos recomendadores, teniendo como atributos predictores las meta-características de los conjuntos de datos, y como clase a predecir el clasificador o clasificadores con mejor *accuracy*, de entre los 12 mencionados en el paso 2, para cada conjunto de datos.

5. Evaluación del recomendador

Los tres conjuntos de prueba, utilizados para evaluar el rendimiento de los recomendadores construidos con J48 y NaïveBayes, se seleccionaron procurando que existiesen diferencias entre ellos con respecto a las características más relevantes, en base a la aplicación y estudio de técnicas de selección de características. Estos conjuntos de prueba, cuyas características se muestran en la Tabla 4.11, representaban a aquellos que tenían un número de instancias bajo (64), medio (194) y alto (304), en comparación con el valor de esta meta-característica en los conjuntos de datos de entrenamiento. Además, el número de atributos también variaba, teniendo los conjuntos test2 y test3 un número bajo o medio, 4 y 10 respectivamente, mientras que el test1 tenía un número notablemente alto, 18. Todos los atributos de los conjuntos test1 y test2 eran numéricos, mientras que 4 de los atributos del test3 eran nominales. En cuanto al balanceo de la clase, tanto

Tabla 4.11: Descripción de los conjuntos de prueba del estudio 3

Conjuntos	N# Ins.	N# Att.	N# Att. Núm.	N# Att. Disc.	N# Clases	% desconocidos	Balanceo
test1	64	18	18	0	2	0	Poco desb.
test2	194	4	4	0	2	0	Balanceada
test3	304	10	6	4	2	0	Balanceada

Tabla 4.12: Ranking de los 5 clasificadores con un mayor accuracy para los conjuntos de prueba del estudio 3

Conjuntos	Clasificador	Ranking	Accuracy
test1	NaiveBayes	1	85.9375
	RandomForest	2	82.8125
	NNge	2	82.8125
	kNearestNeigbourds	3	79.6874
	J48	4	78.125
	BayesNetwork	4	78.125
test2	BayesNetwork	1	84.0206
	SimpleCart	2	83.5052
	DecisionTable	2	83.5052
	J48	3	82.9897
	JRip	3	82.9897
test3	J48	1	86.1963
	kNearestNeigbourds	2	85.5828
	JRip	3	84.3558
	RandomForest	4	83.7423
	SimpleCart	5	83.4653

los conjuntos test2 como test3 tenían unos valores de clase balanceados, mientras que test1 tenía la clase ligeramente desbalanceada.

Resultados del estudio

En la Tabla 4.12 se muestra un ranking con el rendimiento de los 5 algoritmos de clasificación que obtuvieron un mayor *accuracy* para cada uno de ellos. Puede observarse cómo los algoritmos con un mejor rendimiento difieren en cada uno de los conjuntos de datos de prueba: mientras que, por ejemplo, en el conjunto test1 los tres algoritmos con mejor rendimiento son NaïveBayes, RandomForest y NNge, en el test2 este lugar les coresponde a BayesNetwork, SimpleCart y DecisionTable, y en el test3, a J48, kNearest Neighbourds y JRip. En el caso del test1, además, la diferencia entre el *accuracy* del mejor clasificador, NaïveBayes, y los clasificadores en las posiciones 4 y 5, J48 y BayesNetwork, es notablemente alta. En los conjuntos test2 y test3 esta diferencia también es notable aunque, sin embargo, la diferencia entre el de mejor rendimiento y los segundo y tercero mejores es más baja.

Tabla 4.13: Recomendaciones de los meta-modelos construidos con J48 y NaïveBayes para los conjuntos de prueba del estudio 3

Conjuntos	Recomendación con J48	Recomendación con NaïveBayes
test1	JRip	RandomForest
test2	JRip	RandomForest
test3	J48	JRip

En la Tabla 4.13 se muestran las recomendaciones de cada uno de los dos recomendadores, construidos con J48 y NaïveBayes, para cada uno de los conjuntos de datos de prueba. Tal y como puede observarse, el recomendador J48 acierta en recomendar el mejor algoritmo para el conjunto test3, mientras que el recomendador NaïveBayes da como salida para este mismo conjunto al algoritmo JRip, que resulta ser el 3º mejor, y cuya diferencia de *accuracy* con respecto al primero es baja, de menos del 2 %.

En los otros dos conjuntos de prueba, test1 y test2, la recomendación no es tan certera: en el caso del recomendador J48, éste recomienda utilizar en ambos casos JRip, un algoritmo que, si bien no tiene un mal rendimiento en el test2, no aparece entre los 5 primeros con mejor rendimiento para el test1. Algo similar sucede con el recomendador NaïveBayes, cuya salida en ambos casos es RandomForest. Esto puede ser debido, en gran medida, a que este algoritmo resulta ser el que tiene un mejor rendimiento, en base al *accuracy*, en casi un 25 % de los conjuntos utilizados como entrenamiento, lo que descompensa y desbalancea demasiado la clase del meta-conjunto de datos al tener 12 posibles valores, uno por cada clasificador utilizado. Este hecho ya se denotaba en el estudio anterior, y es que teniendo un conjunto limitado de pruebas, es difícil desarrollar esta experimentación con un alto número de clasificadores.

Debido a los problemas encontrados al utilizar un alto conjunto de clasificadores en relación al bajo número de conjuntos de datos, se realizó una segunda comparativa, en la que se seleccionaron 4 de los clasificadores con mejor rendimiento en los conjuntos de datos: JRip, J48, NaïveBayes y BayesNetwork. En base a ellos, se construyeron dos nuevos recomendadores con J48 y NaïveBayes, cuya recomendación para cada uno de los conjuntos de prueba puede verse en la Tabla 4.14.

En este caso, el recomendador J48 acierta con los mejores algoritmos para los conjuntos test1 y test3, que son respectivamente NaïveBayes y J48. En el caso del recomendador NaïveBayes, éste acierta a recomendar el segundo mejor algoritmo de clasificación para los conjuntos test2, J48, y test3, JRip.

Tabla 4.14: Recomendaciones de clasificadores para los conjuntos de entrenamiento del estudio 3

Conjuntos	Recomendación con J48	Recomendación con NaïveBayes
test1	NaïveBayes	JRip
test2	JRip	J48
test3	J48	JRip

Conclusiones del estudio

Al igual que ocurría en el anterior estudio, el reducido número de conjuntos de datos disponibles para realizar el estudio provoca que ambos recomendadores construidos en base a los 12 clasificadores no tengan un buen rendimiento, y que sólo en el caso del conjunto test3 estos recomendadores hayan sido fiables. No obstante, al reducir el número de clasificadores de 12 a 4, el rendimiento de los dos recomendadores aumenta notablemente, hasta el punto de que uno de ellos llega incluso a recomendar al mejor de ellos en 2 de 3 ocasiones.

En la siguiente sección 4.4.4, se incluyen los resultados de una comparativa similar, a la cuál se añadieron, a las meta-características simples utilizadas en el presente estudio, dos tipos de meta-características no usadas todavía en la construcción de los recomendadores: las meta-características de complejidad y las asociadas al contexto de los cursos.

4.4.4. Estudio 4. Más allá de las meta-características simples: las meta-características de complejidad y de contexto

Las meta-características simples han demostrado ser moderadamente útiles, por sí solas, de cara a construir un meta-modelo que, en base a los valores de estas características en los conjuntos de datos, recomiende el algoritmo de clasificación a utilizar. No obstante, en los apartados 4.4.2 y 4.4.3 se han evidenciado las limitaciones de los recomendadores construidos en base a únicamente estas meta-características.

En este apartado, se presenta un nuevo estudio en el cual se han incluido, además de las meta-características simples, otros tipos: las de complejidad y las de contexto.

Configuración del estudio

Pasos dados en esta experimentación:

1. Extracción de los datos de cursos y generación de conjuntos de datos

Originalmente, se generaron 32 conjuntos de datos con la actividad de los estudiantes

Tabla 4.15: Descripción de los conjuntos de datos del estudio 4

Meta-carac.	Rango
N# Instancias	de 13 a 502
N# Atributos	de 5 a 28
N# conjuntos de cursos trasversales	36 de 64
N# conjuntos de e-learning	32 de 64

de los cursos con identificador del 1 al 24. Dado que todos los atributos predictores de estos conjuntos de datos eran numéricos, se generaron otros 32 conjuntos de datos en los que se discretizaron estos atributos utilizando el método *PKIDiscretize* ofrecido por Weka. Las características principales de los 64 conjuntos pueden verse en la Tabla 4.15

De estos 64 conjuntos de datos, 4 fueron seleccionados como prueba, emulando una hipotética situación real en la que un usuario requiere una recomendación para nuevos conjuntos de datos que no han sido utilizados en el entrenamiento.

2. Aplicación de 4 algoritmos de clasificación sobre los 60 conjuntos de datos de entrenamiento

Dado el reducido conjunto de datos, y teniendo en consideración las conclusiones extraídas en los estudios anteriores, el estudio final se ha reducido a utilizar 4 clasificadores de diferente paradigma como base del experimento: NaïveBayes (bayesiano), NNge ("lazy" y reglas), JRip (reglas) y J48 (árbol de decisión).

3. Creación del conjunto de meta-características sobre el que construir los recomendadores

Se generaron 2 meta-conjuntos de datos con los 60 conjuntos de entrenamiento:

- **"md1":** contenía las características de cada uno de estos conjuntos junto con el valor de clase que indica que clasificador tuvo un mejor rendimiento en términos de *accuracy*.
- **"md2":** misma información que "md1", excepto que en este caso la clase indicaba qué clasificador obtuvo el mejor rendimiento en términos de *True Positive Rate* (*TPrate*), que se corresponde con los estudiantes suspensos bien clasificados.

4. Construcción del recomendador de clasificadores

Dada la simplicidad para comprender los meta-modelos construidos con J48 en los estudios anteriores, este algoritmo es el utilizado para generar los dos recomendadores en base a los meta-conjuntos "md1" y "md2".

TABLA 4.16: Características de los 4 conjuntos de prueba del estudio 4

Conjunto de prueba	N# Ins.	N# Att.	Carácter	Tipo de curso	Tipo de att.
test1	17	11	transversal	online	numericos
test2	65	12	transversal	online	discretos
test3	502	17	específico	blended	numéricos
test4	80	14	específico	blended	discretos

Conjunto de prueba	F1	F2	F3	F4	T1	T2	N1	N2	N3
test1	1.16	0	0.53	1	0.88	1.7	0.65	0.84	0.35
test2	4.06	0.01	0.54	0.97	1	5.91	0.31	0.67	0.19
test3	1.83	0	0.33	0.60	1.00	31.50	0.17	0.42	0.112
test4	2.99	0.4	0.05	0.06	1	6.15	0.51	0.86	0.36

Conjunto de prueba	Mejor Acc.	Mejor TPrate
test1	JRip	J48
test2	NaïveBayes	NaïveBayes
test3	J48	J48
test4	NNge	NaïveBayes

5. Evaluación del recomendador

En esta última fase del proceso, se evaluaron los 4 conjuntos de prueba con los recomendadores construidos, para determinar la precisión de los mismos. Como puede observarse en la Tabla 4.16, estos 4 conjuntos se escogieron en base a las diferentes características que tienen entre ellos, e igualmente buscando que el mejor clasificador para cada uno de ellos, tanto en términos de *accuracy* como de *TPrate*, fuese también diferente.

Resultados del estudio

Los recomendadores construidos con el algoritmo J48 utilizando los meta-conjuntos de meta-características "md1" y "md2" se muestran en las Figuras 4.2 y 4.3. En ambos recomendadores, puede destacarse que la meta-característica más relevante es el carácter del curso, lo que quiere decir que el rendimiento de los clasificadores depende en gran medida de si el curso está abierto a todos los perfiles de estudiantes, o sólo a los de un área de conocimiento. No obstante, ésta es la única coincidencia destacable entre los dos recomendadores, ya que si nos fijamos en el resto de características relevantes, éstas difieren notablemente. En el caso del recomendador basado en *accuracy*, las meta-características de complejidad F1 y T2 toman especial relevancia, apareciendo en el segundo nivel del árbol, mientras que en el recomendador basado en *TPrate*, las meta-características de complejidad son un poco menos relevantes, apareciendo F4 y F2 en el nivel 3 del árbol de decisión. En este recomendador, tienen más relevancia las meta-características simples que informan del número de instancias y atributos.

Las recomendaciones resultantes de evaluar los conjuntos de prueba en ambos recomendadores se muestran en la Tabla 4.17. Junto a ellas, se muestra también el algoritmo

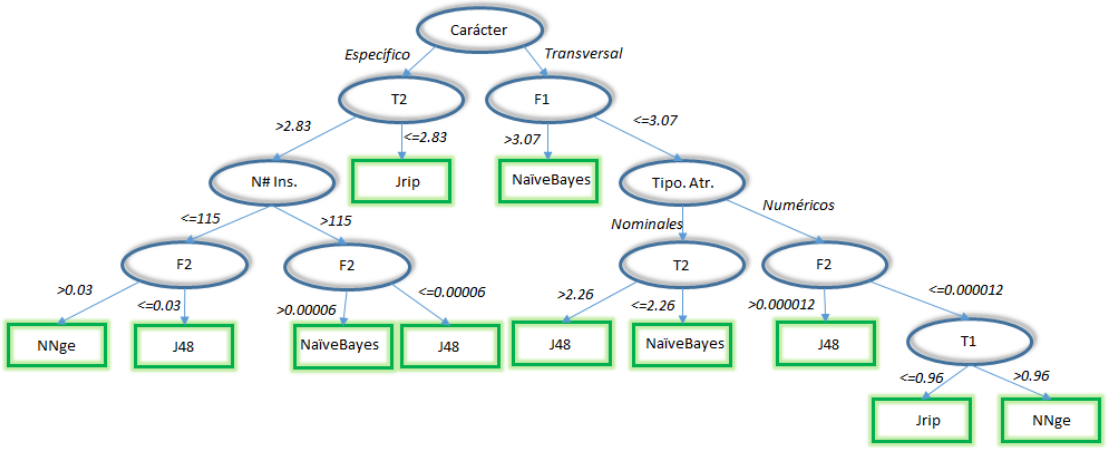


FIGURA 4.2: Recomendador J48 para el conjunto md1 (Mejor accuracy) del estudio 4

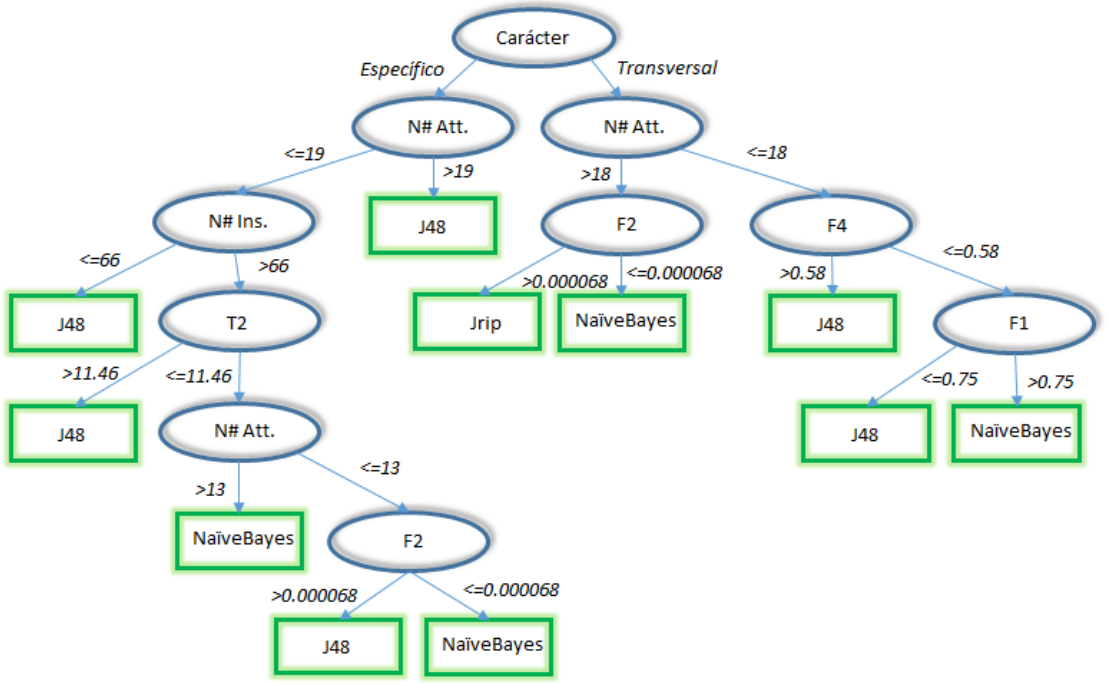


FIGURA 4.3: Recomendador J48 para el conjunto md2 (Mejor TPrate) del estudio 4

de clasificación con mejor rendimiento para cada conjunto de prueba. Como puede observarse, en el caso de realizar la recomendación basándonos en el *accuracy*, en 3 de los 4 conjuntos de datos se recomienda el clasificador con mejor rendimiento. Sólo en el caso del conjunto test4 se recomienda un clasificador diferente, NaïveBayes, al mejor, NNge. No obstante, NaïveBayes resulta ser el segundo mejor clasificador para test4, por lo que la recomendación no es del todo mala. Lo mismo ocurre con los resultados del recomendador basado en *TPrate*, recomendado el mejor clasificador en 3 de 4 ocasiones, y recomendado al segundo mejor en el caso del conjunto test1.

Tabla 4.17: Clasificador recomendado para cada conjunto de prueba del estudio 4

Conjunto	Mejor Acc.	Recom. Alg.	Mejor TPrate	Recom. Alg.
test1	JRip	JRip	J48	NaïveBayes
test2	NaïveBayes	NaïveBayes	NaïveBayes	NaïveBayes
test3	J48	J48	J48	J48
test4	NNge	NaïveBayes	NaïveBayes	NaïveBayes

Conclusiones del estudio

El carácter de los cursos muestra ser una de las meta-características más importantes de cara a escoger el clasificador con mejor rendimiento. Los resultados mostrados con el algoritmo J48 prueban este extremo, por lo que se puede afirmar que, dependiendo de si el curso es transversal o específico, los clasificadores a aplicar para obtener buenos modelos de predicción pueden ser completamente diferentes. Por otro lado, y si bien las meta-características simples siguen mostrando un buen potencial predictor, al igual que en los estudios anteriores, la suma de las meta-características de complejidad al estudio ha redundado en una mejora de los recomendadores. En ambos casos con J48, tanto para recomendar un clasificador en base al *accuracy* como en base al *TPrate*, meta-características como el F1, el F2 o el T2 aparecen en los primeros niveles de los árboles de decision. No obstante, este estudio adolece del mismo problema que los previos, y es que el limitado conjunto de datos disponible para los experimentos provoca que el número de clasificadores a recomendar tenga que ser bajo.

4.4.5. Estudio 5. Nuevo enfoque del proceso de meta-learning: ranking con regresión

Una de las limitaciones que más ha destacado en todas los estudios de este capítulo es la falta de un número suficiente de conjuntos de datos sobre los que realizar el proceso de meta-learning. Este hecho ha obligado a generar nuevos conjuntos de datos mediante la aplicación, sobre los originales, de diferentes técnicas de pre-procesado.

Aunque los recomendadores construidos han tenido un buen rendimiento, dadas las limitaciones acerca del bajo número de clasificadores que pueden ser utilizados con las anteriores propuestas, el siguiente paso lógico consistió en buscar otro enfoque que permitiera ampliar la experimentación y ofreciese un recomendador sin las limitaciones respecto del número de clasificadores que puede recomendar, y que, además, pudiera construirse sin necesidad de tener que crear conjuntos de datos artificiales.

En la literatura, hemos podido ver que existen otras propuestas para construir meta-modelos que van más allá de, simplemente, buscar el mejor clasificador. Una de ellas consiste en aplicar técnicas multi-clase, permitiendo así que para un conjunto de datos con unas meta-características concretas no exista un sólo mejor clasificador, sino varios. No obstante, esto no soluciona el problema de los conjuntos de datos artificiales, ya que con pocos conjuntos, el meta-modelo seguiría teniendo un rendimiento muy bajo. Otra posibilidad, bastante extendida en la literatura, podría ser utilizar el algoritmo kNN para determinar qué conjuntos de datos tienen características similares a un nuevo conjunto sobre el que se desea aplicar un clasificador, y en base al rendimiento de un conjunto de clasificadores sobre los primeros, crear un ranking, de forma que el clasificador recomendado sea aquel que ha tenido un mejor rendimiento en un mayor número de ocasiones. Este procedimiento evitaría tener que construir un meta-modelo con un clasificador, pero el problema de la falta de conjuntos de datos persiste, y de hecho se agrava, ya que la recomendación se haría en base a un subgrupo, reduciendo aún más el número de conjuntos utilizados.

En el estudio cuyos resultados se presentan en este apartado, se exploró la aplicación de un proceso diferente de meta-learning para crear el recomendador: crear un ranking de clasificadores mediante la generación de modelos regresivos, siguiendo la propuesta mostrada en la Figura 4.4. Como puede observarse, en este proceso existen dos fases diferenciadas. En una primera fase la base de datos de experimentos es alimentada con las meta-características de los conjuntos de datos de entrenamiento y las medidas del rendimiento de los clasificadores que se han aplicado a dichos conjuntos.

En la segunda fase, correspondiente al proceso de prueba, en vez de generarse un meta-modelo de clasificación cuyas clases a predecir sean los mejores algoritmos para cada conjunto de entrenamiento, lo que se generan son unos meta-modelos de regresión para cada uno de los clasificadores de la fase de entrenamiento. Estos meta-modelos de regresión generan una predicción del rendimiento que va a tener cada uno de los clasificadores para el conjunto de datos utilizado como prueba. Finalmente, con esta predicción, se crea un ranking que ordena el rendimiento predicho para cada clasificador de mayor a menor, siendo por tanto el clasificador recomendado aquel que tenga un rendimiento predicho más alto.

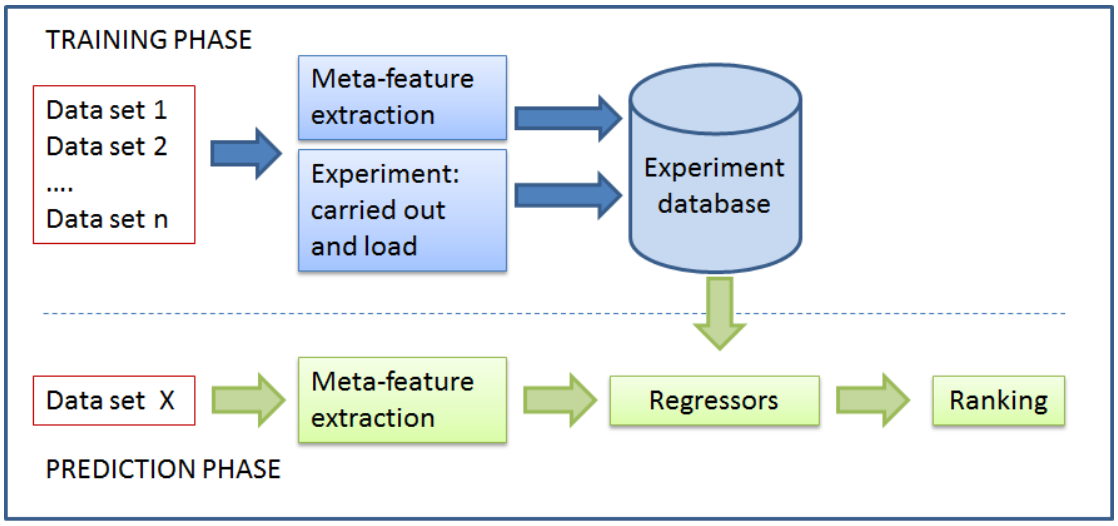


FIGURA 4.4: Propuesta de proceso de meta-learning con modelos de regresión

Configuración del estudio

Pasos dados en esta experimentación:

1. Extracción de los datos de los cursos y generación de conjuntos de datos

En esta experimentación, se generaron 30 conjuntos de datos con la actividad de los estudiantes de los cursos con identificadores del 1 al 24. Al contrario que en los estudios anteriores, no se utilizaron conjuntos con datos artificiales, ni se crearon nuevos conjuntos fruto de la unión de los originales.

2. Aplicación de 11 algoritmos de clasificación sobre los 30 conjuntos

Aunque el sistema de recomendación propuesto pudiera estar inicialmente orientado a usuarios con ciertos conocimientos en minería de datos que no tengan problemas en interpretar modelos de clasificación complejos, el objetivo principal de esta tesis es poder ofrecer modelos de predicción del rendimiento de los estudiantes a profesores de cursos virtuales. Dado el perfil no experto de estos usuarios, los modelos de predicción que se les ofrezcan han de ser sencillos de interpretar.

En base a ello, en el estudio que se presenta se escogieron 11 algoritmos de clasificación cuyos modelos pudiesen responder a esta característica:

- **Árboles:** J48 y SimpleCart
- **Lazy:** NNge (este último también basado en reglas)
- **Reglas:** OneR, JRip, Ridor

- **Meta:** Adaboost y Bagging (con J48 como algoritmo base), y RandomForest.
- **Estadísticos:** BayesNetwork y LogisticRegression

Estos clasificadores fueron aplicados sobre los 30 conjuntos de datos definidos en el paso 1, almacenando el *accuracy* obtenido. Con objeto de garantizar la calidad de los modelos, el *accuracy* se calculó en base al proceso de evaluación *leave-one-out*.

3. Extracción de las meta-características de los conjuntos de datos

En este paso, las características indicadas en el apartado 4.3 fueron extraídas de cada uno de los conjuntos de datos. Como puede verse en las Tablas 4.18, 4.19, 4.20 y 4.21, las meta-características de los conjuntos de datos tenían un amplio rango de valores, lo que garantizó un conjunto de experimentación heterogéneo, con conjuntos de datos de muy diferentes características.

Tabla 4.18: Rango de valores de las meta-características de complejidad en el estudio 5

F1	F1v	F2	F3	F4	L1	L2
0.04-29.46	0.03-370.82	0-0.2	0.02-0.88	0.05-1	0.27-0.92	0.09-0.45
L3	N1	N2	N3	N4	T1	T2
0.07-0.5	0.05-0.93	0.25-1.23	0-0.67	0-0.49	0.6-1	0.82-33.62

Tabla 4.19: Rango de valores de las características simples en el estudio 5

#N Ins.	#N Att.	#N Fail	#N Pass
13-504	3-28	5-220	3-433

Tabla 4.20: Rango de valores de las características estadísticas en el estudio 5

Max Ske.	Min Ske	Avg. Ske	Max. Kurt	Min. Kurt	Avg. Kurt
8.02-500.12	(-)1.51-15.22	2.99-131.43	1.43-22.33	(-)13.48-3.23	(-)2.3-10-36

Tabla 4.21: Rango de valores de los landmarkers en el estudio 5

Acc. NB	Acc. LD	Acc. BN	Acc. RN	Acc. 1NN
23.33-95.35	35-97.5	23.08-100	31.33-100	40-100

4. Generación de los conjuntos de meta-características

Para cada uno de los 11 clasificadores mencionados en el paso 2, se generó un conjunto

de meta-características con 30 instancias, conteniendo como atributos predictores las características de los conjuntos de datos extraídas en el paso 3, y como atributo a predecir el *accuracy* obtenido por el clasificador bajo evaluación.

5. Generación y evaluación de los modelos de regresión

En este estudio, se utilizó Regresión Lineal como técnica para predecir, utilizando los conjuntos de meta-características generados en el paso 4, el *accuracy* de cada uno de los clasificadores del paso 2.

Para poder evaluar la utilidad de la propuesta en estudio, por cada uno de los 11 conjuntos de meta-características generados en el paso 4, se generaron 15 modelos de regresión lineal diferentes, utilizando conjuntos de atributos predictores diferentes. Los criterios para generar estos modelos de regresión fueron los siguientes:

1. Usar todas las meta-características disponibles.
2. Usar las meta-características de un sólo tipo de forma independiente: simples, estadísticas, complejidad o *landmarkers*.
3. Usar sólo las meta-características escogidas con un algoritmo de selección de atributos. Con este propósito, se utilizó la técnica *ClassifierSubSetEval*, ofrecida por Weka, con *BestFirst* como criterio de búsqueda, y Regresión Lineal como clasificador base. Para que las conclusiones sobre la relevancia de las meta-características fuesen lo más fiables posibles, se utilizó *leave-one-out* como método de evaluación. Diez modelos de Regresión Lineal diferentes se generaron con este proceso, eliminando a las instancias con una relevancia inferior al 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 %, 90 % y 95 % respectivamente.

Al igual que se hizo con las técnicas de selección de atributos, todos los modelos de regresión lineal se generaron utilizando la estrategia *leave-one-out* como método de validación. En total, se tiene un conjunto de 4950 experimentos (15 modelos de regresión lineal * 30 iteraciones del proceso leave-one-out * 11 clasificadores a predecir). Para la evaluación de estos modelos, se utilizó el Error Cuadrático Medio (*Root-Mean Squared Error*, *RMSE* por sus siglas).

6. Selección y despliegue de los modelos de regresión y evaluación del sistema

Por cada uno de los 15 modelos de regresión lineal generados para los 11 clasificadores,

Tabla 4.22: RMSE de los meta-regresores del estudio 5

Algoritmo	all	all*	com	com*	ldm	ldm*	sim	sim*	sta	sta*
AdaBoost	0.143	0.069	0.164	0.093	0.045	0.043	0.128	0.117	0.136	0.115
Bagging	0.12	0.074	0.171	0.073	0.049	0.048	0.118	0.099	0.102	0.097
BayesNet	0.217	0.064	0.195	0.115	0.113	0.094	0.203	0.17	0.197	0.18
J48	0.177	0.069	0.111	0.069	0.051	0.049	0.124	0.101	0.103	0.096
JRip	0.195	0.042	0.132	0.074	0.079	0.073	0.165	0.126	0.141	0.124
LogisticRegression	0.292	0.05	0.106	0.05	0.081	0.062	0.154	0.147	0.149	0.133
NNge	0.213	0.037	0.104	0.049	0.043	0.041	0.145	0.129	0.128	0.115
OneR	0.335	0.051	0.108	0.072	0.058	0.051	0.143	0.114	0.131	0.11
RandomForest	0.195	0.047	0.051	0.045	0.053	0.047	0.127	0.109	0.095	0.089
Ridor	0.205	0.048	0.098	0.074	0.052	0.052	0.14	0.119	0.12	0.113
SimpleCart	0.259	0.091	0.197	0.096	0.14	0.125	0.216	0.164	0.173	0.167

se seleccionó 1 por clasificador, aquél con un menor *RMSE*, para ser desplegado en el sistema, siendo por tanto ese modelo de regresión lineal el que predecirá el *accuracy* para un clasificador concreto.

Con objeto de establecer la viabilidad del sistema propuesto, se realizó una exhaustiva evaluación de los resultados, evaluando para cada uno de los 30 conjuntos de datos utilizados cuántas veces nuestro sistema nos recomienda un clasificador con un *accuracy* mayor que el resto, o mayor que la mayoría. Esta comparativa se hizo en base a los modelos de regresión generados en el proceso de evaluación *leave-one-out*, de forma tal que, cuando se evaluó la recomendación para un conjunto de datos, se realizó en base al modelo de regresión generado con los otros 29, sin incluirlo en el conjunto de entrenamiento.

Resultados del estudio

En la Tabla 4.22 se muestra el *RMSE* obtenido por un subconjunto de los modelos de regresión generados en el paso 5. Cada una de las columnas muestra este *RMSE* al utilizar en dichos modelos todas las meta-características ("all"), sólo las de complejidad ("com"), sólo las simples ("sim"), sólo *landmarkers* ("ldm") y sólo las estadísticas ("sta"). Las columnas marcadas con "∗" muestran el *RMSE* obtenido tras aplicar el proceso de selección de atributos con un *threshold* del 95 %, eso es, la columna ("all" "∗") muestra el *RMSE* del modelo de regresión obtenido al utilizar de entre todas las meta-características, únicamente aquellas señaladas por el proceso de selección, mientras que en la columnas ("sim" "∗"),("com" "∗"),("sta" "∗") y ("ldm" "∗") se muestra el *RMSE* de los modelos construidos al sólo utilizar las meta-características seleccionadas de entre las de tipo simple, de complejidad, estadísticas o *landmarkers*.

Como puede ser observado, el *RMSE* de los modelos construidos al utilizar todas las meta-características es considerablemente más alto, por lo que una de las conclusiones que pueden extraerse del análisis de esta tabla es que utilizar todas las meta-características no lleva a obtener un buen modelo de regresión que prediga el *accuracy* de ninguno de los clasificadores. Lo mismo sucede al utilizar solamente meta-características simples o estadísticas, con valores *RMSE* superiores al 0.1, e incluso al 0.2 en algunos casos. En el caso de utilizar meta-características de complejidad, solamente dos de los regresores, los construidos para predecir el *accuracy* de los clasificadores RandomForest y Ridor, parecen tener un rendimiento aceptable, con un *RMSE* por debajo de 0.1, y que en el caso de RandomForest llega a bajar hasta un 0.051. No obstante, es con los *landmarkers* con los que, utilizados de formar aislada como predictores, se consiguen obtener los mejores modelos de regresión sin necesidad de aplicar el proceso de selección de atributos. Excepto en los casos de los regresores que predicen el *accuracy* de SimpleCart y BayesNet, que superan el valor 0.1, el resto de regresores tienen un *RMSE* notablemente inferior a ese umbral, e incluso valores menores del 0.5, como sucede con AdaBoost y Bagging.

Sin embargo, las conclusiones cambian al aplicar la selección de atributos, eliminando aquellas meta-características con un *threshold* inferior al 95 %, previo paso a la generación de los meta-regresores. En todos los casos, bien sea aplicando la selección de atributos sobre todas las meta-características, o bien únicamente sobre un tipo, el *RMSE* de los modelos disminuye.

Para facilitar la comparativa, en las Figuras 4.5 y 4.6 se muestran dos diagramas en los que se comparan los meta-regresores construidos sin aplicar ("no SA") y aplicando ("SA") el proceso de selección al utilizar solamente meta-características simples o estadísticas. Como puede observarse, la mejora al usar meta-características de tipo simple o estadístico no es muy elevada, con muchos de los meta-regresores superando notablemente aún el umbral del 0.1 de *RMSE*.

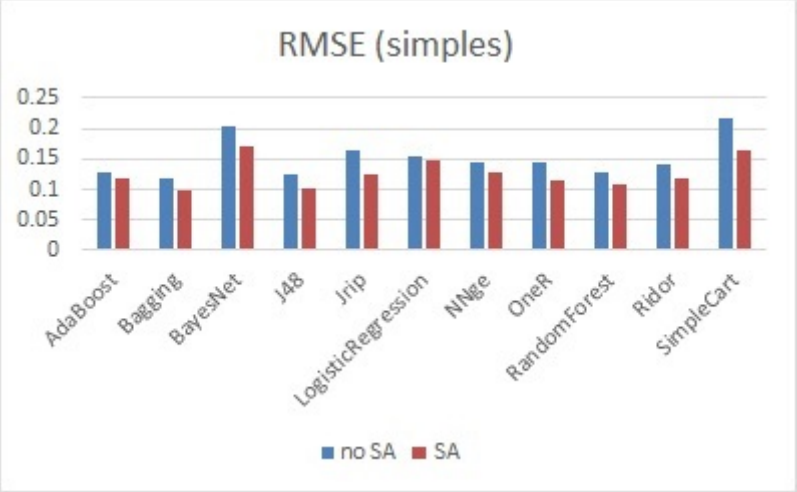


FIGURA 4.5: RMSE utilizando meta-características simples con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5

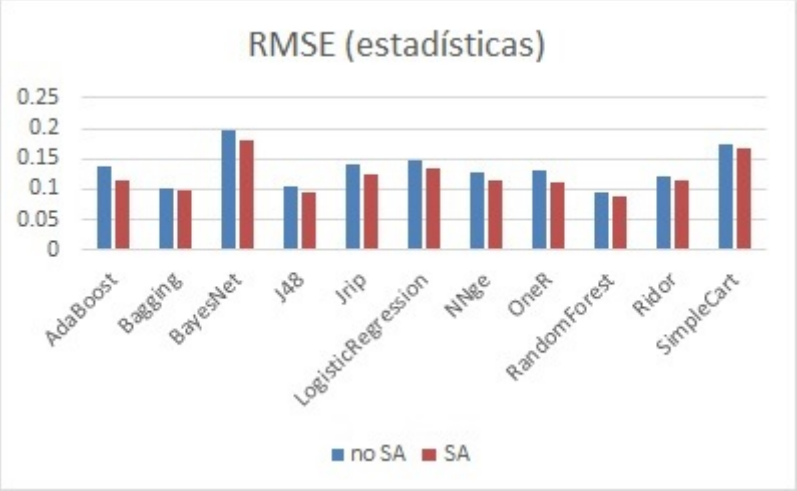


FIGURA 4.6: RMSE utilizando meta-características estadísticas con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5

Con los *landmarkers* la mejora tampoco es excesiva, tal y como se muestra en la Figura 4.7. No obstante, estas meta-características ya permitían obtener meta-regresores con bajo *RMSE* sin aplicar el proceso de selección, por lo que este resultado era esperable. Aún así, se consigue una ligera mejora con los meta-regresores construidos para los 11 clasificadores.

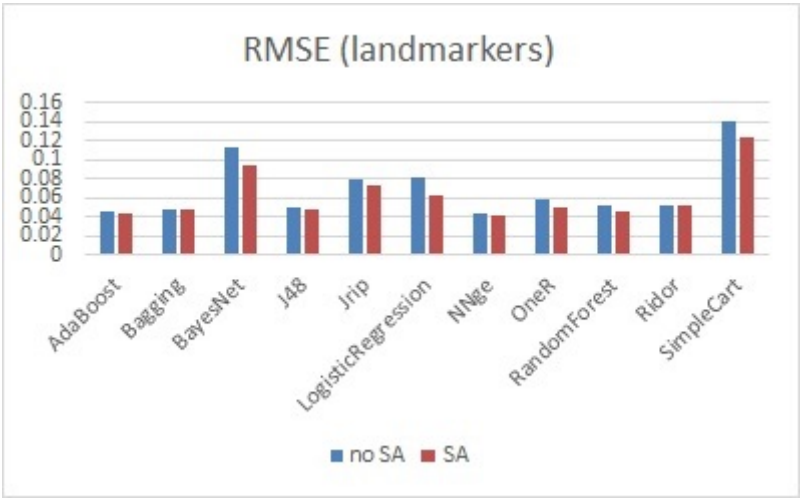


FIGURA 4.7: RMSE utilizando landmarkers con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5

En el caso de las meta-características de complejidad, Figura 4.8, la mejora es mucho más notable. Valgan como ejemplo los dos meta-regresores construidos para predecir el *accuracy* del clasificador Bagging, en el que el meta-regresor construido previa selección de atributos obtiene un *RMSE* de 0.073, mientras que sin aplicar este proceso se obtenía un valor de 0.171.

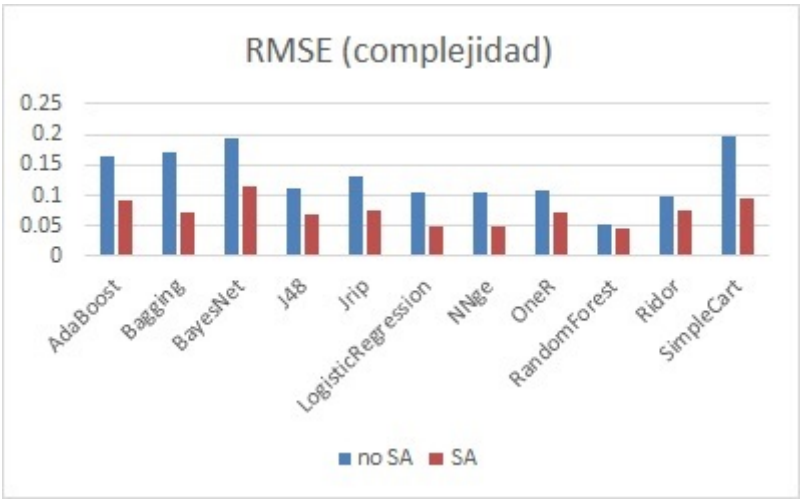


FIGURA 4.8: RMSE utilizando meta-características de complejidad con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5

Sin embargo, la mejora es verdaderamente notable si se aplica el proceso de selección usando todas las meta-características, algo que puede observarse en el diagrama de la Figura 4.9. De hecho, meta-regresores como los construidos para NNge, OneR o LogisticRegression pasan de ser de los que tenían peor rendimiento, con valores *RMSE* de

0.213, 0.335 y 0.292 respectivamente, a ser de los que mejor rendimiento obtienen, con valores de 0.037, 0.051 y 0.05.

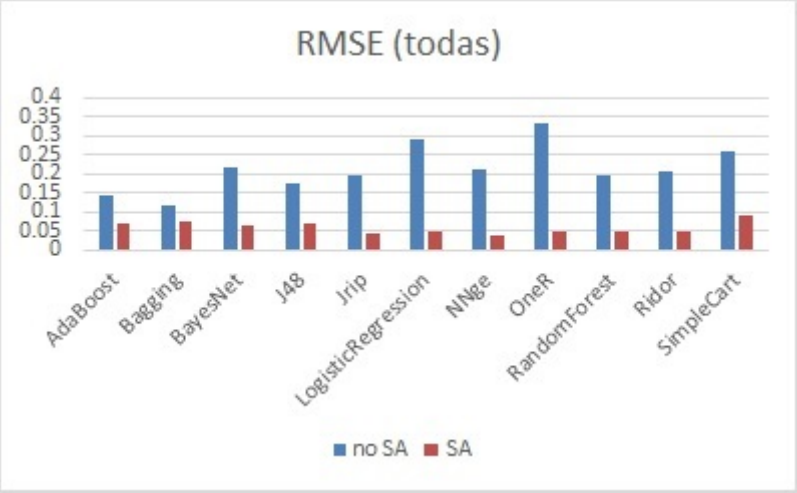


FIGURA 4.9: RMSE utilizando todas las meta-características con (SA) y sin (no SA) selección de atributos para cada clasificador del estudio 5

Al calcular la media de *RMSE* de los meta-regresores con diferentes meta-características, las conclusiones obtenidas acerca del beneficio de utilizar selección de atributos se refuerza. En la Figura 4.10 se muestran estos valores. La media de *RMSE* es más baja, en todos los casos, con selección de atributos, destacando esta mejora sobremanera si se utilizan todas las meta-características.

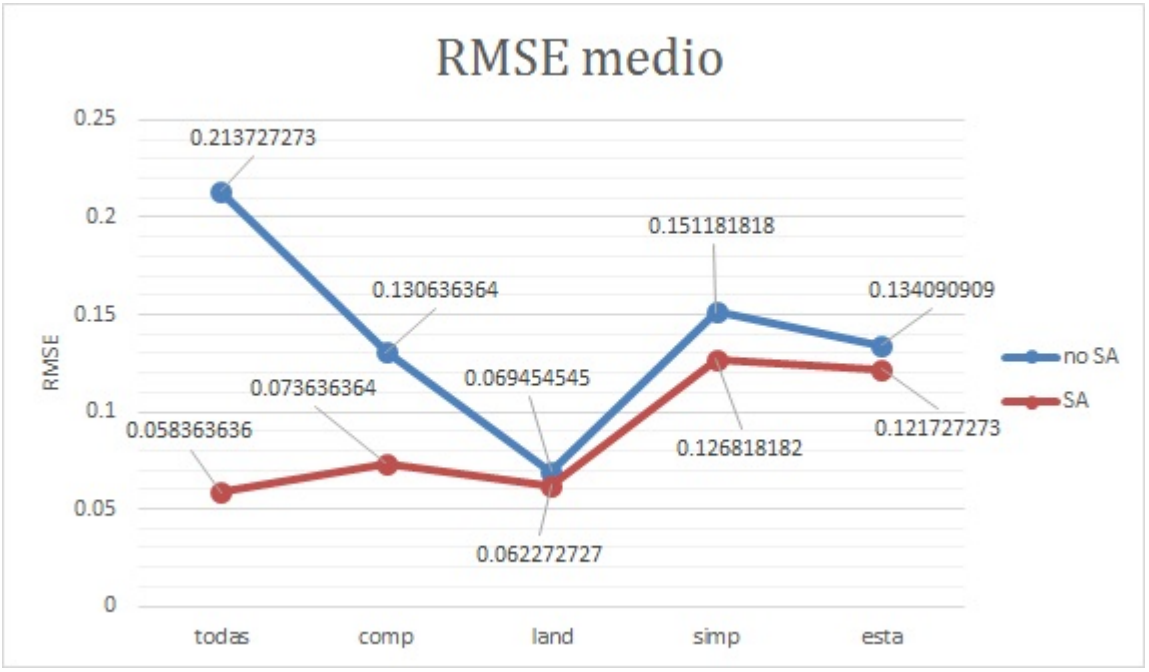


FIGURA 4.10: RMSE medio con (SA) y sin (no SA) selección de atributos en el estudio 5

En la Figura 4.11 se muestra una comparativa del *RMSE* obtenido con los meta-regresores construidos utilizando todas las meta-características y sólo *landmarkers*, tras aplicar el proceso de selección. Puede observarse claramente como con 6 de los clasificadores, el meta-regresor con mejor rendimiento, esto es, menor *RMSE*, es el obtenido con todas las meta-características. En 3 de las ocasiones, correspondientes a J48, Bagging y AdaBoost, el mejor meta-regresor se obtiene utilizando únicamente *landmarkers*, y en otras dos ocasiones, con OneR y RandomForest, los meta-regresores tienen el mismo rendimiento.

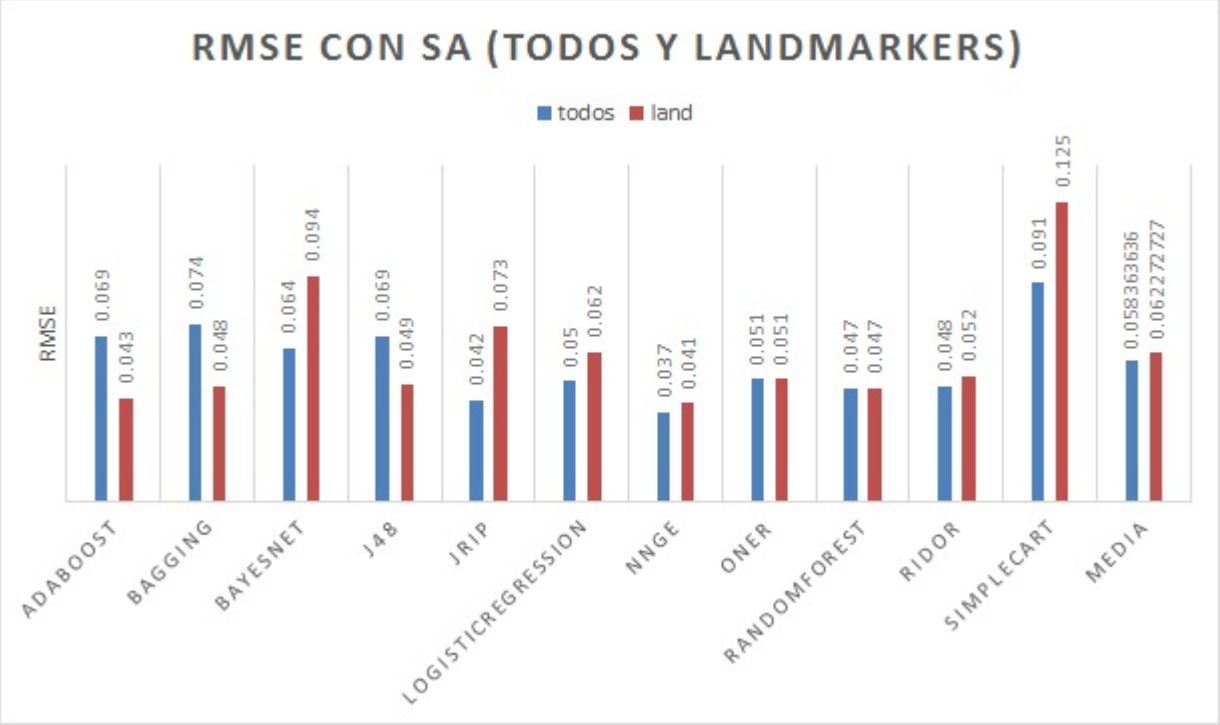


FIGURA 4.11: Comparativa del RMSE utilizando todas las meta-características vs utilizando sólo landmakers con selección de atributos en el estudio 5

Se puede concluir, por tanto, que pese a que los *landmarkers* pueden obtener, por sí solos, buenos meta-modelos de regresión que predigan el *accuracy* de los clasificadores, y pese a que las meta-características de tipo simple o estadístico no tienen buen rendimiento por sí mismas, es utilizando todas las meta-características, previo filtrado con selección de atributos, cuando se obtienen los mejores modelos.

Dado que el objetivo de este estudio es poder envolver el sistema propuesto en E-learning WebMiner, el siguiente paso consiste en comprobar hasta qué punto la propuesta era viable. Con este objetivo, se seleccionaron, para predecir el *accuracy* de los clasificadores, los meta-regresores con un *RMSE* menor para cada uno de ellos. Esto es, en los casos de

J48, AdaBoost y Bagging, se desplegaron en el sistema los meta-regresores construidos con *landmarkers*, y para el resto de clasificadores los construidos usando todas las meta-características previa selección de atributos.

En un siguiente paso, el sistema recomendador fue probado con los mismos 30 conjuntos de datos, siguiendo un proceso *leave-one-out*, de forma tal que cada conjunto de datos fue únicamente evaluado con los meta-modelos generados al utilizar los 29 restantes. De esta forma, cada conjunto de datos, al evaluarse, podría ser considerado como un nuevo conjunto que llega al sistema. En la Tabla 4.23, se muestra en cuántas ocasiones el sistema recomienda el mejor clasificador, esto es, el clasificador con un mayor *accuracy* que el resto, o bien al menos selecciona un clasificador cuyo *accuracy* real se encuentra en el 1º, 2º, 3º o 4º cuartil, esto es, un clasificador cuyo *accuracy* supere al 75 %, 50 %, 25 % o 0 % del *accuracy* del resto de clasificadores.

Como puede observarse, el mejor algoritmo de clasificación es recomendado un 23.33 % de las ocasiones, y en un 56.67 % el algoritmo recomendado está en el 1º cuartil. Más aún, el 83.34 % de las veces el sistema escoge un clasificador del 1º o 2º cuartil, y solamente en el 6.67 % de los casos, el clasificador cae en el 4º cuartil.

Tabla 4.23: Número y porcentaje de veces que el mejor clasificador es recomendado por el sistema del estudio 5. Se incluye comparativa por cuartiles

	# veces (sobre 30)	% veces
mejor clasificador	7	23.33
primer cuartil	17	56.67
segundo cuartil	8	26.67
tercer cuartil	3	10.00
cuarto cuartil	2	6.67

En la Tabla 4.24, se muestra un ejemplo de cómo quedaría el ranking de predicción de *accuracy* para uno de los conjuntos de datos utilizados en la experimentación, comparando el *accuracy* predicho para cada algoritmo (Columna “Acc. Pred.”) con el real (columna “Acc. Real”). En este caso, el algoritmo recomendado es RandomForest con un *accuracy* predicho por el sistema del 88.237 %, siendo su *accuracy* real el segundo más alto, 88.083 %. Por otro lado, Bagging, cuyo *accuracy* real es el mismo que RandomForest, 88.083 %, es clasificado en segundo lugar, con un *accuracy* predicho de 87.054 %.

El clasificador con mayor *accuracy* real, JRip, con un 88.601 %, es a su vez clasificado en la mitad de la tabla.

Tabla 4.24: Ranking basado en el accuracy predicho (Acc. Pred.) y real (Acc. Real) para un conjunto de datos del estudio 5

Clasificador	Rank.	Acc. Pred.	Acc. Real	Diferencia
RandomForest	1	88.237	88.083	0.154
Bagging	2	87.054	88.083	-1.029
SimpleCart	3	86.927	87.565	-0.638
AdaBoost	4	86.768	86.528	0.240
J48	5	86.596	88.083	-1.487
JRip	6	86.565	88.601	-2.036
NNge	7	86.562	86.528	0.034
BayesNet	8	86.058	88.063	-2.005
OneR	9	86.034	84.456	1.578
LogisticRegression	10	82.464	82.902	-0.438
Ridor	11	80.824	85.492	-4.668

Uno de los principales problemas que afecta al ranking es el alto error obtenido en la predicción del *accuracy* del que es el clasificador con mejor rendimiento, JRip, habiendo una diferencia con el *accuracy* real de -2.036. Dado que el error con el resto de los clasificadores con un *accuracy* predicho superior al de JRip es menor de 2, y en su mayoría menor de 1, el hecho de que éste sea tan alto para JRip provoca que el sistema no logre recomendar el mejor clasificador, si bien ha conseguido recomendar a uno de los que obtiene el segundo mejor *accuracy*, RandomForest.

No obstante, el sistema no sólo ha logrado recomendar a uno de los mejores clasificadores, sino que los clasificadores con un peor *accuracy* real, OneR, LogisticRegression y Ridor, son a su vez clasificados como los peores por el sistema, con los tres valores de *accuracy* predicho más bajos. De hecho, mientras que la diferencia de *accuracy* real entre el clasificador recomendado, RandomForest, y el clasificador con mejor rendimiento, JRip, es de sólo 0.518 %, la diferencia entre RandomForest y el tercer peor clasificador se eleva al 2.591 %, y con respecto al peor, LogisticRegression, al 5.181 %.

Llegado a este punto, se puede afirmar que el sistema alcanza un rendimiento bastante bueno de cara a recomendar algoritmos de clasificación, siendo muy pocas las ocasiones en las que recomienda clasificadores con bajo *accuracy* en comparación con el resto.

Conclusiones del estudio

El proceso de recomendación de clasificadores propuesto obtiene buenos resultados, llegando a recomendar en un 23.33 % de las ocasiones al mejor clasificador, en un 56.67 % a un clasificador del primer cuartil, y en 83.34 % a un clasificador del primer o segundo cuartil. Este proceso, por tanto, es capaz de recomendar clasificadores con un buen rendimiento y, al menos, evitar que se utilicen aquellos de que vayan a tener un rendimiento más bajo que el resto.

Por otro lado, al predecir no ya el mejor clasificador, como se hacía en los estudios anteriores, sino el rendimiento de cada uno de los incluidos en el estudio con meta-regresores, se evita el problema de no poder utilizar un amplio conjunto de clasificadores. Con este nuevo proceso, se pueden incluir tantos clasificadores como se desee, sin importar que el conjunto de datos disponible sea relativamente bajo.

En cuanto a las meta-características, los *landmarkers* por sí solos son suficientemente útiles como para obtener predicciones certeras. No obstante, la conclusion principal que se extrae de los resultados es que, para construir el mejor recomendador posible, se hace necesario el uso de todos los tipos de meta-características (simples, estadísticas, de complejidad, *landmarkers*), realizando una posterior selección de aquellas con mayor relevancia.

4.5. Conclusiones y trabajo futuro

Tanto en base a la literatura como a los estudios realizados en el desarrollo de esta tesis, puede afirmarse que no existe un sólo algoritmo de clasificación que sea mejor que el resto, en el caso que nos preocupa, para la predicción del rendimiento de los estudiantes en base a la actividad que éstos realizan en cursos virtuales. En este capítulo, se ha mostrado cómo los procesos de meta-learning para la selección y recomendación de algoritmos pueden ser útiles de cara a seleccionar y recomendar aquellos clasificadores que se espera tengan un mejor rendimiento.

Las meta-características simples, que son también las que tienen un menor coste computacional, pueden ser útiles por sí solas para este propósito, si bien los mejores recomendadores de algoritmos se obtienen al añadir otros tipos de meta-características, como son las estadísticas, las de complejidad, los *landmarkers*, o las de contexto. De hecho, en el estudio 5 los *landmarkers* prueban ser las meta-características con mayor poder predictivo.

Por otro lado, el escaso número de conjuntos de datos disponibles, un problema común en la literatura sobre predicción del rendimiento en cursos virtuales, hace que procesos de recomendación más sencillos como la selección del mejor clasificador (estudios 2, 3 y 4) tengan la limitación de no poder recomendar en base a un conjunto elevado de clasificadores. No obstante, en el estudio 5 se prueba que, al utilizar un proceso de ranking, este problema puede ser solventado, pudiendo hacer recomendaciones sobre un conjunto de clasificadores más elevado.

En definitiva, la recomendación de clasificadores mediante procesos de meta-learning puede convertir en una caja negra para el usuario final el proceso de selección de un clasificador que le garantice un modelo de predicción de calidad. De esta forma, los profesores de cursos virtuales podrían obtener estos modelos sobre sus estudiantes sin necesidad de intervenir en el proceso. A tal efecto, en el capítulo 7 se propone la inclusión del proceso del estudio 5 en la herramienta ElWM.

Como trabajo futuro, los estudios mostrados en esta tesis pueden ampliarse utilizando nuevos tipos de meta-características, como las de teoría de la información, o las basadas en modelos. Por otro lado, el proceso del estudio 5 podría ser redefinido al combinarlo con los procesos de selección basados en Nearest-Neighbours, de forma que los meta-regresores para predecir el *accuracy* sólo se obtengan en base a los conjuntos de datos cuyas meta-características sean más similares a la del nuevo conjunto sobre el que queremos predecir. El mismo estudio 5 también puede ser extendido utilizando como valor a predecir otras medidas que no sean el *accuracy*.

Capítulo 5

Detección y eliminación de comportamientos anómalos para la mejora de los modelos de predicción

Durante el desarrollo de las fases de extracción y análisis de los datos, así como en las de modelado y evaluación de patrones de predicción del rendimiento de los estudiantes, se observó que en muchos de los cursos existían estudiantes cuyo nivel de actividad no parecía corresponder con el rendimiento obtenido. Así, por ejemplo, se detectaron casos de estudiantes con bajo rendimiento cuyo tiempo y sesiones dedicadas a las distintas actividades del curso eran muy altas, y en ocasiones incluso las más altas en comparación con la actividad del resto de sus compañeros. Sin embargo, lo habitual en estudiantes con ese rendimiento es tener una actividad baja. Igualmente, en el análisis de los datos se pudieron encontrar estudiantes con una actividad muy baja, e incluso casi nula, que no obstante habían aprobado el curso, siendo este un comportamiento más cercano a los estudiantes que suspenden o abandonan.

Una de las explicaciones que puede tener esta situación es que el estudiante con alta actividad y bajo rendimiento simplemente dejase abierta la conexión al curso durante un tiempo prolongado, sin realizar ninguna acción en el mismo, por lo que la actividad extraída de la base de datos del LCMS no se correspondería con el comportamiento real

de este estudiante. En el caso de los estudiantes aprobados con muy baja actividad, en algunos cursos pudo observarse que ciertos de estos estudiantes tenían tendencia a descargarse los contenidos del curso en su ordenador, por lo que no requerirían permanecer conectados al sistema para consultar los apuntes. Al incluirse a este estudiante en los conjuntos de entrenamiento, estos comportamientos anómalos provocan que los modelos de clasificación construidos tengan una menor precisión.

En los estudios de este capítulo, se realiza una comparativa de técnicas y procesos de detección y eliminación, y se propone una nueva, para eliminar estas instancias de los conjuntos de datos de entrenamiento, y con el objetivo final de mejorar los modelos de predicción obtenidos.

Esta nueva técnica, denominada DARIM (siglas de *Distance-based Algorithm to Remove Instances that Are Misclassified*), puede ser también usada con objeto de detectar, durante la impartición de un curso, a aquellos estudiantes con bajo rendimiento pese a tener una actividad similar a los estudiantes con buen rendimiento. De esta forma, el profesor tendría la posibilidad de identificar a estos estudiantes y proporcionarles una realimentación personalizada que les ayudase a mejorar sus resultados, evitando así la perdida de motivación y posible abandono del curso.

El capítulo se estructura de la siguiente forma: en el apartado 5.1 se presenta un resumen de los principales trabajos y técnicas para el tratamiento de *class-outliers*, así como de los trabajos del área educativa relacionados con la detección de comportamientos anómalos. En el apartado 5.2 se muestran y analizan gráficamente varios ejemplos de comportamientos anómalos por parte de los estudiantes, detectados en los cursos utilizados en esta tesis, y que afectan al rendimiento de los clasificadores. El apartado 5.3 describe la hipótesis de partida e incluye un resumen de los estudios más relevantes realizados en el desarrollo de esta tesis. El apartado 5.4 se muestran la configuración, el proceso seguido, los resultados y las conclusiones de cada uno de los estudios incluidos en este documento. Finalmente, en el apartado 5.5 se establecen las conclusiones finales con los resultados de todas las estudios.

5.1. Estado del arte: detección de outliers y aplicación al entorno educativo

En su trabajo, Hawkins [198] define *outlier* como una observación que se desvía notablemente respecto de otras observaciones, pudiendo ser debido a que ha sido generada por un mecanismo diferente al del resto. En la literatura, existen otras muchas definiciones de lo que es un *outlier*, pero la mayoría de ellos siguen la misma definición que Hawkins. Existen multitud de técnicas para detectar *outliers* [199]. Algunas de estas técnicas están basadas en métodos estadísticos [200, 201]; otras se basan en densidades, como *Local Outlier Factor* (LOF por sus siglas) [202]; o incluso en distancias, como *Class Outliers Distance-Based* (CODB por sus siglas) y *Enhanced CODB* (ECODB por sus siglas) [203]. El campo de la detección de *outliers* es muy amplio y tiene muchas aplicaciones prácticas, como son la limpieza de los datos [204][205], la detección del fraude [206], o la detección de intrusiones en redes de comunicación [207] [208]. La existencia de *outliers* puede indicar la existencia de individuos, o grupos de individuos, con un comportamiento notablemente diferente en comparación con la mayoría de los individuos en un conjunto de datos. Aunque la eliminación de *outliers* pueda mejorar los estimadores, hay ocasiones en las que estos *outliers* tienen un significado concreto, y esta información puede perderse si los *outliers* son eliminados. En esta línea es en la que se enfocan numerosos trabajos de detección de *outliers* [209], [210].

Si bien la detección de *outliers* es un campo muy estudiado en el área de la minería de datos, en comparación son pocos los algoritmos y técnicas que tengan en cuenta el valor de clase de las instancias, lo que se conoce por el término *class-outlier* [211]. Una de las principales líneas de investigación con los *class-outliers* es el efecto que estos tienen en los procesos de clasificación con minería de datos, y cómo su detección y tratamiento puede ser beneficioso para mejorar los modelos de predicción obtenidos, tal y como se contempla en el *survey* de Frenay et al. [1]. En él, los autores proponen una clasificación, resumida en la Tabla 5.1, de las técnicas o procesos que pueden utilizarse para obtener mejores modelos de predicción en presencia de *class-outliers*. Se puede observar que en los encuadramientos I, II y IV, lo que se proponen son técnicas consideradas "robustas" o, al menos, "tolerantes" a los *class-outliers*. En el encuadramiento III, por el contrario, lo que se propone son técnicas que sirvan para detectar y eliminar instancias del conjunto de entrenamiento consideradas *class-outliers*.

TABLA 5.1: Encuadramiento de los procesos y técnicas de clasificación en presencia de ruido, según el Survey de Frenay et al. [1]

I. Métodos robustos ante la presencia de class-outliers	**II. Métodos probabilisticos tolerantes a la presencia de class-outliers**
Ejemplos: Ensambladores como LogitBoost [212] y BrownBoost [213], o "boosting with margin maximization"[214]; entre otros [215, 216].	Ejemplos: algoritmos bayesianos [217–219], métodos de clustering para asignar clases [220, 221]; entre otros [222, 223]
III. Métodos de limpieza de outliers	**IV. Métodos basados en modelos tolerantes a la presencia de class-outliers**
Detección de instancias mal etiquetadas ("misslabeled instances") [224, 225]; basadas en modelos de predicción, como la eliminación directa en el conjunto de entrenamiento de instancias mal clasificadas [226] o mediante técnicas de voto [227] y particionado [228]; basados en vecinos cercanos [229]; métodos de ensamblado con eliminación de instancias de mayor peso [230] o las que son frecuentemente mal clasificadas [231]; entre otros [232, 233].	Variantes tolerantes a outliers del algoritmos basados en percentrones [234], como los citados en [234–236]; algunos árboles de decisión como CN2 [237]; métodos de boosting [238–240] o de combinación de boosting y bagging [241, 242]; métodos semi-supervisados [243–245]; o algoritmos basados en SVM con procesos embebidos de limpieza de datos [246].

Algunos ejemplos de técnicas que detecten *class-outliers* son las ya mencionadas CODB y ECODB [203]. Por otra parte, y con objeto de mejorar los modelos de clasificación, en la literatura existen numerosos trabajos que demuestran que los métodos de ensamblado de clasificadores, consistentes en detectar como *class-outliers* a aquellas instancias que hayan sido mal clasificadas por un porcentaje de estos clasificadores ("Voto Mayoritario"), obtienen un buen rendimiento [247, 248].

En cuanto a la detección de *outliers* en conjuntos de datos educativos, si bien en el capítulo 3 ha quedado constancia de que en la literatura pueden encontrarse numerosos trabajos sobre clasificación del rendimiento de los estudiantes, apenas existen unos pocos trabajos enfocados en la mejora de estos modelos predictivos [249, 250]. De hecho, la mayor parte de los trabajos se enfocan en temas como la detección de trampas en

exámenes en plataformas virtuales [251], la detección de comportamientos poco usuales [252], o el detectar patrones de aprendizaje irregulares por parte de los estudiantes [253].

5.2. Visualizando los comportamientos anómalos

Con objeto de mostrar al lector el tipo de *outliers* que pueden encontrarse en los conjuntos de datos educativos utilizados en la presente tesis, en este apartado se muestran ejemplos de ellos gráficamente.

En la Figura 5.1 se muestra una gráfica con la distribución de los estudiantes en base a dos atributos de actividad: el número de sesiones (eje Y) y el tiempo total empleado por los estudiantes (eje X) en los cursos con identificadores 1, 2 y 3. Los estudiantes aprobados son representados con cuadrados, y los suspensos con triángulos. Como es de esperar, la mayor parte de los estudiantes que suspendieron los cursos tienen un valor bajo en ambos atributos y, viceversa, aquellos que aprobaron tienen un valor alto. A la izquierda de la primera de las líneas verticales dibujadas en la gráfica están los estudiantes suspensos con una dedicación muy baja o incluso casi nula en el curso. Tras la segunda línea vertical se encuentran los estudiantes que han dedicado un alto número de sesiones y tiempo al curso, habiendo aprobado la mayor parte de ellos. Podemos observar como en esta zona de la gráfica existe un conjunto de estudiantes que, pese a tener también una alta dedicación en el curso, suspendieron. Este comportamiento irregular o anómalo es debido, principalmente, a que estos estudiantes mantuvieron abiertas algunas sesiones en el curso y, pese a ello, no realizaron ninguna actividad en él.

Dentro del área delimitada por las dos líneas verticales están mezclados varios de los estudiantes que aprobaron con otros que suspendieron. No obstante, la mayor parte de los estudiantes aprobados en este área dedicaron mayor tiempo y sesiones que los suspensos, aunque en este caso es más difícil detectar a aquellos con comportamientos irregulares, necesitando por tanto de la inclusión de otros atributos de actividad. Teniendo en cuenta que sólo se están utilizando dos atributos, esta dificultad para detectarlos se incrementa exponencialmente al incluir más atributos y, por tanto, más dimensiones.

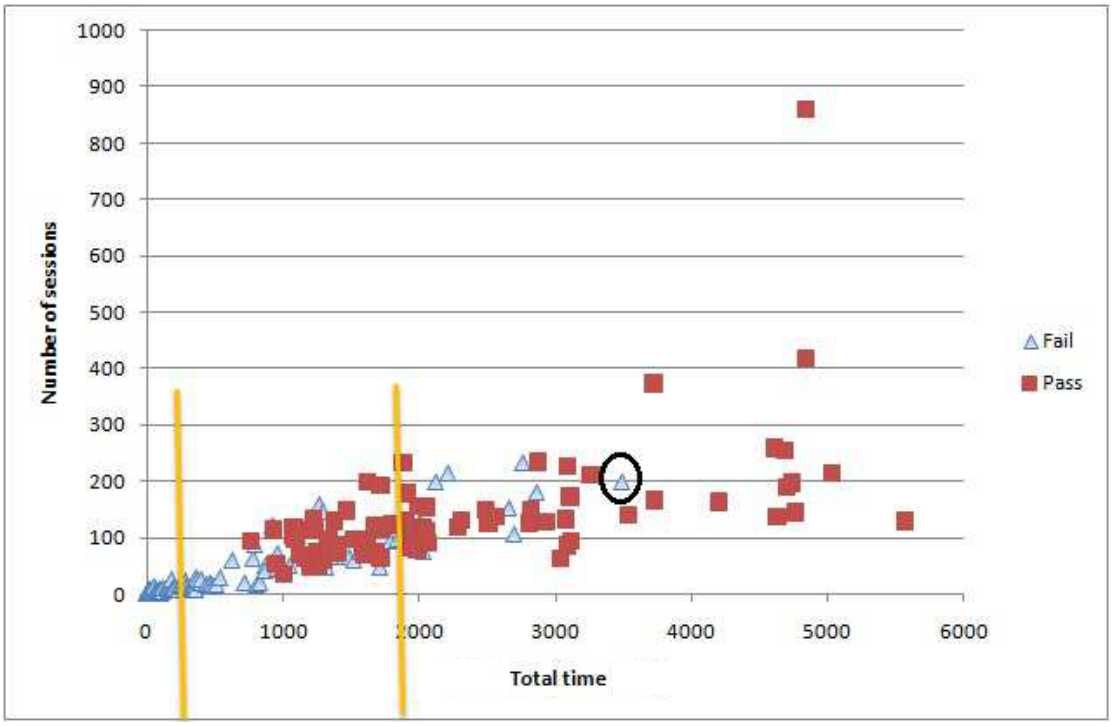


FIGURA 5.1: Distribución de los estudiantes de acuerdo al tiempo total y número de sesiones

Este tipo de comportamientos irregulares, que están presentes en mayor o menor medida en todos los cursos utilizados en los estudios, pueden ser mejor observados en la Figura 5.2, en donde se puede ver un ejemplo de los estudiantes (rodeados por un círculo) que, pese a haber suspendido, dedican una gran actividad al curso, poniendo como ejemplo el tiempo total que han estado conectados.

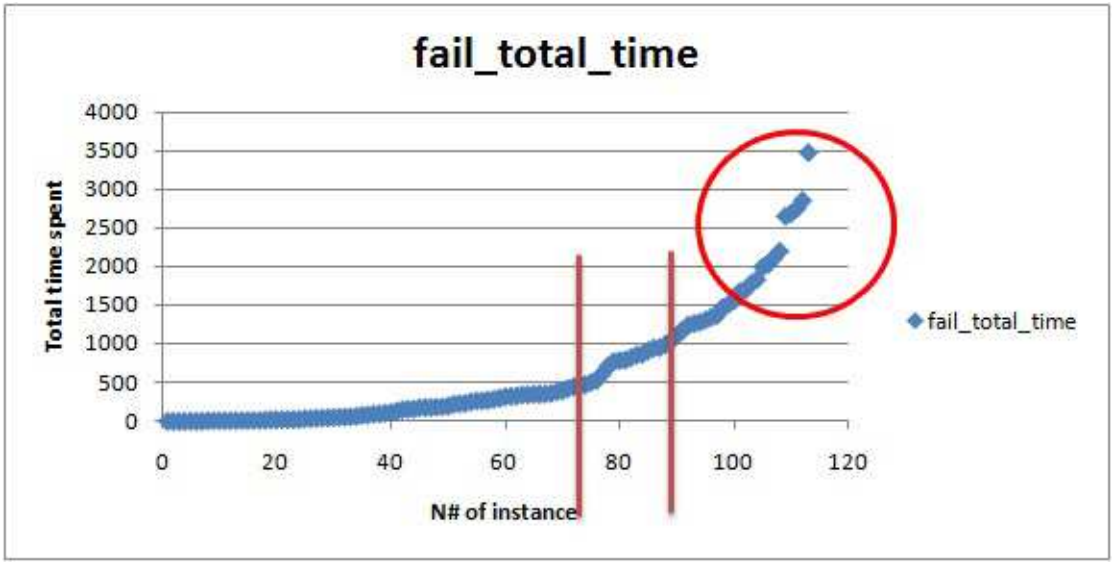


FIGURA 5.2: Distribución de los estudiantes que suspendieron de acuerdo al tiempo total

En la Figura 5.3 se muestra otro ejemplo de comportamiento irregular, esta vez tomando como medidas de actividad el número total de auto-test realizados y el tiempo dedicado a ellos en los cursos con identificadores del 6 al 11. Como puede constatarse, pueden detectarse a simple vista 3 estudiantes (rodeados por un círculo) que, pese a su alto grado de actividad, suspendieron el curso. Igualmente, existe otro estudiante (rodeado por un cuadrado) que, pese a tener una baja dedicación, aprobó el curso, algo inusual si observamos que la mayor parte de los estudiantes aprobados tienen una alta dedicación.

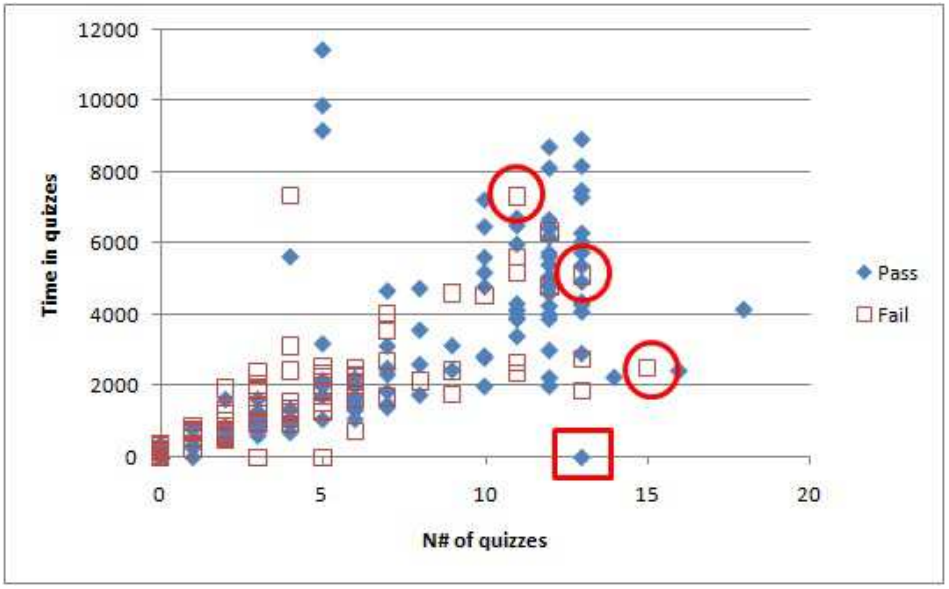


FIGURA 5.3: Distribución de los estudiantes de acuerdo al tiempo total dedicado a los tests y número de tests realizados en los cursos con identificador del seis al once

5.3. Hipótesis de partida, organización y resumen de los estudios

Uno de los objetivos principales de la presente tesis es que los profesores puedan obtener modelos de minería de datos sin ser expertos en el área, por lo que el proceso de mejora de los modelos de predicción mediante la detección de comportamientos anómalos no ha de requerir que el usuario intervenga en su configuración y ejecución. Por tanto, la **hipótesis principal** a demostrar, de la que parten los estudios de este capítulo, fue la siguiente: **¿puede construirse un proceso automático de detección y eliminación de comportamientos anómalos por parte de los estudiantes que ayude a mejorar la clasificación de su rendimiento en cursos virtuales?**.

En base a la anterior hipótesis, y siguiendo la línea de trabajo sobre detección de comportamientos anómalos, se planteó una nueva hipótesis: **¿puede el proceso de detección de comportamientos anómalos detectar durante el desarrollo del curso a aquellos estudiantes con comportamientos anómalos en riesgo de abandono, de forma que el profesor pueda proporcionarles una realimentación personalizada y evitar que abandonen del curso?**.

Con objeto de validar o refutar estas hipótesis, en el presente capítulo se muestran, de entre todos los estudios realizados y resultados obtenidos, aquellos más relevantes:

- **Estudio 1. Aplicación de técnicas de detección de *outliers* para eliminar comportamientos anómalos**: se utilizaron dos técnicas de uso extendido y bien conocidas, LOF y ECODB, para detectar y eliminar comportamientos anómalos en los conjuntos de entrenamiento. La segunda de las mencionadas técnicas, ECODB, se basa en la detección de *outliers* teniendo en cuenta la clase a la que pertenecen, por lo que entra dentro del conjunto de técnicas de tipo *class-outlier*. La técnica LOF, sin embargo, no tiene en cuenta la clase, pero ha sido incluida en este estudio debido a su gran popularidad y los numerosos trabajos existentes en los que se proponen extensiones de la misma [254] y al buen rendimiento que obtiene en ellos [255]. En este estudio se constató, por una parte, que las técnicas como LOF, que no tienen en cuenta el valor de clase para detectar comportamientos anómalos, no son efectivas con objeto de mejorar los modelos de predicción. Por otro lado, se concluyó que ECODB tiene un comportamiento inestable, no obteniendo mejoras significativas.

- **Estudio 2. Aplicación de boosting y bagging para la mejora de los modelos de predicción**: se realizó una comparativa de las mejoras obtenidas con las técnicas Adaboost, Multiboost, Bagging y RandomForest, algunas de ellas estando consideradas como técnicas "robustas" o "tolerantes" ante la presencia de *class-outliers* (ver apartado 5.1). En este estudio se constató que no se puede afirmar rotundamente que este conjunto de técnicas obtengan un buen rendimiento, al establecer que, si bien en media mejoran los modelos de predicción, observando los resultados por cada conjunto de datos se pudo ver que obtienen grandes mejoras en algunos y sin embargo empeoraron la calidad de otros.

- **Estudio 3. Ensamblado de clasificadores con Voto Mayoritario para detectar y eliminar comportamientos anómalos**: se utilizó una combinación de técnicas de clasificación con las que se detectaron y eliminaron las instancias que son mal clasificadas por la mayoría de dichas técnicas, considerando por tanto a estas instancias como comportamientos anómalos. La principal ventaja de este proceso es que se centra en las instancias mal clasificadas, al contrario que los utilizados en el estudio 1. Se concluyó que este proceso funciona considerablemente bien, mejorando los modelos de predicción notablemente, sobre todo en base a su *accuracy*. No obstante, también se demostró que tiene la desventaja de ser lento en tiempo de ejecución, además de tener el usuario que decidir el conjunto de clasificadores a utilizar en el ensamblado, así como el parámetro que determina cuando existe una mayoría suficiente de votos como para marcar a una instancia como *outlier*.

- **Estudio 4. DARIM, una nueva técnica de detección y eliminación de *outliers***: se propuso y aplicó una nueva técnica, llamada DARIM (*Distance-based Algorithm for Removing Instances that are Misclassified*), que detectase y eliminase comportamientos anómalos y mejorase la calidad de los modelos de predicción. Al igual que con el proceso de Voto Mayoritario del estudio 4, DARIM se enfoca en buscar *class-outliers* entre las instancias mal clasificadas. El objetivo fue tratar de automatizar esta tarea, de forma tal que el usuario final no tuviese que intervenir en su configuración, utilizando para ello el algoritmo de particionado k-Means sobre las instancias mal clasificadas. Se compararon los resultados con los obtenidos en el estudio 3. Por un lado, se concluyó que para conjuntos de datos con pocas instancias, como son los del ámbito educativo, es posible construir con DARIM un proceso automático, en el que el usuario no tenga que configurar los parámetros de ejecución. Por otra parte, se vio cómo, mientras que el proceso de ensamblado conseguía mejores resultados en base a medidas como el *accuracy*, DARIM obtiene mayores mejoras en la clasificación de los estudiantes suspensos (*TPrate*), además de tener un tiempo de ejecución mucho más bajo.

- **Estudio 5. DARIM, aproximación basada en densidad**: se muestran los resultados de utilizar la técnica DARIM sustituyendo el algoritmo k-Means por otro basado en densidad, DBSCAN. Se probó cómo esta nueva aproximación puede servir para mejorar los resultados de la versión inicial de DARIM en determinados

casos. No obstante, esta nueva versión se mostró más compleja para ser automatizada.

- **Estudio 6. Detección de comportamientos anómalos para evitar el bajo rendimiento o el abandono**: en este último estudio, se aplicó un proceso de detección de comportamientos anómalos basado en DARIM con objeto no de mejorar los modelos de predicción del rendimientos de los estudiantes, sino para detectar los comportamientos anómalos de los estudiantes en el desarrollo del curso. Concretamente, con este proceso se trató de identificar a aquellos estudiantes que, pese a su alta actividad, tenían un bajo rendimiento en las actividades del curso, siendo estos estudiantes los de riesgo más alto de abandono del mismo. El proceso propuesto ofreció buenos resultados, detectando con notable precisión a este tipo de estudiantes, lo que podría permitir en un futuro construir una herramienta que informase al profesor de estos comportamientos anómalos, pudiendo así personalizar la realimentación de estos estudiantes y evitar la desmotivación y abandono.

5.4. Configuración, proceso y resultados de los estudios

El presente apartado se divide en otros 6 apartados diferentes, cada uno detallando la configuración y procesos seguidos en cada uno de los estudios resumidos en el apartado 5.3. En todos los estudios del 1 al 5, se ha seguido un proceso similar para realizarlos, que se describe a continuación:

- I. Selección de los cursos y generación de los conjuntos de datos

  Se utilizaron los datos de los cursos del 1 al 24, correspondiendo 23 de los conjuntos a la actividad de los estudiantes en cada uno de los cursos por separado, excepto el 12, que sólo tiene 9 estudiantes, y generando los 4 restantes al combinar los datos de actividad de cursos con varias ediciones (como es el caso de los cursos 1, 2 y 3) o con una estructura similar (en el caso de los cursos del 6 al 12).

  Sobre estos conjuntos de datos se aplicaron diversas técnicas de pre-procesado y de selección de atributos, de forma tal que los modelos de predicción que se puedan obtener con los diferentes clasificadores tuvieran la mejor calidad posible. Todos los atributos predictores eran numéricos, y todos los atributos de clase de

cada conjunto de datos fueron binarios, siendo el suspenso la clase positiva y el aprobado la clase negativa.

- II. Aplicación de clasificadores y extracción de medidas de calidad de los modelos
  Sobre cada uno de los 27 conjuntos de datos generados, se aplicaron 31 técnicas de clasificación de diferente paradigma, utilizando validación cruzada con *10-folds* (10-CV en adelante). Los nombres de las 31 técnicas de clasificación que aparecen en las tablas de resultados de los diferentes estudios se corresponden con los nombres de sus respectivas clases Java en Weka, en cuyo código fuente se puede consultar la información al respecto de la implementación de cada uno de ellos.

  Para cada una de las ejecuciones del proceso de validación cruzada, con cada una de las técnicas y sobre cada uno de los conjuntos de datos, se almacenaron las siguientes medidas de calidad de los modelos de predicción obtenidos: *accuracy*, *TPrate* (suspensos bien clasificados), *TNrate* (aprobados bien clasificados), *f-Measure (f-M)* y área bajo la curva (AUC). En total, se realizaron un conjunto de 8370 ejecuciones (27 conjuntos * 31 clasificadores * 10 iteraciones del proceso CV) iniciales.

- III. Tratamiento de comportamientos anómalos y comparativa de resultados
  En cada uno de los estudios, se aplicaron las técnicas mencionadas anteriormente para tratar los comportamientos anómalos y mejorar los modelos de predicción. En el caso de los estudios 1, 3, 4 y 5, estas técnicas fueron ejecutadas con diferentes configuraciones y parámetros iniciales, almacenándose en cada caso los resultados obtenidos al aplicar los mismos clasificadores mencionados en el paso II sobre los mismos conjuntos de datos, y eliminando de los conjuntos de entrenamiento las instancias detectadas como anómalas. Dado que se utilizó un proceso 10-CV, esta eliminación de comportamientos anómalos se realizó en cada conjunto de entrenamiento de cada iteración.

  En el caso del estudio 2, en el que se utilizaron técnicas de boosting y bagging para mejorar los modelos, no se realizó la tarea de eliminación de *outliers*, limitándose el estudio a la aplicación de estas técnicas y a la comparación de sus resultados con respecto a los obtenidos en el paso II.

  En todos los estudios, se compararon los 8370 modelos obtenidos en el paso II con los obtenidos tras el tratamiento de los comportamientos anómalos, determinando

las mejoras obtenidas en base, por una parte, a las medidas ya mencionadas en dicho paso II. Esta comparativa se complementó, en el caso de DARIM, con la comparativa de la desviación estándar de las iteraciones del proceso de validación cruzada para cada técnica y conjunto de datos. Esto es, se trató de establecer si DARIM no solamente mejora en media en medidas como el *accuracy* sino si, además, la diferencia de *accuracy* de los diferentes modelos obtenidos en el proceso de validación cruzada se reduce al realizar el entrenamiento de cada uno de ellos tras haber eliminado los comportamientos anómalos.

Además, como medida adicional, se utilizó el tiempo de ejecución para comparar DARIM con la técnica de ensamblado con Voto Mayoritario que, como se verá en el texto, es junto con DARIM la que obtiene unas mejoras más significativas en los modelos de predicción del rendimiento de los estudiantes.

La configuración del estudio 6, diferente a la del resto de estudios, se detalla en el propio apartado que contiene los resultados del mismo.

5.4.1. Estudio 1. Aplicación de técnicas de detección de outliers para eliminar comportamientos anómalos

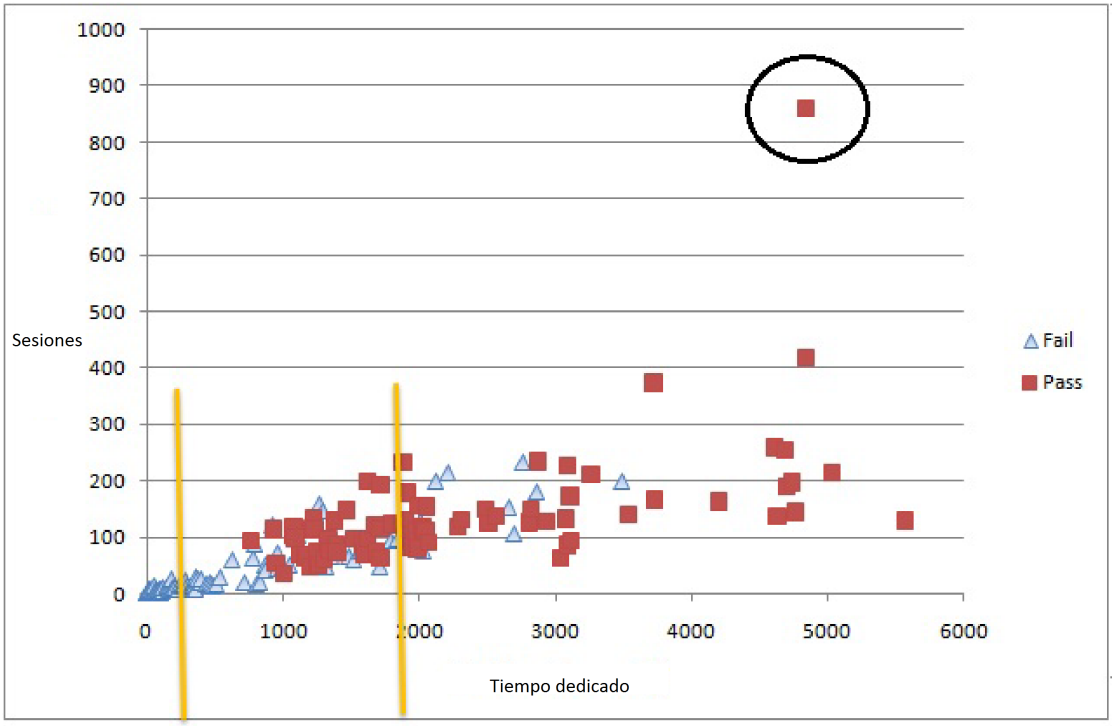
El algoritmo LOF fue ejecutado en base a diferentes configuraciones, modificando sus parámetros iniciales en cada una de ellas. El primero de estos parámetros, minPointsLo, varió en un rango de [1,3,5,7,20], mientras que el segundo de los parámetros, minPointsUp, varió en un rango de [3,7,10,15,30]. En cada una de las ejecuciones, el porcentaje de instancias a ser consideradas como *outliers* varió en un rango del 1 % al 30 % en intervalos del 5 %. En total, se realizaron 29295 ejecuciones (35 ejecuciones de LOF * 31 algoritmos * 27 conjuntos de datos), además de las 292950 ejecuciones del proceso de validación cruzada (29295 * 10 iteraciones CV). En la Tabla 5.2 se muestran los resultados, en media, obtenidos con una configuración de minPointsLo=5 y minPointsUp=10.

Tabla 5.2: Mejoras obtenidas en los modelos de predicción al aplicar LOF, por ejecución, en el estudio 1

% eliminados	Acc.	TPrate	TNrate	f-Measure	AUC
1	-0,020	7,294	-7,244	5,785	-0,010
5	-0,128	7,304	-7,691	5,723	0,048
10	-0,367	6,515	-7,808	5,111	-0,217
15	-0,381	5,506	-7,518	4,518	-0,423
20	-0,603	5,517	-8,274	4,261	-0,778
25	-1,355	3,646	-8,731	2,171	-1,964
30	-1,594	3,520	-9,324	1,845	-2,542
Estadísticas:					
Media	-0,635	5,615	-8,084	4,202	-0,841
Mediana	-0,381	5,517	-7,808	4,518	-0,423
Desv. Est.	0,607	1,569	0,736	1,604	1,017
Máx.	-0,02	7,304	-7,244	5,785	0,048
Mín.	-1,594	3,52	-9,324	1,845	-2,542

Como puede observarse, LOF no sólo no obtiene mejoras significativas en ninguna de las medidas, sino que en no pocos casos incluso empeora el rendimiento de los modelos de predicción. El motivo de esto lo podemos encontrar en la Figura 5.4, en donde se muestra la distribución de los estudiantes en uno de los cursos utilizados en la experimentación. Rodeada con un círculo se encuentra una de las instancias que LOF detecta como *outlier*, que representa a un estudiante que aprobó el curso y que dedicó un alto número de sesiones y tiempo al mismo. Este comportamiento, si bien se sale de la norma, no puede considerarse como anómalo en la clase de los aprobados, ya que la tendencia general de estos estudiantes es a dedicar una alta actividad al curso. El motivo por el cuál el algoritmo LOF detecte este tipo de comportamientos que no son anómalos en su clase se debe a que no tiene en cuenta esta clase en su proceso de detección. De hecho, la mencionada instancia es correctamente clasificada por la gran mayoría de los clasificadores utilizados.

FIGURA 5.4: Detección de comportamientos anómalos con LOF en el estudio 1



Con respeto a ECODB, esta técnica fue ejecutada con 42 configuraciones diferentes, en las que el parámetro k, que determina el número de vecinos cercanos, tomó los valores de 1, 2, 3, 5, 7 y 10; mientras que el parámetro que indica el número de *class-outliers* a detectar (y, en este caso, también a eliminar) varió en un rango del 1 al 30 % sobre el número de instancias del conjunto de datos a tratar, con intervalos del 5 %. En total, se almacenaron y analizaron los resultados de 35154 ejecuciones (42 configuraciones de ECODB * 31 clasificadores * 27 conjuntos de datos), además de los 351540 resultados del proceso de validación cruzada (35154 * 10 iteraciones CV).

Mientras que una modificación del parámetro k no parece suponer una notable variación en los resultados, el número de instancias a eliminar si muestra tener un efecto considerable en el rendimiento de los clasificadores al eliminar comportamientos anómalos del conjunto de entrenamiento. En la Tabla 5.3 se muestran las diferencias medias obtenidas en los criterios de calidad de los clasificadores tras eliminar comportamientos anómalos con ECODB utilizando un valor de k=7. Puede observarse claramente como, en términos de *accuracy* y *TPrate*, a medida que se aumenta el porcentaje de instancias a eliminar, aumenta también la mejora obtenida, hasta el 15 % de instancias eliminadas. A partir de

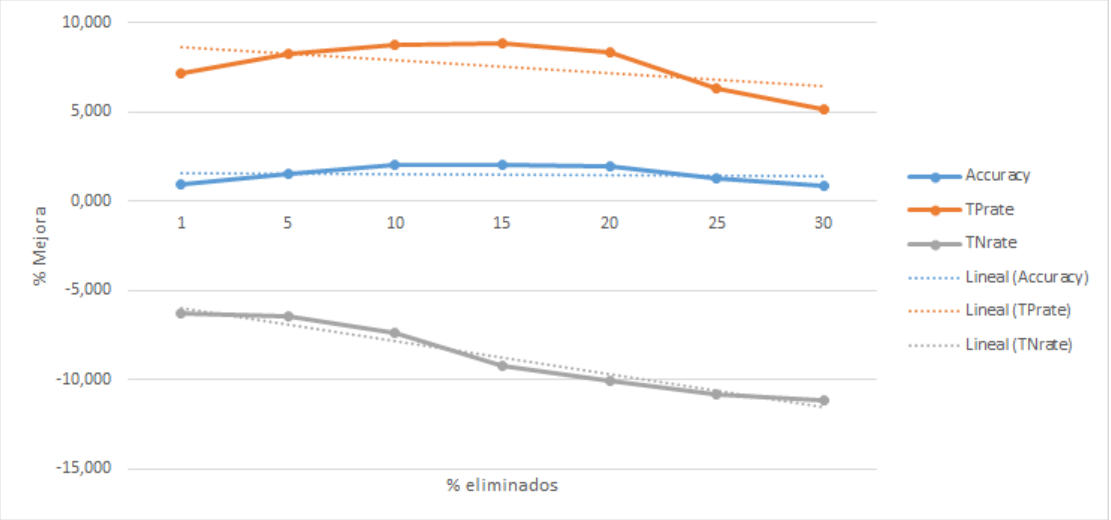
este valor, estas mejoras vuelven a disminuir en ambas medidas. Esta tendencia puede observarse de forma más clara en la Figura 5.5.

Aunque ECODB consigue mejorar los modelos de predicción en términos de *accuracy*, *TPrate* y *f-Measure*; esto no ocurre de igual manera con el *TNrate* y el *AUC*. De hecho, en todos los casos, el *TNrate* obtiene una mejora negativa, esto es, ECODB empeora esa medida de forma muy notable. No obstante, obtiene unas mejoras muy altas en el *TPrate*, con lo que se infiere claramente que al utilizar ECODB para la detección y eliminación de comportamientos anómalos, se está sacrificando la clasificación de los aprobados (*TNrate*) en aras de conseguir una mejora en la clasificación de los suspensos (*TPrate*).

Tabla 5.3: Mejoras obtenidas en los modelos de predicción al aplicar ECODB, por ejecución, en el estudio 1

% eliminados	k	Acc.	TPrate	TNrate	f-Measure	AUC
1	7	0,907	7,155	-6,310	6,106	0,748
5	7	1,514	8,276	-6,434	6,892	1,356
10	7	2,028	8,745	-7,422	7,464	1,407
15	7	2,064	8,803	-9,240	7,051	0,049
20	7	1,936	8,302	-10,037	6,543	-0,740
25	7	1,287	6,287	-10,811	4,776	-2,192
30	7	0,840	5,165	-11,181	3,312	-3,039
Estadísticas:						
Media		1,511	7,533	-8,777	6,020	-0,345
Mediana		1,514	8,276	-9,240	6,543	0,049
Desv. Est.		0,481	1,284	1,895	1,366	1,610
Máx.		2,064	8,803	-6,310	7,464	1,407
Mín.		0,840	5,165	-11,181	3,312	-3,039

FIGURA 5.5: Variación de las mejoras obtenidas por ECODB al aumentar el % de instancias eliminadas con k=7 en el estudio 1



En la Tabla 5.4 se muestran los resultados, agrupados por conjunto de datos, obtenidos con k=7 y un 10 % de instancias eliminadas. Se muestran también datos estadísticos, así como las veces que, para cada medida, ECODB mejora o empeora los modelos de predicción. En términos de *accuracy*, solamente en 2 de los 27 conjuntos de datos, dataset2 y dataset6, obtiene un resultado que empeora el modelo de predicción. No obstante, existen grandes irregularidades en el rendimiento obtenido tanto en *TPrate* como en *TNrate*, mejorando en apenas un 55,56 % (15 de 27) del total de conjuntos, poco más de la mitad en ambos casos. De hecho, existen grandes diferencias entre el mejor resultado obtenido, del 78.003 % y 34,919 % respectivamente, y el peor en cada medida, -27,291 % y -78.031 %, lo que denota una gran irregularidad en el proceso de eliminación de comportamientos anómalos con ECODB.

Tabla 5.4: Mejoras obtenidas en los modelos de predicción al aplicar ECODB con k=7 y 10 % de instancias eliminadas, por conjunto de datos, en el estudio 1

Conjunto	Acc.	TPrate	TNrate	f-Measure	AUC
dataset1	3,763	5,645	0,806	3,577	3,120
dataset2	-0,758	46,646	-68,669	38,939	-10,404
dataset3	1,168	-2,336	5,178	0,003	0,622
dataset4	0,783	2,076	-6,298	0,621	-2,809
dataset5	2,800	37,653	-35,532	35,444	-0,642
dataset6	-0,364	56,930	-68,147	44,994	-3,254
dataset7	3,391	10,868	-5,072	5,811	2,812
dataset8	3,480	-4,839	7,320	2,405	-0,010
dataset9	3,317	-6,048	6,041	2,131	1,239
dataset10	0,587	78,003	-78,031	75,454	1,863
dataset11	0,978	-2,447	3,662	-0,230	2,976
dataset12	3,226	-8,871	16,359	-10,593	5,530
dataset13	1,882	2,925	-1,843	1,540	0,763
dataset14	2,500	-27,291	34,919	-23,117	5,737
dataset15	1,847	3,211	-5,101	1,318	-3,172
dataset16	1,125	23,065	-30,081	20,182	-5,628
dataset17	2,730	-5,806	8,065	-1,026	4,556
dataset18	6,072	21,774	-9,901	16,678	8,826
dataset19	2,419	0,587	3,605	2,597	4,796
dataset20	3,763	-0,461	5,049	6,292	6,802
dataset21	2,225	-2,151	4,194	2,155	3,259
dataset22	0,669	12,408	-9,382	-1,577	1,077
dataset23	2,621	1,466	3,850	2,540	3,177
dataset24	1,241	12,341	-9,729	2,055	2,247
dataset25	1,310	-18,272	22,168	-22,147	1,740
dataset26	1,171	-0,025	2,951	0,748	1,786
dataset27	0,798	-0,932	3,213	-5,265	0,985
Estadísticas:					
Media	2,028	8,745	-7,422	7,464	1,407
Mediana	1,882	1,466	2,951	2,131	1,786
Desv. Est.	1,472	22,880	26,867	20,535	3,988
Máx.	6,072	78,003	34,919	75,454	8,826
Mín.	-0,758	-27,291	-78,031	-23,117	-10,404
N# de veces que empeora, obtiene el mismo resultado, o mejora:					
Empeora	2	12	12	7	7
Mejora	25	15	15	20	20

En el apartado 5.4.4 se expondrá una comparativa de los comportamientos anómalos detectados por ECODB y DARIM, y se explicará el motivo por el cual ECODB no tiene un buen rendimiento en la tarea de detección de comportamientos anómalos para la mejora de los modelos de predicción.

5.4.2. Estudio 2. Aplicación de boosting y bagging para la mejora de los modelos de predicción

Las Tablas 5.5, 5.6, 5.7 y 5.8 muestran, respectivamente, las mejoras obtenidas respecto a los modelos de predicción iniciales al aplicar las técnicas RandomForest, MultiBoost, Adaboost y Bagging, los tres últimos con DecisionStump como algoritmo base. Como apoyo para una mejor interpretación y comparativa, se muestra en la Figura 5.6 una comparativa gráfica de las veces en que cada una de las técnicas ha conseguido mejora en términos de *accuracy*, y en cuántas de estas ocasiones esta mejora está entre el 0 y 1 %, entre el 1 y 2 %, o es mayor o igual al 2 %.

Respecto al *accuracy*, RandomForest alcanza los mejores resultados, con un 89 % de conjuntos en los que se consigue mejorar los modelos de predicción, de los cuáles un 62 % obtienen una mejoría igual o superior al 2 %. Le sigue muy de cerca Bagging, con un 85 % de conjuntos en los que se mejora. Más lejos quedan Adaboost y Multiboost, ambos con un 67 % de conjuntos con mejoría, y ambos también con un 61 % en los que se iguala o supera el 2 %.

No obstante, no existe esta diferencia de resultados si nos atenemos a otras medidas del rendimiento, como el *TPrate*. En este caso, RandomForest, MultiBoost, Adaboost y Bagging empeoran en 11, 15, 10 y 10 conjuntos de 27 respectivamente, un número considerablemente alto en todos los casos. Al igual que ocurría con ECODB, las mejoras obtenidas en estos casos son muy irregulares, y en muchos casos la mejora en la clasificación de estudiantes suspensos o *TPrate* implica un sacrificio en el rendimiento de la clasificación de estudiantes aprobados o *TNrate*, y viceversa.

En base a estos resultados, se puede concluir que, si bien las técnicas utilizadas, y de las que en la literatura existentes algunos autores afirman que son robustas o tolerantes a *class-outliers*, pueden mejorar los modelos de predicción en determinados casos, esta

afirmación no se puede extender a todos ellos, por lo menos en lo que concierne a conjuntos de datos del campo educativo.

Tabla 5.5: Estadísticas de las mejoras obtenidas en los modelos de predicción con RandomForest, por conjunto de datos, en el estudio 2

Medida	Acc.	TPrate	TNrate	f-Measure	AUC
Estadísticas:					
Media	2,598	1,216	3,405	4,550	6,031
Mediana	2,055	0,717	3,950	1,941	6,162
Desv. Est.	3,664	4,332	6,133	6,160	5,892
Máx.	16,888	10,081	22,939	22,606	22,558
Mín.	-4,480	-10,411	-9,063	-4,849	-4,123
N# de veces que empeora, obtiene el mismo resultado, o mejora:					
Empeora	3	11	6	3	3
Mejora	24	16	21	24	24
N# de veces que la mejora es mayor que 0, 1 o 2 %:					
Entre 0 y 1	4	3	0	4	3
Entre 1 y 2	5	4	6	7	0
Mayor de 2	15	9	15	13	21

Tabla 5.6: Estadísticas de las mejoras obtenidas en los modelos de predicción con MultiBoost, por conjunto de datos, en el estudio 2

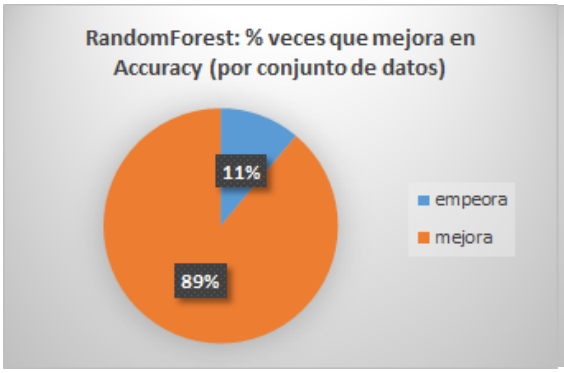
Medida	Acc.	TPrate	TNrate	f-Measure	AUC
Estadísticas:					
Media	1,486	-0,580	3,740	2,066	4,005
Mediana	1,613	-1,613	3,109	1,300	4,591
Desv. Est.	5,186	8,383	6,133	7,983	4,792
Máx.	16,888	21,935	22,939	23,883	12,917
Mín.	-8,031	-18,548	-6,190	-17,394	-7,866
N# de veces que empeora, obtiene el mismo resultado, o mejora:					
Empeora	9	15	5	10	4
Mejora	18	12	22	17	23
N# de veces que la mejora es mayor que 0, 1 o 2 %:					
Entre 0 y 1	2	1	5	2	1
Entre 1 y 2	5	3	2	3	1
Mayor de 2	11	8	15	12	21

Tabla 5.7: Estadísticas de las mejoras obtenidas en los modelos de predicción con AdaBoost, por conjunto de datos, en el estudio 2

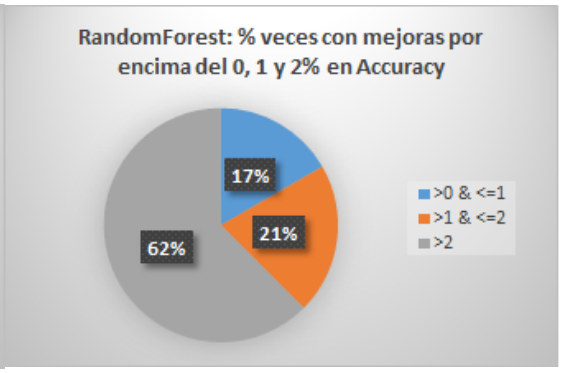
Medida	Acc.	TPrate	TNrate	f-Measure	AUC
Estadísticas:					
Media	1,985	2,603	1,498	4,473	4,891
Mediana	0,806	1,414	1,267	1,922	5,331
Desv. Est.	3,737	8,451	5,306	10,582	5,491
Máx.	14,640	31,452	14,217	49,272	24,555
Min.	-3,175	-12,500	-10,653	-8,274	-6,608
N# de veces que empeora, obtiene el mismo resultado, o mejora:					
Empeora	9	10	10	6	3
Mejora	18	17	17	21	24
N# de veces que la mejora es mayor que 0, 1 o 2 %:					
Entre 0 y 1	5	3	3	6	3
Entre 1 y 2	2	2	2	2	1
Mayor de 2	11	12	12	13	20

Tabla 5.8: Mejoras obtenidas en los modelos de predicción con Bagging, por conjunto de datos, en el estudio 2

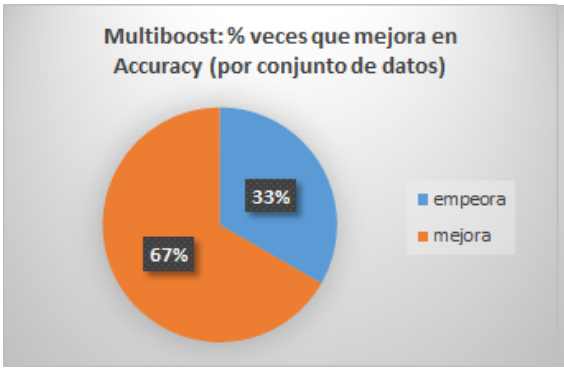
Medida	Acc.	TPrate	TNrate	f-Measure	AUC
Estadísticas:					
Media	2,244	0,454	1,030	1,396	4,114
Mediana	2,401	3,955	1,902	3,372	3,740
Desv. Est.	3,486	13,967	7,634	13,451	5,550
Máx.	7,771	21,935	18,145	22,606	17,003
Mín.	-7,762	-52,080	-15,136	-54,916	-7,387
N# de veces que empeora, obtiene el mismo resultado, o mejora:					
Empeora	4	10	9	5	8
Mejora	23	17	18	22	19
N# de veces que la mejora es mayor que 0, 1 o 2 %:					
Entre 0 y 1	4	1	2	3	1
Entre 1 y 2	3	1	5	4	2
Mayor de 2	16	15	11	15	16



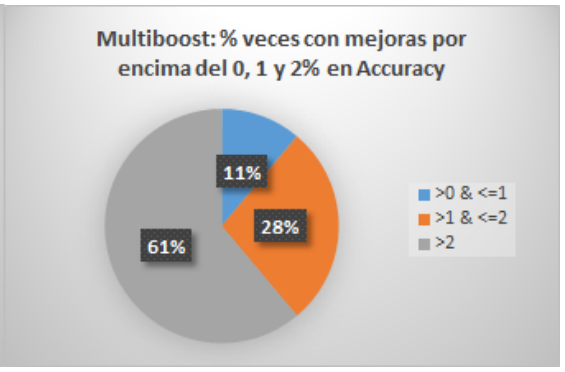
(A) % Mejora RandomForest



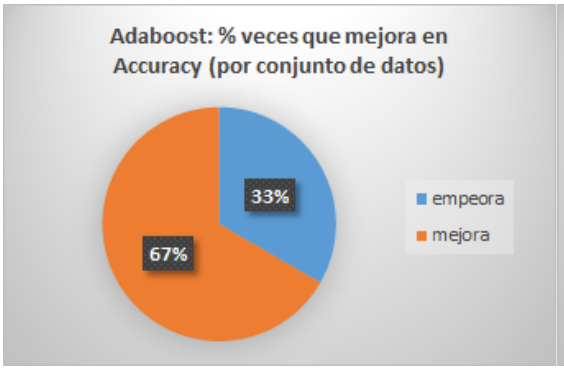
(B) % Rangos RandomForest



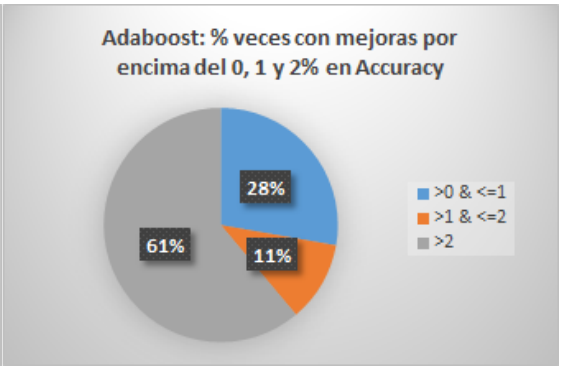
(C) % Mejora Multiboost



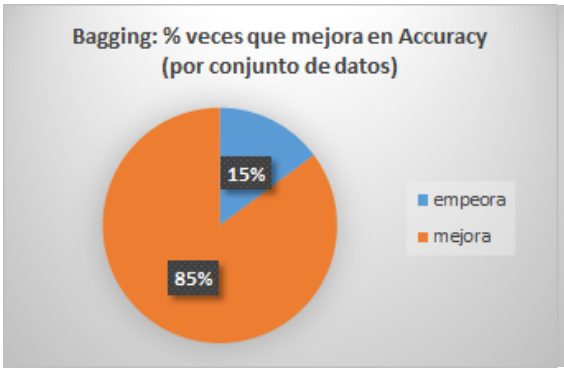
(D) % Rangos Multiboost



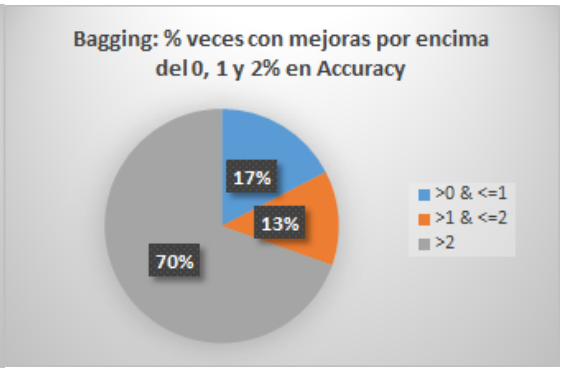
(E) % Mejora Adaboost



(F) % Rangos Adaboost



(G) % Mejora Bagging



(H) % Rangos Bagging

FIGURA 5.6: Porcentaje de veces que los algoritmos de boosting y bagging mejoran en términos de accuracy en el estudio 2

5.4.3. Estudio 3. Ensamblado de clasificadores con Voto Mayoritario para detectar y eliminar comportamientos anómalos

En este estudio, se utilizó un proceso de Voto Mayoritario para detectar y eliminar *outliers*, en el que se hizo uso de un conjunto de clasificadores con los que determinar que instancias eran *class-outliers* en base a aquellas que hubiesen sido mal clasificadas por un subconjunto de estos. Tanto la selección de clasificadores incluidos en el proceso como el número de votos necesarios para clasificar las instancias como *class-outliers* (denominado umbral o *threshold* de ahora en adelante) fueron, por tanto, parámetros de entrada del proceso.

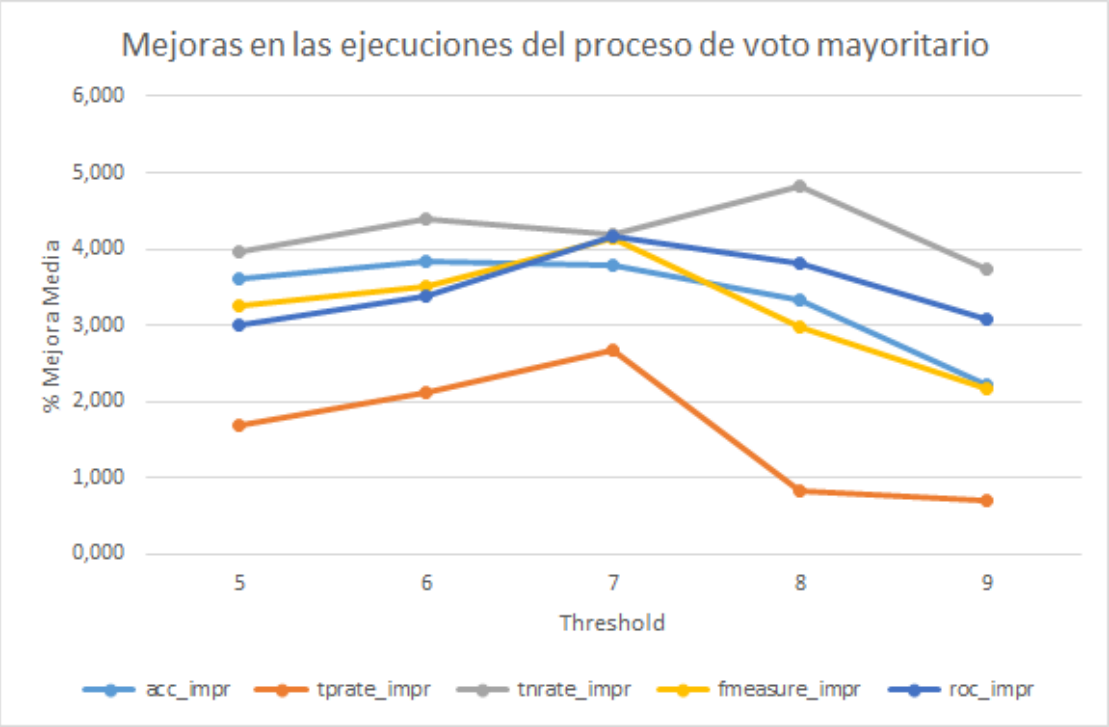
En cuanto a la selección de clasificadores, durante el desarrollo de esta línea de investigación se probó con varias combinaciones de clasificadores, bien propuestas en diversos trabajos de los mencionados en el apartado de estado del arte de este capítulo, o bien seleccionadas en base a diferentes criterios, como la inclusión de técnicas de diferente paradigma. En este apartado, se muestran los resultados obtenidos con un conjunto de 9 clasificadores propuestos por Smith et al. [256], combinación que obtuvo los mejores resultados de entre todas las investigadas. Estos clasificadores son Ridor, NaïveBayes, RandomForest, RIPPER (JRip en Weka), NNge, 5-NN, Locally Weighted Learning, C4.5 (J48 en Weka) y MultilayerPerceptron.

Como umbral, inicialmente se utilizaron valores en un rango del 2 al 9, esto es, siendo necesarios en cada ejecución entre 2 y 9 votos para marcar una instancia como *class-outlier*. No obstante, con umbral de 2, 3 y 4, dado que el número de votos necesarios es muy bajo, resultó imposible ejecutar el proceso completo debido a que en algunos conjuntos de datos se eliminaban demasiadas instancias y no era posible llevar a cabo la validación cruzada. En la Tabla 5.9 se incluyen las mejoras medias obtenidas para los valores de umbral 5, 6, 7, 8 y 9. Se adjunta como apoyo a visualización de los resultados la Figura 5.7, que contiene una representación gráfica de la evolución de las mejoras medias por ejecución.

En todas las ejecuciones, se obtienen mejoras en todas las medidas mostradas, salvo en el caso de la desviación estándar del *f-Measure*, que empeora ligeramente con un umbral de 9. A tenor de estos resultados, y en comparación con los obtenidos en los estudios 1 y 2, puede afirmarse que el proceso de Voto Mayoritario es notablemente

eficiente al detectar y eliminar comportamientos anómalos de los estudiantes. También puede observarse que, en media, el *accuracy* alcanza sus mayor mejora con un umbral de 6, mientras que el *TPrate*, el *f-Measure* y el *AUC* lo alcanzan con un umbral de 7, y el *TNrate* con un umbral de 8. Esto denota que existe un crecimiento en las mejoras obtenidas al aumentar el umbral hasta un determinado punto en el que el valor de las mejoras vuelve a disminuir. Puede concluirse, por tanto, que para obtener resultados óptimos con este proceso, el valor de umbral ha de estar equilibrado, estando por encima de la mitad del número de clasificadores utilizados, pero por debajo del número total.

FIGURA 5.7: Variación de las mejoras obtenidas por el proceso de Voto Mayoritario al aumentar el umbral en el estudio 3



En la Tabla 5.10 se muestran los resultados del proceso de Voto Mayoritario por conjunto de datos, con un umbral de 6, que fue la configuración que mejor resultados obtuvo en *accuracy*, y como apoyo a la interpretación de estos resultados, se muestran en la Figura 5.25, en porcentajes, las veces en las que el proceso mejora los modelos obtenidos, además de en cuántas de estas ocasiones estas mejoras están entre el 0 y 1 %, 1 y 2 %, o son mayores o iguales al 2 %.

Es de destacar que solamente en un conjunto de datos, dataset10, el proceso de Voto Mayoritario empeora el resultado en términos de *accuracy*, siendo esta pérdida menor del 1 %. De hecho, de entre los 26 conjuntos restantes, en un 77 % de estos (20/26) se

Tabla 5.9: Mejoras obtenidas en los modelos de predicción al aplicar el proceso de Voto Mayoritario por ejecución en el estudio 3

	Mejora media por ejecución:					**Mejora media de la desviación estándar por ejecución:**					
N# votos	Acc.	TPrate	TNrate	f-Measure	AUC	Acc.	TPrate	TNrate	f-Measure	AUC	Tiempo
5	3,620	1,687	3,956	3,267	3,004	0,883	1,358	4,458	0,580	1,476	3026,368
6	3,849	2,118	4,396	3,500	3,383	1,200	1,559	4,101	0,713	1,403	3013,108
7	3,796	2,662	4,184	4,135	4,170	1,127	0,978	3,685	0,376	1,045	3015,608
8	3,342	0,836	4,834	2,972	3,810	0,928	0,488	2,842	0,023	0,561	3016,073
9	2,214	0,692	3,734	2,174	3,075	0,485	0,059	1,280	-0,071	0,242	3041,593
Estadísticas:											
Media	3,364	1,599	4,221	3,210	3,488	0,925	0,888	3,273	0,324	0,945	3022,550
Mediana	3,620	1,687	4,184	3,267	3,383	0,928	0,978	3,685	0,376	1,045	3016,073
Desv. Est.	0,602	0,750	0,378	0,644	0,444	0,250	0,553	1,133	0,305	0,478	10,550
Máx.	3,849	2,662	4,834	4,135	4,170	1,200	1,559	4,458	0,713	1,476	3041,593
Mín.	2,214	0,692	3,734	2,174	3,004	0,485	0,059	1,280	-0,071	0,242	3013,108

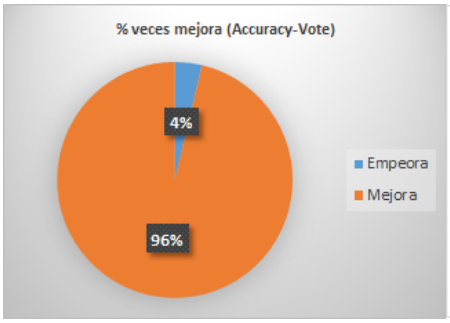
consiguen mejoras superiores al 2 %, y solamente en un caso estas mejoras están por debajo del 1 %, superando así los resultados obtenidos con las técnicas de los estudios 1 y 2. De forma similar, se consigue mejorar en el 85, 93 y 89 % de los conjuntos de datos las medidas *TNrate*, *f-Measure* y AUC. Solamente en el caso del *TPrate* la mejora no es tan notable, alcanzado a dos tercios de los conjuntos.

Otro hecho notable es que también se consigue mejorar mayoritáriamente, en 24 y 25 conjuntos respectivamente, la desviación estándar del *accuracy* y *TNrate* de los modelos obtenidos en el proceso de validación cruzada. No obstante, vuelve a suceder que con el *TPrate* sólo se obtienen mejoras en 17 de los 27 conjuntos.

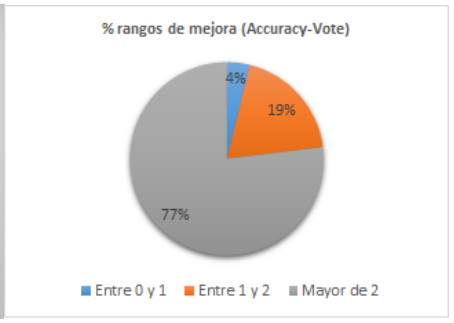
Tabla 5.10: Mejoras obtenidas en los modelos de predicción al aplicar el proceso de Voto Mayoritario, por conjunto de datos, en el estudio 3

	Mejora media por conjunto:					**Mejora media de la desviación estándar:**					
Conjunto	Acc.	TPr.	TNr.	f-M	AUC	Acc.	TPr.	TNr.	f-M	AUC	Tiempo
dataset1	5,332	9,897	-1,843	5,453	3,434	2,304	7,087	-2,592	5,191	1,122	952,161
dataset2	0,996	5,610	0,261	6,242	0,515	0,142	-0,403	0,309	0,832	0,142	18030,645
dataset3	6,056	4,190	8,192	5,550	4,891	-1,701	0,500	3,791	-1,073	-0,799	5657,226
dataset4	1,494	3,394	-8,909	1,077	-5,088	3,232	4,867	11,360	2,111	10,966	1943,968
dataset5	4,484	10,685	2,263	11,369	2,279	-0,825	0,845	1,319	1,576	-1,551	6044,710
dataset6	1,769	-8,685	4,542	-4,070	2,329	1,066	2,024	3,152	-0,086	2,400	1052,226
dataset7	3,309	-0,949	6,598	3,308	2,350	1,402	0,185	10,919	-0,345	-0,894	770,000
dataset8	6,027	4,570	6,700	8,531	5,387	3,023	3,743	6,901	4,100	0,900	878,613
dataset9	4,180	4,234	4,164	7,617	5,768	1,059	2,876	3,096	3,076	2,244	1593,258
dataset10	-0,978	-15,323	1,001	-16,697	-7,369	1,026	13,032	7,059	14,035	10,600	751,387
dataset11	6,549	1,669	10,375	5,916	5,963	0,939	-3,572	4,116	-4,650	-0,078	1364,387
dataset12	3,226	1,843	5,645	2,540	5,415	1,873	-2,387	1,036	-3,375	0,557	244,581
dataset13	6,552	10,495	-7,527	5,105	1,958	5,220	8,592	8,643	4,666	11,860	3225,290
dataset14	4,960	-4,241	24,069	1,975	9,569	0,629	-1,107	7,308	-0,639	2,860	1450,194
dataset15	3,361	5,421	-7,127	2,342	-5,787	1,584	3,004	3,189	1,086	-0,852	5602,419
dataset16	1,650	7,258	0,369	4,824	0,858	0,920	1,587	-0,866	1,201	-0,220	1524,742
dataset17	5,211	-1,935	9,677	2,940	7,379	1,349	1,840	2,511	0,415	-4,652	226,387
dataset18	13,662	15,323	12,186	14,657	16,420	2,142	-0,415	3,603	-2,261	-2,596	399,548
dataset19	3,111	-0,293	5,313	3,111	7,504	2,364	-0,104	8,450	-0,800	2,195	346,677
dataset20	3,763	-0,461	5,049	6,292	6,802	2,102	-0,025	11,590	-1,908	-0,765	542,968
dataset21	3,226	0,000	4,677	4,033	4,583	1,712	-1,527	4,226	-2,760	2,478	914,387
dataset22	1,354	-3,658	8,284	0,608	0,802	0,255	-0,779	5,157	-0,827	0,510	8887,516
dataset23	1,310	0,978	1,665	1,278	1,949	0,074	0,618	1,034	0,183	-0,448	1444,387
dataset24	3,176	1,955	4,435	3,093	4,235	0,600	0,647	3,593	0,181	1,428	1270,129
dataset25	3,730	2,314	7,348	2,564	3,565	0,013	0,197	0,221	-0,551	0,520	2284,161
dataset26	4,198	3,374	5,425	3,336	3,253	0,919	1,361	1,517	0,909	0,667	11720,645
dataset27	2,224	-0,474	5,854	1,493	2,375	-1,038	-0,587	0,086	-1,050	-0,720	2231,290
Estadísticas:											
Media	3,849	2,118	4,396	3,500	3,383	1,200	1,559	4,101	0,713	1,403	3013,108
Mediana	3,361	1,955	5,049	3,308	3,434	1,059	0,618	3,593	0,181	0,520	1444,387
Desv. Est.	2,705	6,176	6,545	5,396	4,719	1,414	3,510	3,807	3,527	3,872	4107,392
Máx.	13,662	15,323	24,069	14,657	16,420	5,220	13,032	11,590	14,035	11,860	18030,645
Mín.	-0,978	-15,323	-8,909	-16,697	-7,369	-1,701	-3,572	-2,592	-4,650	-4,652	226,387
N# de veces que empeora, obtiene el mismo resultado, o mejora:											
Empeora	1	9	4	2	3	3	10	2	13	11	-
Mejora	26	18	23	25	24	24	17	25	14	16	-
N# de veces que la mejora es mayor que 0, 1 o 2 %:											
Entre 0 y 1	1	2	2	1	3	9	6	3	5	6	-
Entre 1 y 2	5	3	2	4	2	8	3	4	3	2	-
Mayor de 2	20	13	19	20	19	7	8	18	6	8	-

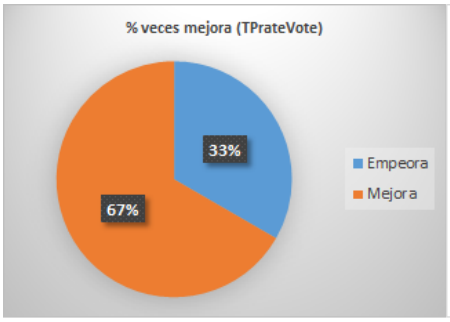
FIGURA 5.8: Porcentajes de mejora por medida aplicando el proceso Voto Mayoritario en el estudio 3



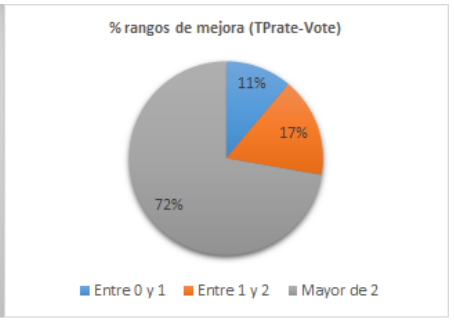
(A) % de mejoras en Acc.



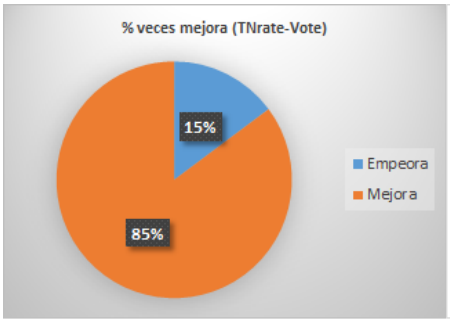
(B) % de rangos de mejora en Acc.



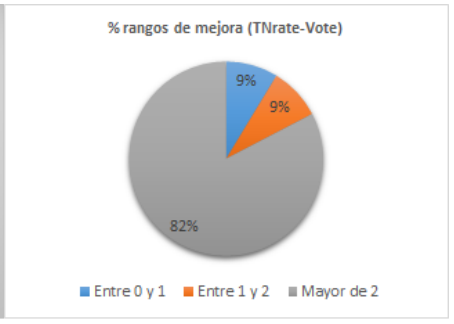
(C) % de mejoras en TPrate



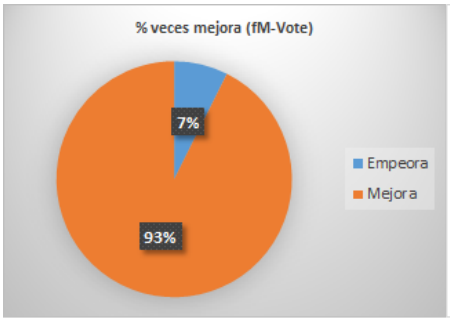
(D) % de rangos de mejora en TPrate



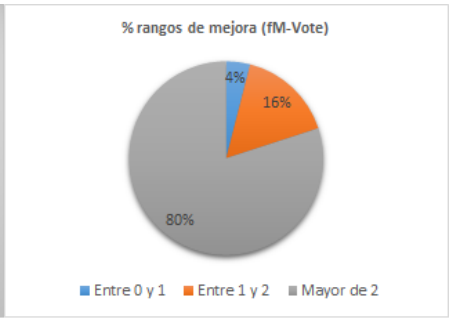
(E) % de mejoras en TNrate



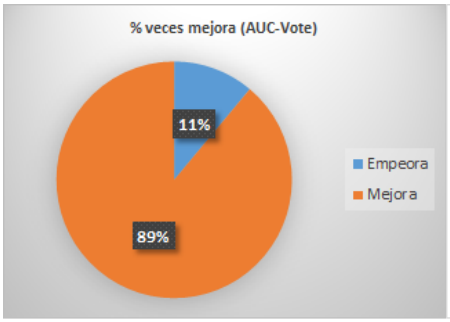
(F) % de rangos de mejora en TNrate



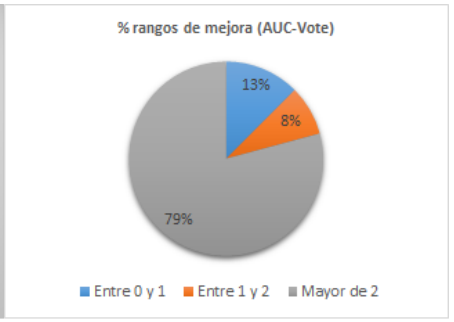
(G) % de mejoras en f-M



(H) % de rangos de mejora en f-M



(I) % de mejoras en AUC



(J) % de rangos de mejora en AUC

La siguiente Tabla 5.11 contiene los mismos resultados, agrupados por clasificador en vez de por conjunto de datos. Y al igual que ocurre al agrupar por conjuntos de datos, el *accuracy*, *TNrate*, *f-Measure* y *AUC* obtienen muy buenos resultados, alcanzándose el 100 % de clasificadores mejorados en las dos primeras medidas, teniendo en ambos casos también que en un 90,32 % (28/31) de los clasificadores se obtienen mejoras por encima del 2 %. El *TPrate*, al contrario, vuelve a obtener un peor resultado que el resto de medidas, al obtenerse mejoras en un 83.87 % (26/31). Si bien es un resultado bastante bueno, se encuentra notablemente por debajo del resto de medidas, al igual que ocurría al agrupar los resultados por conjuntos de datos. Este hecho denota que, si bien el proceso por Voto Mayoritario muestra ser notablemente eficiente, tiene importantes desventajas, y es que parece sacrificar en ocasiones la clasificación de los estudiantes suspensos en aras de mejorar la de los aprobados.

Tabla 5.11: Mejoras obtenidas en los modelos de predicción al aplicar el proceso de Voto Mayoritario, por clasificador, en el estudio 3

Clasificador	Acc.	TPr.	TNr.	f-M	AUC
BayesNet	3,497	2,551	4,018	4,071	6,627
NaiveBayes	6,167	-5,387	6,352	2,243	-1,056
Logistic	6,197	3,501	7,572	5,245	2,685
MLP	4,438	1,582	6,303	4,431	4,190
RBFNetwork	3,806	-2,524	7,493	0,460	-0,021
SimpleLogistic	2,258	0,826	4,142	2,517	1,967
SMO	1,046	0,901	2,222	1,577	1,561
SPegasos	5,154	2,402	3,813	4,037	3,107
IB1	6,273	1,854	5,698	3,772	3,776
IBk	2,584	2,610	0,770	0,931	-0,945
KStar	6,432	1,999	6,135	5,336	5,315
LWL	2,810	6,255	0,985	5,781	3,647
ConjunctiveRule	2,473	3,609	2,403	4,036	3,861
DecisionTable	3,460	6,008	2,784	5,762	6,419
DTNB	3,589	3,045	4,464	4,118	5,274
JRip	3,126	-0,201	4,940	2,159	5,405
NNge	4,855	0,927	4,542	2,069	2,734
OneR	1,465	-2,265	3,226	-2,553	0,480
PART	4,108	0,855	6,060	3,990	2,918
Ridor	2,271	3,082	0,288	1,220	1,685
ADTree	5,874	3,881	5,190	4,439	2,939
BFTree	5,484	6,202	5,335	7,588	5,538
DecisionStump	3,671	4,722	4,303	5,875	4,580
FT	3,269	1,285	4,827	2,928	4,106
J48	4,701	2,210	6,782	4,635	4,795
LADTree	2,605	5,455	3,145	4,462	3,138
LMT	2,416	1,235	3,750	2,578	2,379
NBTree	4,201	3,757	4,879	5,538	4,737
RandomTree	5,430	0,444	4,987	2,633	2,413
REPTree	0,985	-0,104	2,998	0,144	3,049
SimpleCart	4,686	4,946	5,865	6,464	7,567
N# de veces que empeora, obtiene el mismo resultado, o mejora:					
Empeora	0	5	0	1	3
Mejora	31	26	31	30	28
N# de veces que la mejora es mayor que 0, 1 o 2 %:					
Entre 0 y 1	1	5	3	3	1
Entre 1 y 2	2	5	0	2	3
Mayor de 2	28	16	28	25	24

En la Tabla 5.12, y como apoyo en la Figura 5.9, se muestran los resultados agrupados según el *accuracy* inicial obtenido en cada ejecución de un clasificador sobre un conjunto de datos, esto es, el *accuracy* obtenido antes de aplicar el proceso de Voto Mayoritario para eliminar comportamientos anómalos. La última de las columnas indica el porcentaje de conjuntos de datos con un *accuracy* inicial en el rango descrito en la fila.

A simple vista, puede verse claramente que, cuanto más bajo es el *accuracy* inicial del modelo, más alta es la mejora obtenida al eliminar comportamientos anómalos. No obstante, sólo en un 4 % de las ejecuciones el *accuracy* inicial es menor del 60 %, por lo que estos casos no son tan representativos como los del resto de rangos.

Fijándonos en el *TPrate*, podemos observar que se produce un notable empeoramiento de los resultados en las ejecuciones con un *accuracy* inicial por encima del 80 %, lo que explica por qué en las anteriores agrupaciones por conjuntos y por clasificadores esta medida obtenía peores resultados que el resto. De ello se puede concluir que el proceso de Voto Mayoritario, si bien es muy eficiente en la mayor parte de los casos y en base a la mayoría de las medidas de calidad de los modelos, comienza a perder de forma notable su eficacia cuando el *accuracy* inicial de estos modelos aumenta, debido fundamentalmente a que el número de instancias mal clasificadas por las técnicas incluidas en el proceso de Voto Mayoritario disminuye, lo que le expone a cometer más errores. Otro de los motivos que explica este comportamiento, y que refuerza al ya mencionado, es que en una buena parte de los conjuntos de datos con una mayor tasa de estudiantes aprobados que de suspensos, el porcentaje de estudiantes aprobados con respecto al total es considerablemente alto, algo que puede observarse por ejemplo en los cursos con identificadores 5, 8, 12, 17, 19 y 23, con tasas de aprobados superiores al 70 % e incluso al 80 %. Por ello, es más probable que el proceso de Voto Mayoritario detecte un mayor número de comportamientos anómalos en los estudiantes aprobados. Como consecuencia, el *TNrate* mejora notablemente más que el *TPrate*.

TABLA 5.12: Mejoras obtenidas en los modelos de predicción al aplicar el proceso de Voto Mayoritario, por accuracy inicial, en el estudio 3

Rango Acc.	Acc.	TPrate	TNrate	f-Measure	AUC	% de conjuntos
menor de 50	25,776	36,357	8,078	34,121	24,888	1 %
50 a 60	16,150	15,470	11,705	14,201	12,167	3 %
60 a 70	7,063	5,078	8,089	5,853	6,444	16 %
70 a 80	4,403	2,445	5,451	4,510	3,363	28 %
80 a 90	1,867	-0,076	2,734	1,790	1,814	38 %
mayor de 90	0,612	-0,731	0,822	-0,651	1,116	14 %

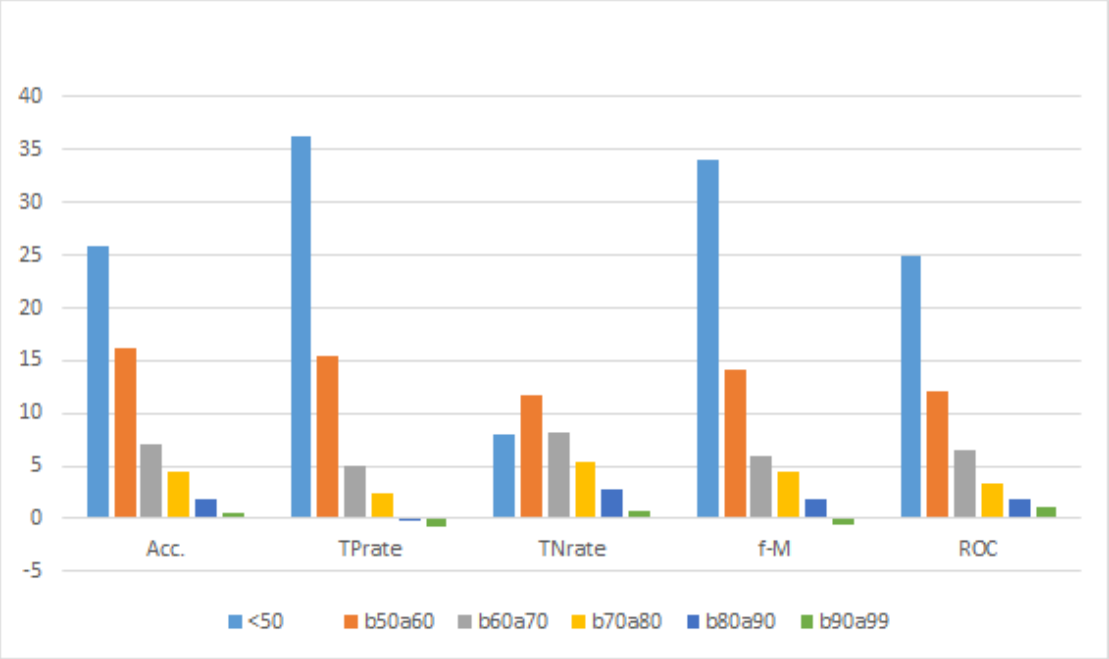


FIGURA 5.9: Mejoras por rango de accuracy inicial (Voto Mayoritario) en el estudio 3

5.4.4. Estudio 4. DARIM: una nueva técnica de detección y eliminación de outliers

Los resultados obtenidos en los estudios 1 y 2 muestran que tanto las técnicas de detección de *outliers* propuestas en la literatura, así como las consideradas como tolerantes a *class-outliers*, no obtienen un buen rendimiento de cara a mejorar los modelos de predicción del rendimiento de los estudiantes. En el estudio 3, no obstante, se ha encontrado que las técnicas basadas en Voto Mayoritario si son considerablemente eficaces en esta tarea, si bien sus resultados demuestran que tienen ciertos problemas, siendo el más

destacado el que, en presencia de conjuntos de datos con estudiantes mayoritáriamente aprobados, no siempre obtienen buenos resultados en la clasificación de los estudiantes suspensos. Dado que son estos estudiantes, los que suspenden, aquellos a los que es más imperativo identificar correctamente en el proceso de predicción, se hace necesario buscar o desarrollar otras técnicas que mejoren su clasificación. Más aún, todas las técnicas empleadas anteriormente requieren de parámetros de configuración que el usuario ha de manipular antes de ejecutar el proceso de eliminación de comportamientos anómalos. Dado que el objetivo de esta tesis es poder ofrecer a profesores de cursos virtuales la posibilidad de obtener modelos de predicción sin necesidad de conocimientos en minería de datos, las técnicas o procesos desarrollados han de realizar el proceso de forma automática, sin necesidad de configurar ninguno de sus parámetros.

El nombre del proceso propuesto para la detección y eliminación automática de comportamientos anómalos, DARIM, es un acrónimo de *Distance-based Algorithm to Remove Instances that Are Misclassified* o, en castellano, *Algoritmo basado en Distancias para Eliminar Instancias Mal clasificadas*.

DARIM, por tanto, ha de ser entendido como una técnica o proceso que realiza la tarea de detección de *outliers* o comportamientos anómalos teniendo en cuenta la clase de las instancias, y con objeto de obtener modelos de clasificación más precisos. El proceso funciona en 3 pasos: en un primer paso, se construye un modelo de clasificación inicial con todas las instancias del conjunto de datos; en segundo lugar, se detectan los comportamientos anómalos o *outliers* solamente entre aquellas instancias que han sido mal clasificadas en cada clase, utilizando una técnica basada en distancias y eliminando del conjunto de datos de entrenamiento las etiquetadas como anómalas; y finalmente, en el tercer paso, se construye el nuevo clasificador.

Cada una de estas fases está descrita a continuación. La Figura 5.11 muestra el proceso de forma gráfica para facilitar su comprensión.

- **Primer paso: construcción del modelo de clasificación inicial**

  En el primero paso, DARIM carga el conjunto de datos y aplica el clasificador seleccionado por el usuario, siendo la salida de este paso un modelo de clasificación con las instancias mal clasificadas etiquetadas como tales. (ver Figure 5.11, etiqueta "First Classification Model")

- **Segundo paso: detección de *outliers***

  En la segunda fase, DARIM separa las instancias mal clasificadas de cada clase en un nuevo conjunto de datos, y las bien clasificadas en otro diferente. Suponiendo que el conjunto de datos tenga dos clases (si bien DARIM puede ejecutarse sin problemas en conjuntos multi-clase), se generarían 3 conjuntos de datos: uno para las instancias bien clasificadas, otro para las instancias mal clasificadas de la clase positiva, y otro para las instancias mal clasificadas de la clase negativa.

  DARIM entonces aplica un algoritmo de *clustering* basado en particiones, k-Means, sobre las instancias mal clasificadas de cada clase, por separado, con objeto de obtener los clusters en los que se agrupan estas instancias.

  Una vez obtenidos los clusters y calculados sus centroides, se calcula la distancia euclídea de estos con respecto a la media del conjunto de instancias bien clasificadas que tienen el mismo valor de clase, marcando como anómalas a las instancias de los clusters cuya distancia sea mayor que la del resto, dado que serán estas instancias las que muestren un mayor comportamiento irregular con respecto al resto de instancias de la misma clase.

  Se muestra la Figura 5.10 para ayudar a comprender este proceso. Las instancias representadas pertenecen a la clase positiva de un conjunto de datos genérico y, en este caso, se han generado dos clusters, C1 y C2, sobre las instancias mal clasificadas. Las instancias del cluster cuyo centro este más alejado de la media de las instancias bien clasificadas serán marcadas como anómalas.

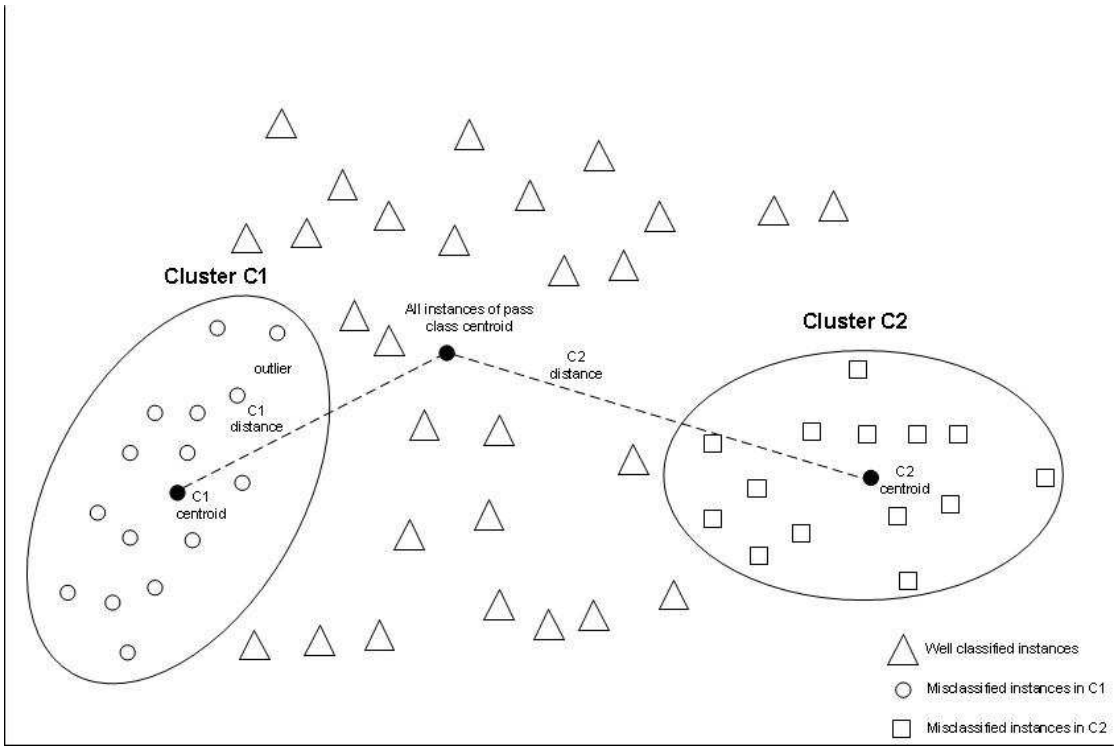


FIGURA 5.10: Clusters C1 y C2 agrupando instancias de la clase positiva que han sido mal clasificadas

Una vez marcadas las instancias anómalas, todos los conjuntos de datos vuelven a unirse en un único conjunto.

- **Tercer paso: clasificación final**

  Finalmente, DARIM elimina del conjunto de datos de entrenamiento a las instancias marcadas como anómalas, y construye de nuevo el modelo de clasificación.

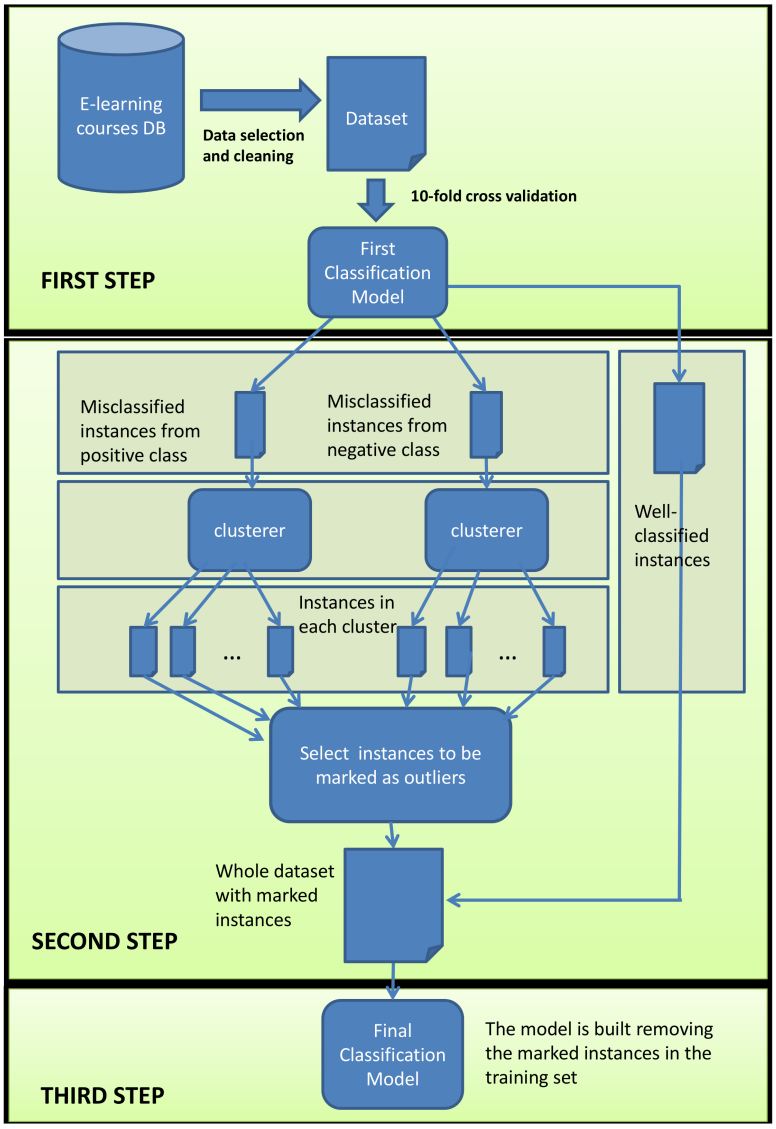


FIGURA 5.11: Proceso seguido por DARIM

En la Tabla 5.13 se muestran los resultados obtenidos al ejecutar DARIM con diferentes configuraciones. La configuración inicial de estas ejecuciones varió en dos parámetros: el número de clusters generados ("Clus. Gen.") para cada conjunto de instancias mal clasificadas, y el número de clusters sobre los que se considera que contienen las instancias de los estudiantes con comportamientos anómalos ("Clus. Elim.") que han de ser eliminados de los conjuntos de entrenamiento. A este respecto, las distancias euclídeas de los centroides de cada cluster con respecto a la media del resto de las instancias de la clase son ordenadas de mayor a menor, considerando siempre, y al menos, a las instancias del cluster con mayor distancia como *outliers*. En todos los casos, la ejecución de DARIM supone una mejora en todas y cada una de las medidas de calidad utilizadas

en el estudio, un hecho que contrasta con los procesos utilizados en los estudios 1 y 2, en los cuáles siempre existían medidas en las que se empeoraba.

En la Figura 5.12 se muestra la evolución de las mejoras obtenidas en términos de *accuracy*, *TPrate*, *TNrate*, *f-Measure* y *AUC* para los casos en los que se detectan y eliminan las instancias de todos los clusters excepto uno (2, 3, 4 y 5 "Clus. Gen", y 1, 2, 3 y 4 "Clus. Elim." respectivamente.). A medida que aumenta el número de clusters generados, también lo hacen las mejoras en *accuracy* y *TNrate*, llegando a obtener el mejor resultado con 5 clusters. Ocurre al contrario con el *TPrate*, siendo el mejor resultado el obtenido con 2 clusters. No obstante, excepto con el *TNrate*, en el que la diferencia de mejora entre la peor y la mejor ejecución es de algo más del 1.1 %, en los demás casos la diferencia de resultados no resulta ser notablemente significativa. Sí lo es en los casos en los que se generan 3, 4 y 5 clusters y sólo se eliminan las instancias de uno de ellos, debido al hecho de que en estas configuraciones se detectan menos comportamientos anómalos.

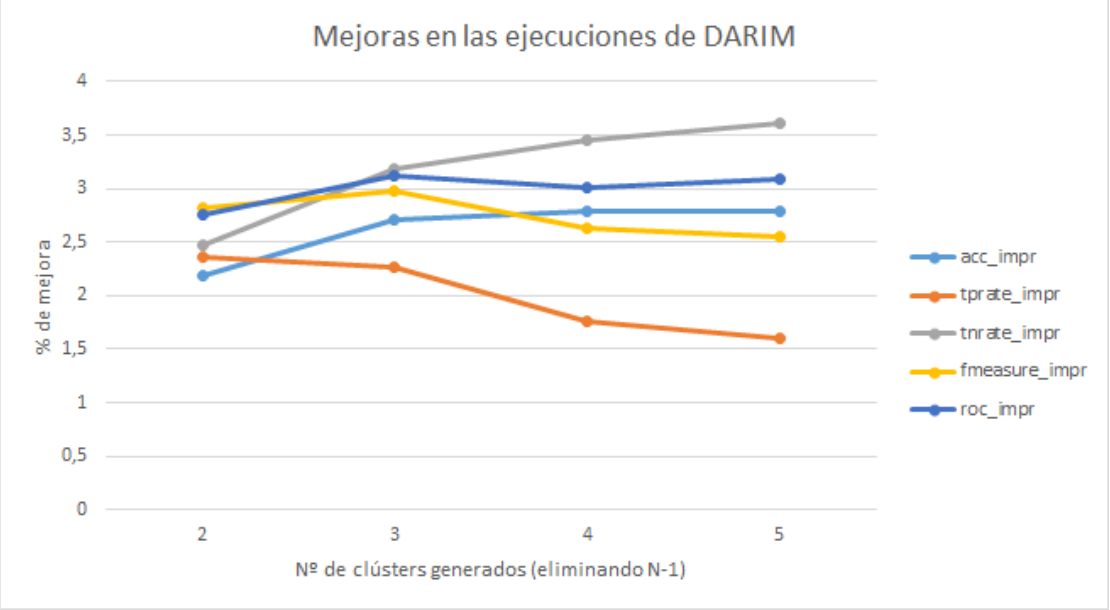


FIGURA 5.12: Mejoras obtenidas por DARIM utilizando de 2 a 5 clústers (y eliminando N-1) en el estudio 4

En las Figuras 5.13, 5.14 y 5.15 se comparan respectivamente las mejoras obtenidas por DARIM y por el proceso de Voto Mayoritario en media, las mejoras de la desviación estándar del proceso CV, y el tiempo medio de ejecución. En términos generales, la diferencia más notable la encontramos en el *TNrate*, superando el proceso de Voto Mayoritario ampliamente a DARIM. En términos de *accuracy*, el Voto Mayoritario tiene

Tabla 5.13: Mejoras obtenidas en los modelos de predicción al aplicar DARIM, por ejecución, en el estudio 4

		Mejora media por conjunto:					**Mejora media de la desviación estándar:**					
Clus. Gen.	Clus. Elim.	Acc.	TPrate	TNrate	f-Measure	AUC	Acc.	TPrate	TNrate	f-Measure	AUC	Tiempo
2	1	2,183	2,357	2,473	2,818	2,752	0,535	0,838	1,383	0,391	0,230	0,634
3	1	1,589	1,865	1,835	2,224	2,296	0,299	0,388	0,718	0,202	0,084	0,852
3	2	2,713	2,275	3,191	2,977	3,118	0,835	1,397	2,363	0,799	0,393	0,668
4	1	1,437	1,592	1,583	2,018	1,921	0,316	0,379	0,767	0,163	0,014	0,872
4	2	2,392	2,103	2,764	2,805	2,743	0,646	0,844	1,988	0,423	0,175	0,871
4	3	2,781	1,759	3,460	2,635	3,010	0,868	1,542	2,686	0,959	0,594	1,031
5	1	1,342	1,482	1,362	1,864	1,706	0,278	0,310	0,687	0,122	0,034	1,118
5	2	2,313	1,957	2,895	2,708	2,707	0,576	0,843	1,727	0,461	0,103	0,820
5	3	2,502	1,860	3,012	2,565	2,936	0,760	1,408	2,530	0,944	0,581	1,097
5	4	2,791	1,600	3,613	2,548	3,097	0,839	1,683	3,192	1,020	0,862	0,931
Estadísticas:												
Media		2,204	1,885	2,619	2,516	2,629	0,595	0,963	1,804	0,548	0,307	0,889
Mediana		2,353	1,863	2,829	2,600	2,747	0,611	0,843	1,857	0,442	0,202	0,872
Desv. Est.		0,528	0,278	0,746	0,346	0,469	0,222	0,488	0,852	0,333	0,274	0,154
Máx.		2,791	2,357	3,613	2,977	3,118	0,868	1,683	3,192	1,020	0,862	1,118
Min.		1,342	1,482	1,362	1,864	1,706	0,278	0,310	0,687	0,122	0,014	0,634

también un rendimiento notablemente superior. No obstante, DARIM alcanza mejores rendimientos en el *TPrate*, esto es, en la clasificación de suspensos. Las mismas conclusiones pueden extraerse al comparar las mejoras de la desviación estándar del proceso CV. En cuanto al tiempo de ejecución, destaca el hecho de que, en el caso de DARIM, sea prácticamente inapreciable, estando por debajo de la escala de los milisegundos en media, mientras que el proceso de Voto Mayoritario tiene una media superior a los 3 segundos. Esto es algo lógico, ya que mientras DARIM sólo requiere de la ejecución de un algoritmo de *clustering* sobre un conjunto pequeño de instancias (las mal clasificadas de cada clase) y un cálculo de distancias, el proceso de Voto Mayoritario ejecuta 9 algoritmos de clasificación, muchos de los cuales son bastante "pesados".

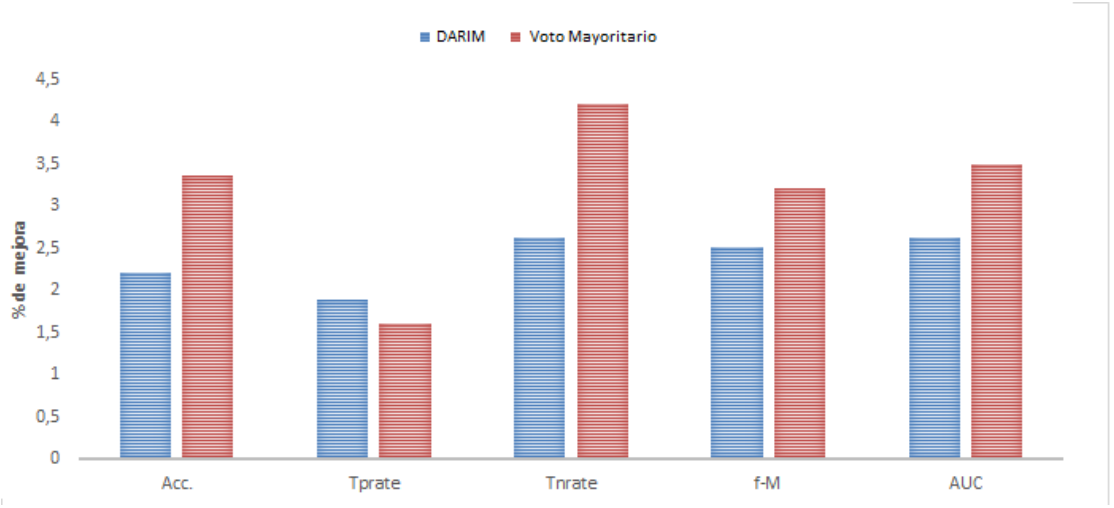


FIGURA 5.13: Comparativa de mejora media entre DARIM y el proceso Voto Mayoritario en el estudio 4

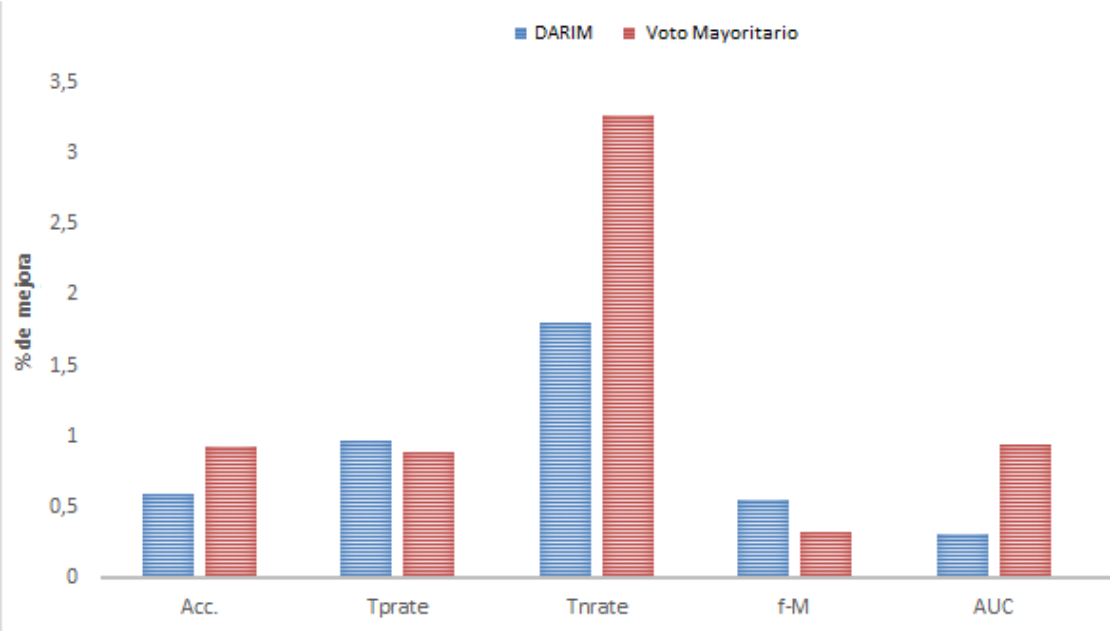


FIGURA 5.14: Comparativa de mejora en la desviación estándar del proceso CV entre DARIM y el proceso Voto Mayoritario en el estudio 4

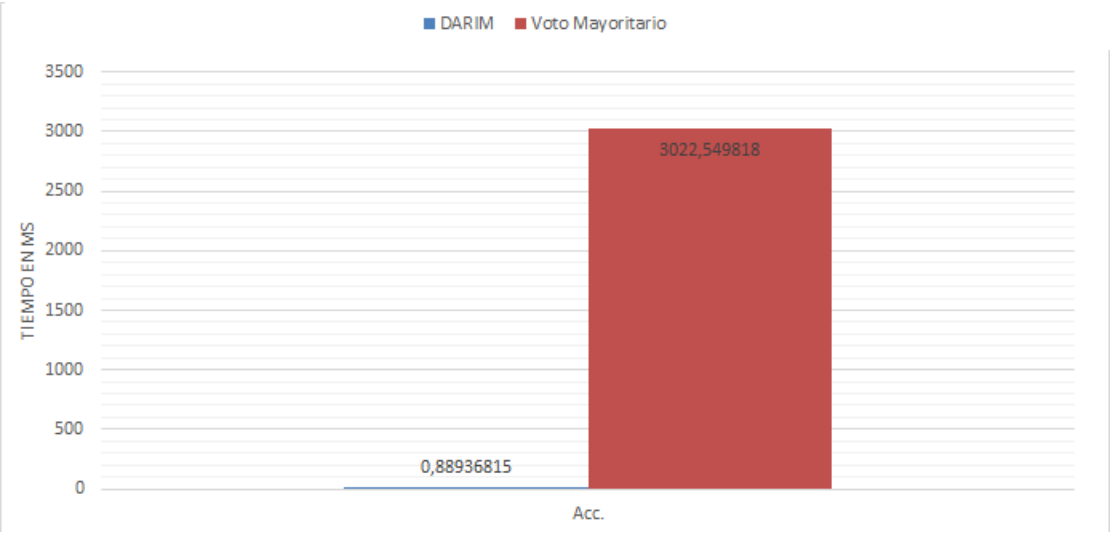


FIGURA 5.15: Comparativa de tiempos de ejecución entre DARIM y el proceso Voto Mayoritario en el estudio 4

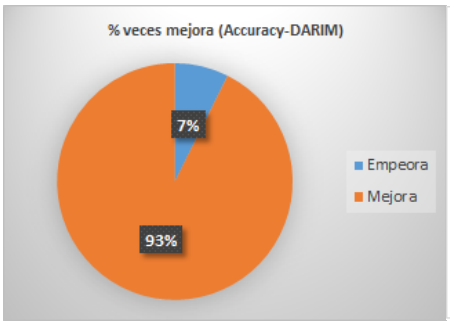
En la Tabla 5.14 se exponen los resultados de aplicar DARIM con 2 clusters por cada conjunto de instancias mal clasificadas, proceso que obtiene el mejor resultado medio en *TPrate*, esto es, en clasificación de suspensos. Como ayuda a la visualización de los resultados, se incluye la Figura 5.16, conteniendo en términos porcentuales en cuántas ocasiones DARIM mejora en cada una de las cinco medidas y en cuántas ocasiones la mejora se sitúa entre el 0 % y el 1 %, el 1 % y el 2 %, o es mayor del 2 %.

En términos de *accuracy*, DARIM consigue mejorar en 25 de los 27 conjuntos de datos utilizados, lo que supone el 93 % de los mismos. Más aún, en los dos únicos conjuntos en los que empeora, dataset2 y dataset22, el porcentaje de reducción de *accuracy* es muy bajo, de solamente 0,019 y 0,1 respectivamente. De entre los 25 casos en los que consigue mejoría, en 10 (40 %) alcanza mejoras de entre el 1 y el 2 %, y en otros 10 la mejora es superior al 2 %. También obtiene buenos resultados en el resto de medidas, destacando además que la *f-Measure* sólo empeora en 1 caso.

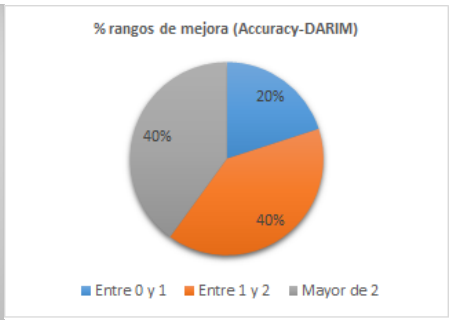
Tabla 5.14: Mejoras obtenidas en los modelos de predicción al aplicar DARIM, por conjunto de datos, en el estudio 4

	Mejora media por conjunto:					**Mejora media de la desviación estándar:**					
Conjunto	Acc.	TPr.	TNr.	f-M	AUC	Acc.	TPr.	TNr.	f-M	AUC	Tiempo
dataset1	1,927	-0,880	6,336	0,829	4,527	1,618	2,635	-1,083	2,259	0,899	0,516
dataset2	-0,019	2,338	-0,395	2,049	-0,375	-0,066	-0,670	0,008	-0,240	-0,329	1,548
dataset3	2,573	3,522	1,486	2,732	2,009	-0,212	0,332	2,686	0,049	0,003	0,484
dataset4	0,403	0,533	-0,307	0,243	-0,965	0,423	1,093	1,349	0,319	2,148	0,000
dataset5	1,631	11,089	-1,757	7,395	2,930	-0,390	0,065	-0,635	2,067	-0,318	0,968
dataset6	1,665	2,978	1,317	4,259	1,646	0,062	-0,666	0,245	-1,038	0,617	0,516
dataset7	0,248	-1,328	1,466	0,093	0,975	-0,136	0,824	1,444	0,732	-1,184	0,000
dataset8	5,772	7,796	4,839	9,147	5,201	3,012	7,362	5,388	7,303	1,212	0,000
dataset9	1,817	1,008	2,053	2,803	3,059	0,070	0,733	1,063	-0,697	1,733	1,516
dataset10	1,075	-1,613	1,446	1,682	0,431	0,480	1,347	1,600	2,641	2,225	0,484
dataset11	4,057	6,452	2,180	5,588	4,296	-0,194	0,391	0,857	-0,452	-0,134	0,484
dataset12	3,959	1,843	7,661	2,998	6,264	1,964	-2,353	1,553	-3,285	0,285	0,000
dataset13	1,747	2,581	-1,229	1,390	2,805	2,024	2,648	1,779	1,824	1,510	0,516
dataset14	2,621	1,553	4,839	1,727	1,792	0,271	2,258	5,606	0,764	1,011	0,000
dataset15	1,256	0,899	3,076	0,761	-1,292	0,303	1,049	-1,404	0,246	-0,285	1,484
dataset16	2,026	5,242	1,290	4,910	0,847	1,117	1,226	-0,299	1,063	-0,209	1,000
dataset17	4,467	1,290	6,452	2,796	6,331	2,136	1,963	2,171	0,716	-4,118	0,000
dataset18	9,867	12,500	7,527	12,361	12,724	0,780	-2,124	2,708	-3,395	-2,519	0,484
dataset19	2,535	1,466	3,226	2,966	6,581	0,984	0,744	5,140	0,252	1,437	0,516
dataset20	0,860	-0,461	1,262	1,078	4,618	-0,017	-0,025	0,240	-0,189	0,290	0,516
dataset21	1,780	1,434	1,935	2,720	2,764	0,383	-0,212	2,004	-1,057	0,436	1,000
dataset22	-0,100	-0,749	0,796	-0,174	-0,055	-0,112	0,086	-0,381	-0,227	0,298	1,548
dataset23	1,865	0,978	2,810	1,805	2,994	0,870	3,183	3,900	2,215	0,864	0,516
dataset24	0,943	0,587	1,310	0,912	0,979	-0,238	0,176	0,697	-0,364	0,184	0,516
dataset25	2,016	0,982	4,659	1,363	2,791	-0,875	-0,437	0,678	-1,363	0,305	0,484
dataset26	1,024	0,419	1,924	0,698	-0,055	0,396	0,405	1,248	0,256	0,138	1,000
dataset27	0,917	1,184	0,558	0,943	0,478	-0,202	0,583	-1,220	0,161	-0,293	1,032
Estadísticas:											
Media	2,183	2,357	2,473	2,818	2,752	0,535	0,838	1,383	0,391	0,230	0,634
Mediana	1,780	1,290	1,924	1,805	2,764	0,303	0,583	1,248	0,246	0,290	0,516
Desv. Est.	2,065	3,486	2,509	2,911	2,996	0,921	1,836	1,917	2,002	1,325	0,499
Máx.	9,867	12,500	7,661	12,361	12,724	3,012	7,362	5,606	7,303	2,225	1,548
Min.	-0,100	-1,613	-1,757	-0,174	-1,292	-0,875	-2,353	-1,404	-3,395	-4,118	0,000
N# de veces que empeora, obtiene el mismo resultado, o mejora:											
Empeora	2	5	4	1	5	10	7	6	11	9	-
Mejora	25	22	23	26	22	17	20	21	16	18	-
N# de veces que la mejora es mayor que 0, 1 o 2 %:											
Entre 0 y 1	5	6	2	7	5	11	10	6	9	11	-
Entre 1 y 2	10	7	9	6	2	3	5	7	2	5	-
Mayor de 2	10	9	12	13	15	3	5	8	5	2	-

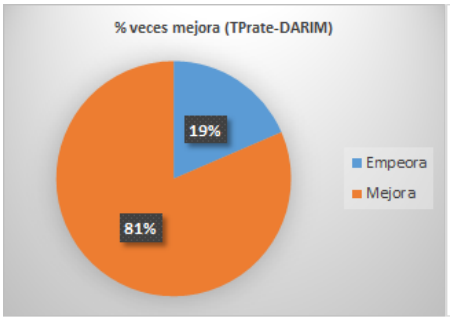
FIGURA 5.16: Porcentajes de mejora por medida usando DARIM en el estudio 4



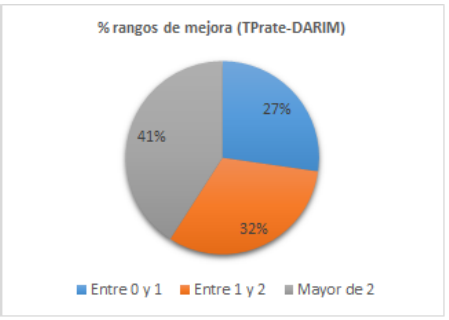
(A) % de mejoras en Acc.



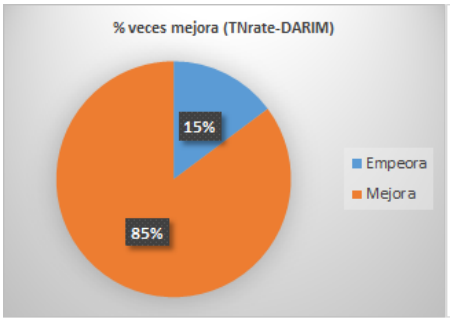
(B) % de rangos de mejora en Acc.



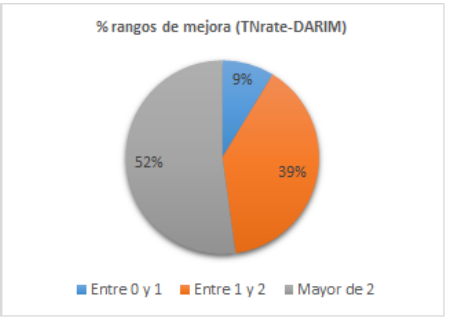
(C) % de mejoras en TPrate



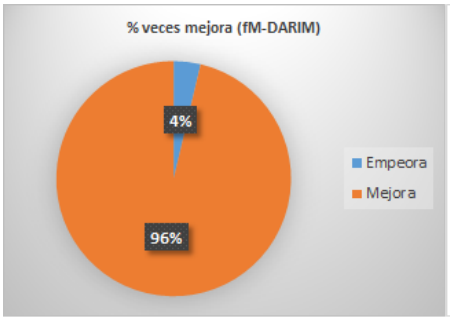
(D) % de rangos de mejora en TPrate



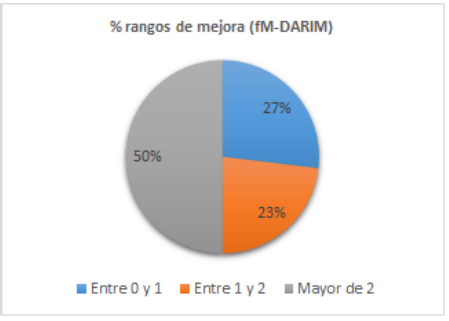
(E) % de mejoras en TNrate



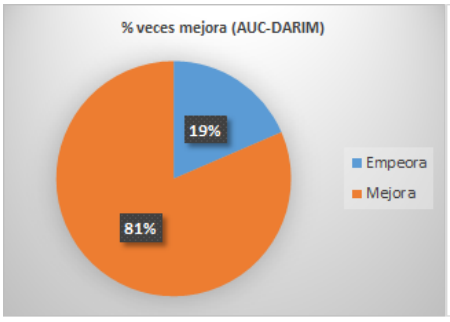
(F) % de rangos de mejora en TNrate



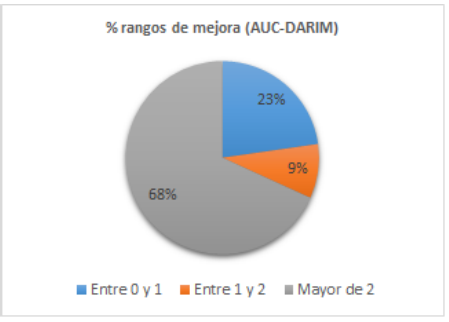
(G) % de mejoras en f-M



(H) % de rangos de mejora en f-M



(I) % de mejoras en AUC

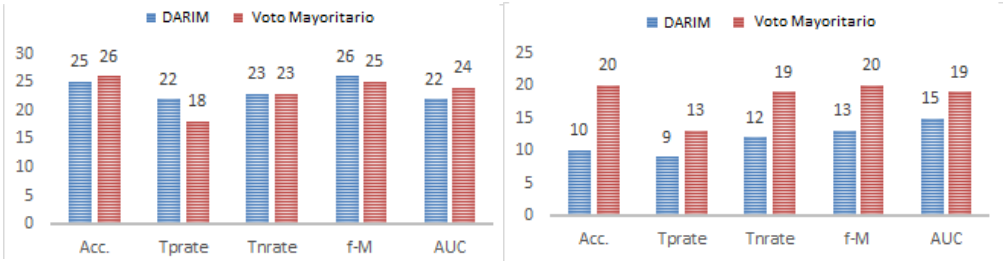


(J) % de rangos de mejora en AUC

En la siguiente Figura 5.17 se incluyen dos gráficas con la comparativa entre las mejoras de DARIM obtenidas por conjunto y las alcanzadas con el proceso de Voto Mayoritario cuyos resultados fueron expuestos en el anterior estudio 3. La Gráfica 5.17a, que compara el número de conjuntos en que cada uno de los procesos obtiene mejoras en cada medida, muestra que no existen, en general, grandes diferencias entre los resultados obtenidos por ambos. El proceso de Voto Mayoritario consigue mejorar en *accuracy* en un caso más, y en dos casos más con respecto al *AUC*. Sin embargo, con respecto a la *f-Measure*, DARIM sale mejor parado de la comparativa, al superar al proceso de Voto Mayoritario. Más aún, la diferencia más notable se produce en el *TPrate*, esto es, en la predicción de estudiantes suspensos, en la que DARIM le supera ampliamente. Como ya se ha mencionado anteriormente, esto es especialmente significativo ya que una de las principales preocupaciones de cara a mejorar los modelos de clasificación es la mejora de la predicción de los estudiantes suspensos, al ser estos los que de forma prioritaria han de ser identificados por los profesores.

Pese a este mejor rendimiento de DARIM en cuanto al número de casos en los que se obtiene mejoría, el proceso de Voto Mayoritario, cuando mejora los modelos de predicción, lo suele hacer en unos porcentajes más altos que DARIM, tal y como se desprende de la Gráfica 5.17b. Para todas las medidas, el proceso de Voto Mayoritario obtiene en un mayor número de ocasiones una mejora de más del 2 %.

FIGURA 5.17: Comparativa de mejoras entre DARIM y el proceso Voto Mayoritario por conjunto en el estudio 4



(A) Veces que mejoran (B) Veces que la mejora es mayor al 2 %

No obstante a lo comentado, si observamos la gráficas 5.18 y 5.19, que comparan los máximos y mínimos alcanzados por ambos procesos en la agrupación por conjuntos, puede claramente observarse que, mientras que los máximos alcanzados por uno y por otro tienen valores similares, el proceso de Voto Mayoritario alcanza unos mínimos mucho

menores en todas las medidas que los alcanzados por DARIM, que apenas son de un -0.1 % en *accuracy* o de -1.613 en *TPrate*. De estas gráficas se desprenden dos conclusiones: la primera de ellas, que pese al buen rendimiento del proceso de Voto Mayoritario, cuando este empeora los modelos de predicción lo hace de forma muy notable, mientras que las pocas ocasiones en las que DARIM empeora lo hace de forma mucho menos perceptible. En segundo lugar, se denota que DARIM es un proceso bastante más estable que el de Voto Mayoritario, al no existir unas diferencias tan grandes como en este último entre las mejoras máximas y mínimas alcanzadas. Este hecho se debe, sobretodo, a que DARIM realiza la detección y eliminación de instancias con comportamientos anómalos en ambas clases, algo que el proceso de Voto Mayoritario no puede garantizar.

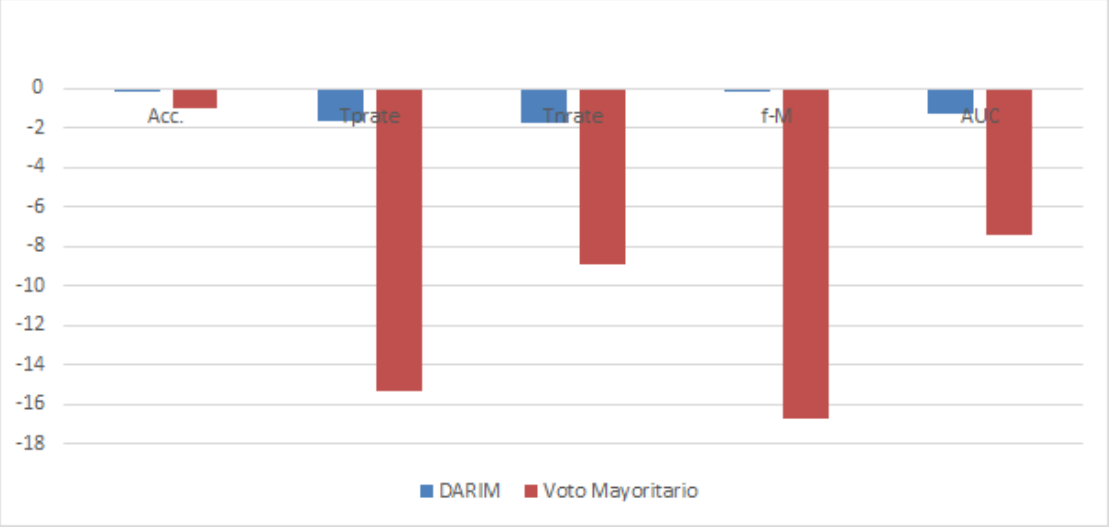


FIGURA 5.18: Comparativa de mejoras mínimas alcanzadas por DARIM y por el proceso de Voto Mayoritario, por conjunto, en el estudio 4

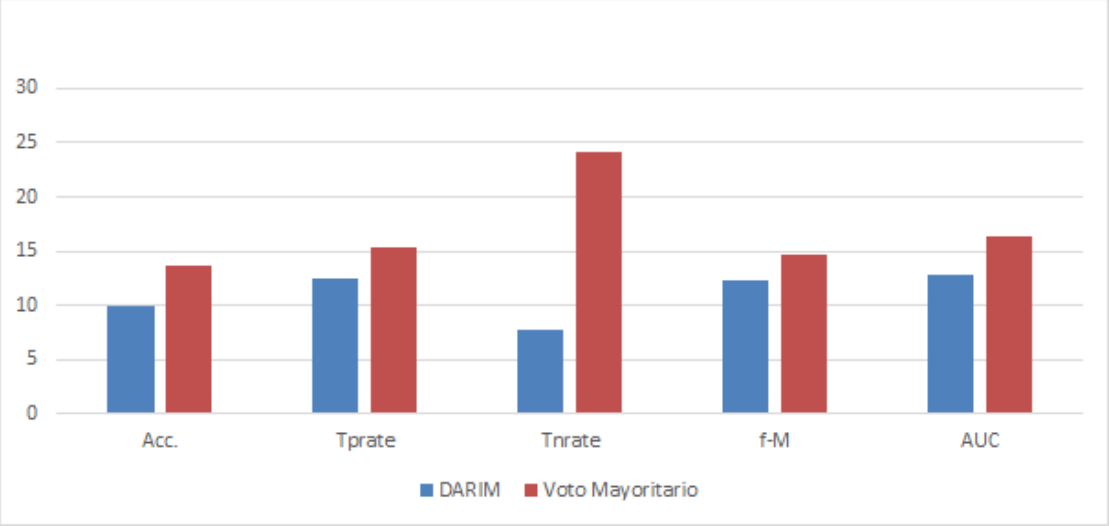


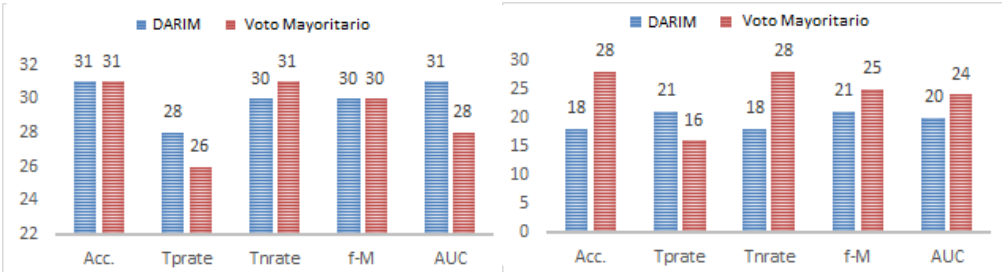
FIGURA 5.19: Comparativa de mejoras máximas alcanzadas por DARIM y por el proceso de Voto Mayoritario, por conjunto, en el estudio 4

En la Tabla 5.15 se muestran los resultados de DARIM agrupados por clasificador. Las conclusiones son similares a las obtenidas en el agrupamiento por conjuntos de datos. Sin embargo, puede constatarse que en términos de número de veces en los que se obtiene mejoría, hay un 100 % de mejora en *accuracy* y *AUC*, esto es, se mejoran en media los modelos de los 31 clasificadores. Respecto al resto de medidas, el número de veces que se mejora es también muy alto, de 28, 30 y 30 en *TPrate*, *TNrate* y *f-Measure* respectivamente.

Tabla 5.15: Mejoras obtenidas en los modelos de predicción al aplicar DARIM, por clasificador, en el estudio 4

Clasificador	Acc.	TPr.	TNr.	f-M	AUC
BayesNet	3,241	3,966	4,188	4,202	5,052
NaiveBayes	1,208	-3,926	5,058	0,460	0,456
Logistic	2,355	4,302	1,747	3,716	1,958
MLP	1,621	4,852	1,819	4,145	3,855
RBFNetwork	0,908	-2,972	3,194	-1,071	0,163
SimpleLogistic	0,775	0,269	0,763	0,694	1,772
SMO	0,992	2,465	-0,312	1,758	1,077
SPegasos	2,552	3,383	0,587	3,464	1,985
IB1	2,944	2,676	3,987	3,848	3,331
IBk	1,716	2,263	1,153	1,141	1,422
KStar	3,256	2,101	4,545	3,545	3,765
LWL	2,251	5,643	0,994	5,014	1,575
ConjunctiveRule	2,185	0,858	2,760	1,446	2,134
DecisionTable	1,411	1,946	2,269	2,254	3,224
DTNB	1,887	-0,748	5,894	1,565	3,178
JRip	2,549	2,636	1,678	2,803	2,739
NNge	1,658	3,196	0,313	2,508	1,755
OneR	2,357	4,397	0,688	3,969	2,543
PART	2,138	1,609	2,138	3,107	1,519
Ridor	1,163	1,044	2,047	0,647	1,545
ADTree	3,246	4,577	2,063	3,799	3,636
BFTree	2,743	4,512	2,529	4,490	3,549
DecisionStump	2,648	4,676	0,934	4,945	3,383
FT	1,944	1,224	3,470	1,866	2,659
J48	2,557	2,245	3,845	3,244	3,127
LADTree	3,689	2,397	6,468	3,298	6,554
LMT	0,959	1,351	0,774	1,480	2,623
NBTree	3,109	3,488	3,663	4,441	3,185
RandomTree	3,268	2,471	3,532	3,661	3,213
REPTree	1,630	2,248	0,841	2,145	3,000
SimpleCart	2,697	3,922	3,023	4,757	5,330
N# de veces que empeora, obtiene el mismo resultado, o mejora:					
Empeora	0	3	1	1	0
Mejora	31	28	30	30	31
N# de veces que la mejora es mayor que 0, 1 o 2 %:					
Entre 0 y 1	4	2	8	3	2
Entre 1 y 2	9	5	4	6	9
Mayor de 2	18	21	18	21	20

En la siguientes Gráficas 5.20a y 5.20b se comparan las mejoras agrupadas por clasificador obtenidas con DARIM y con el proceso de Voto Mayoritario. Al igual que ya ocurría en el agrupamiento por conjuntos de datos, ambos procesos tienen unos resultados similares si se comparan en base al número de veces que mejoran, siendo DARIM el que mejor rendimiento obtiene en la predicción de estudiantes suspensos, *TPrate*. No obstante, y pese a que en la mayor parte de las medidas el proceso de Voto Mayoritario obtiene en un número más elevado de veces una mejora superior al 2 %, en el agrupamiento por clasificadores DARIM le supera en este aspecto al comparar el *TPrate*, lo que refuerza aún más la conclusión de que DARIM muestra ser un proceso bastante más eficiente a la hora de detectar comportamientos anómalos entre los estudiantes suspensos y mejorar así su predicción.



(A) Veces que mejoran (B) Veces que la mejora es mayor al 2 %

FIGURA 5.20: Comparativa de mejora por clasificador entre DARIM y Voto Mayoritario en el estudio 4

En la siguiente Tabla 5.16, y como apoyo a la comparativa en la Figura 5.21, están incluidos los resultados de mejora obtenidos por DARIM según los rangos iniciales de *accuracy* de los modelos de predicción, esto es según el *accuracy* de los modelos antes de aplicar DARIM. Claramente, puede observase una tendencia por la cual, a medida que el *accuracy* inicial de los modelos es más alto, la mejora obtenida por DARIM disminuye en casi todos los casos, a excepción de un caso notable de crecimiento de mejora con las medidas de *TPrate* y *f-Measure* entre los rangos "60 a 70" y "70 a 80". De hecho, este crecimiento de mejora es en el *TPrate* superior al 1.2 %.

TABLA 5.16: Mejoras obtenidas en los modelos de predicción al aplicar DARIM, por accuracy inicial, en el estudio 4

Rango Acc.	Acc.	TPrate	TNrate	f-Measure	AUC	% de conjuntos
menor de 50	19,128	21,286	16,468	25,187	18,894	1 %
50 a 60	11,063	13,497	6,161	10,640	8,968	3 %
60 a 70	3,327	2,236	5,160	2,805	4,229	16 %
70 a 80	2,478	3,492	2,391	3,679	2,701	28 %
80 a 90	1,239	1,450	1,631	2,279	1,972	38 %
mayor de 90	-0,011	-0,807	0,248	-0,458	1,035	14 %

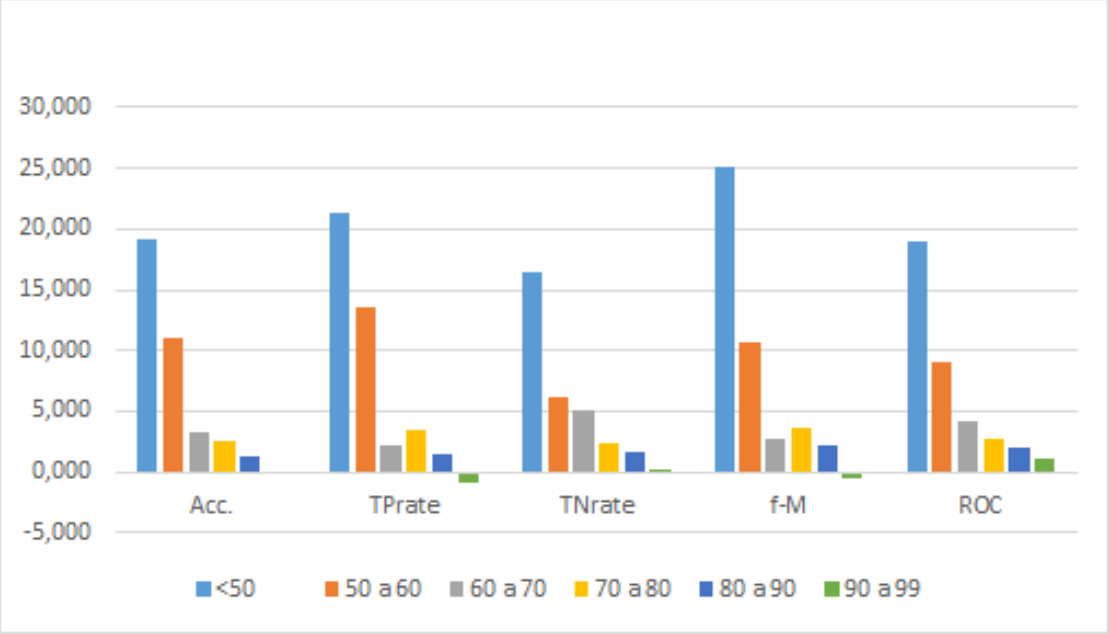
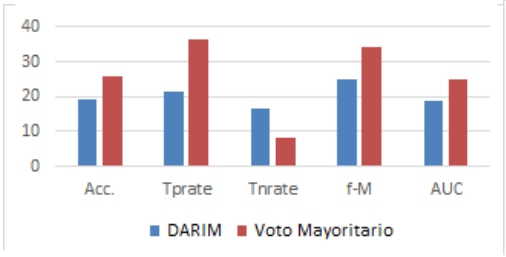


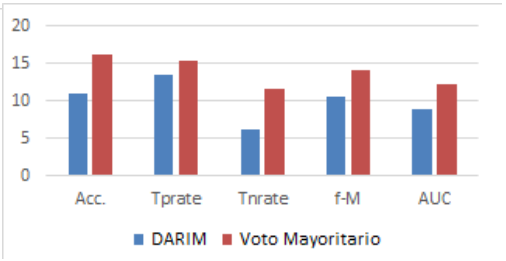
FIGURA 5.21: Mejoras por rango de accuracy inicial (DARIM) del estudio 4

En la Figura 5.22 se muestra una comparativa entre DARIM y el proceso de Voto Mayoritario por rangos de *accuracy* inicial. En términos generales, el proceso de Voto Mayoritario obtiene porcentajes de mejora más altos en la mayor parte de las ocasiones. En ambos casos, se empeora o no se consiguen mejoras notables cuando los modelos tienen un *accuracy* inicial superior al 90 %. Sin embargo, para los rangos de "70 a 80" y de "80 a 90", que engloban al 66 % de las ejecuciones, las mejoras obtenidas con DARIM en términos de *TPrate* son notablemente mejores que con Voto Mayoritario, una tendencia que ya se ha puesto de manifiesto en las comparativas anteriores. De hecho, el proceso de Voto Mayoritario empeora el *TPrate* de los modelos en el intervalo "80 a 90", mientras que DARIM alcanza una mejora por encima del 1 %.

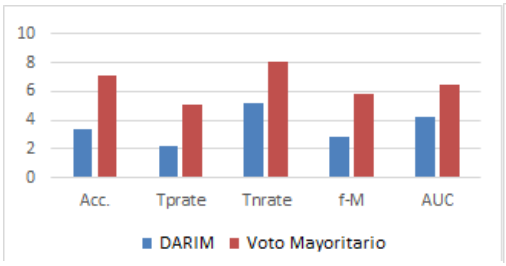
FIGURA 5.22: Comparativa de mejoras entre DARIM y el proceso de Voto Mayoritario, según rango de accuracy inicial, en el estudio 4



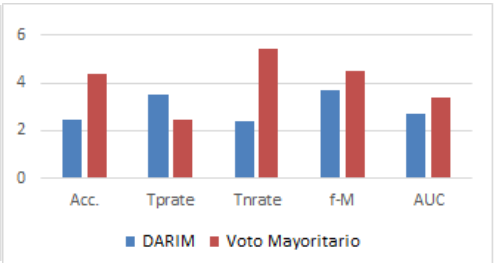
(A) Acc. inicial menor de 50 %



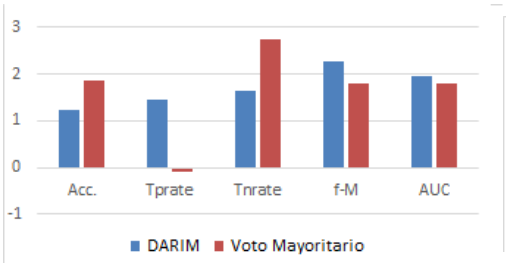
(B) Acc. inicial entre 50 % y 60 %



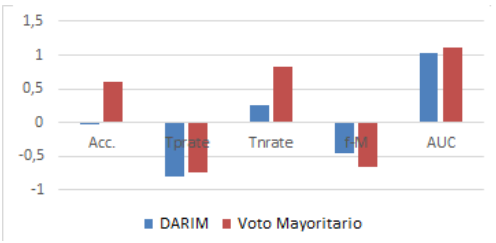
(C) Acc. inicial entre 60 % y 70 %



(D) Acc. inicial entre 70 % y 80 %



(E) Acc. inicial entre 80 % y 90 %



(F) Acc. inicial mayor de 90 %

Tanto DARIM como el proceso de Voto Mayoritario obtienen unos resultados notablemente superiores que los obtenidos al utilizar algoritmos bien conocidos de eliminación de *class-outliers* como es ECODB, y cuyos resultados han sido expuestos en el estudio 1. El motivo principal de este mejor rendimiento lo podemos encontrar al observar las Gráficas de las Figuras 5.23 y 5.24, en las que se muestran la distribución de los estudiantes en uno de los cursos y, respectivamente, los comportamientos anómalos detectados por ECODB (con k=7 y porcentaje de instancias a eliminar = 10) y DARIM (con 2 clusters). En todos los casos, los estudiantes suspensos son representados con triángulos azules, y los aprobados con cuadrados rojos. Los estudiantes clasificados como anómalos en su comportamiento son representados, en ambos casos, con relleno de color sólido, señalando además con círculos verdes numerados las zonas en las que se encuentran los estudiantes suspensos marcados como anómalos. En las Gráficas superiores de ambas

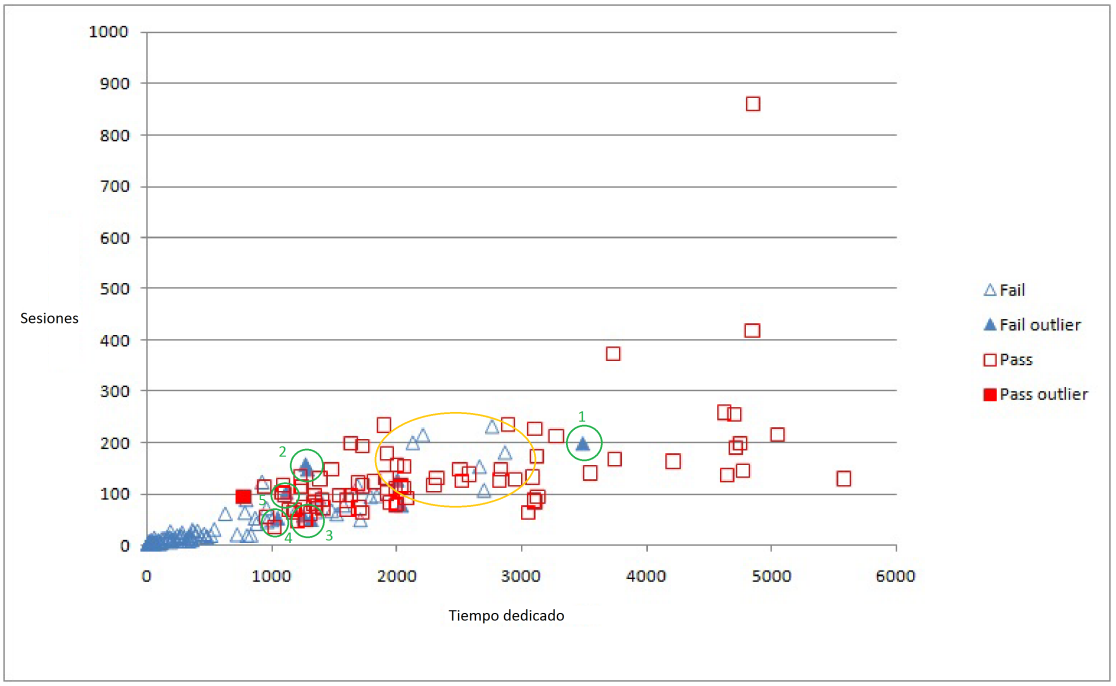
Figuras, 5.23a y 5.24a, están representados todos los estudiantes. Para facilitar la identificación y características de los detectados como anómalos, además, las Gráficas 5.23b y 5.24b de ambas Figuras contienen únicamente la representación de estos estudiantes.

Observando estas gráficas, puede comprobarse el motivo por el cuál ECODB no está siendo eficiente. Por un lado, sólo señala una instancia de la clase de los aprobados como anómala. Si bien puede verse que representa a uno de los estudiantes aprobados con menor actividad, y por tanto ECODB parece estar acertando al detectarla, existen muchas otras instancias de estudiantes aprobados con baja actividad que se le escapan. Por otro lado, de los nueve estudiantes suspensos detectados, solamente el del círculo 1 responde al perfil de estudiante que, pese a suspender, tiene una alta actividad en el curso y más cercana a la de sus compañeros aprobados que al resto de suspensos. Los restantes 8, si bien es cierto que también tienen una actividad más alta que la media de su clase, se encuentran por debajo de muchas otros estudiantes suspensos con mayor actividad, marcados con una elipse naranja en la Gráfica 5.23a, y que probablemente tienen un comportamiento más anómalo que los 8 detectados.

En cambio, DARIM consigue detectar claramente a los 8 estudiantes suspensos con un comportamiento "más anómalo"que el resto de los suspensos (se muestran rodeados en una elipse verde en la gráfica 5.24a). En cuanto a los aprobados, DARIM detecta a tres, un mayor número que ECODB. Estos dos hechos sumados explican los motivos por los cuales DARIM tiene un rendimiento superior: (1) al enfocarse en la detección de *outliers* sobre instancias previamente mal clasificadas, es más probable detectar aquellas con comportamientos anómalos, esto es, cuyos valores de atributos se diferencien notablemente del resto de las de su clase; y (2), al contrario que ECODB, DARIM en su proceso fuerza a detectar *outliers* en ambas clases, por lo que es menos probable que se le escapen los *outliers* de una de las clases al detectar muchos de la otra.

FIGURA 5.23: Comportamientos anómalos detectados por ECODB

(A) Distribución de los estudiantes, marcando comportamientos anómalos con ECODB



(B) Distribución de los estudiantes con comportamientos anómalos en la Figura 5.23a

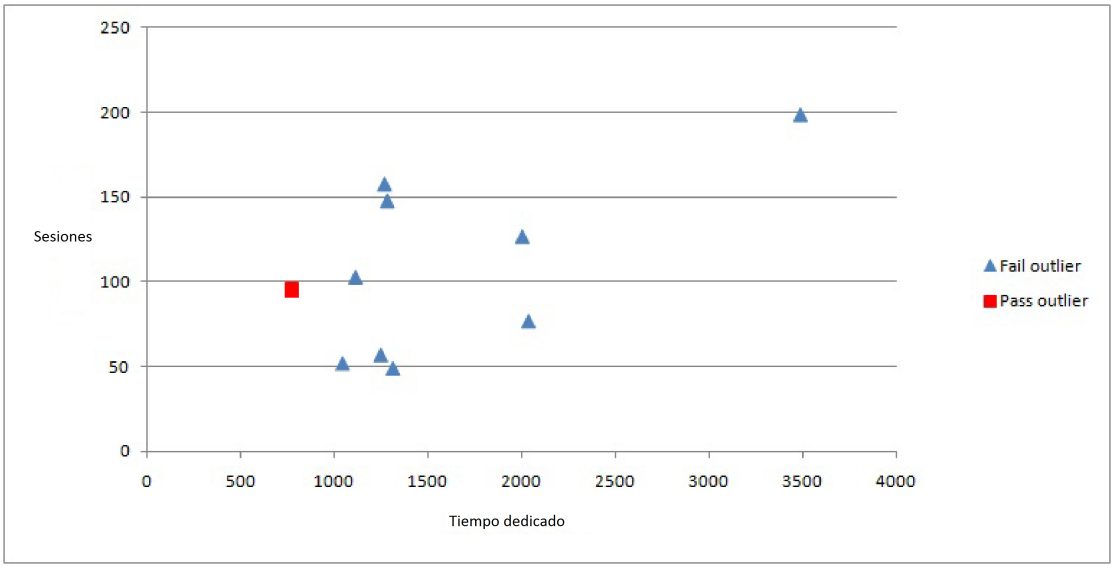
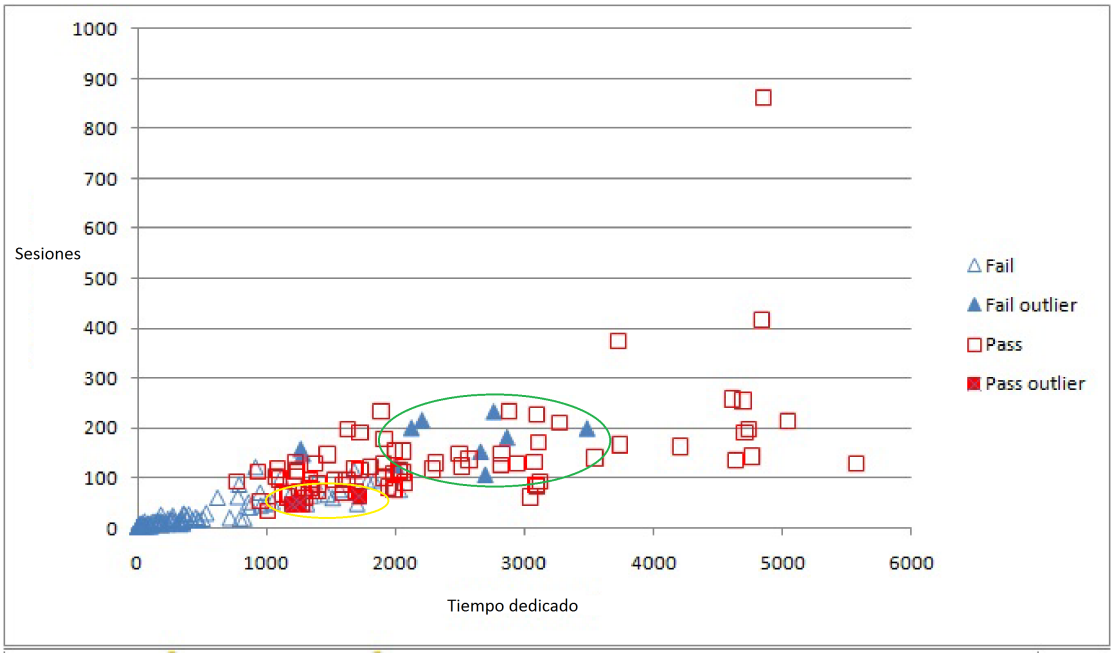
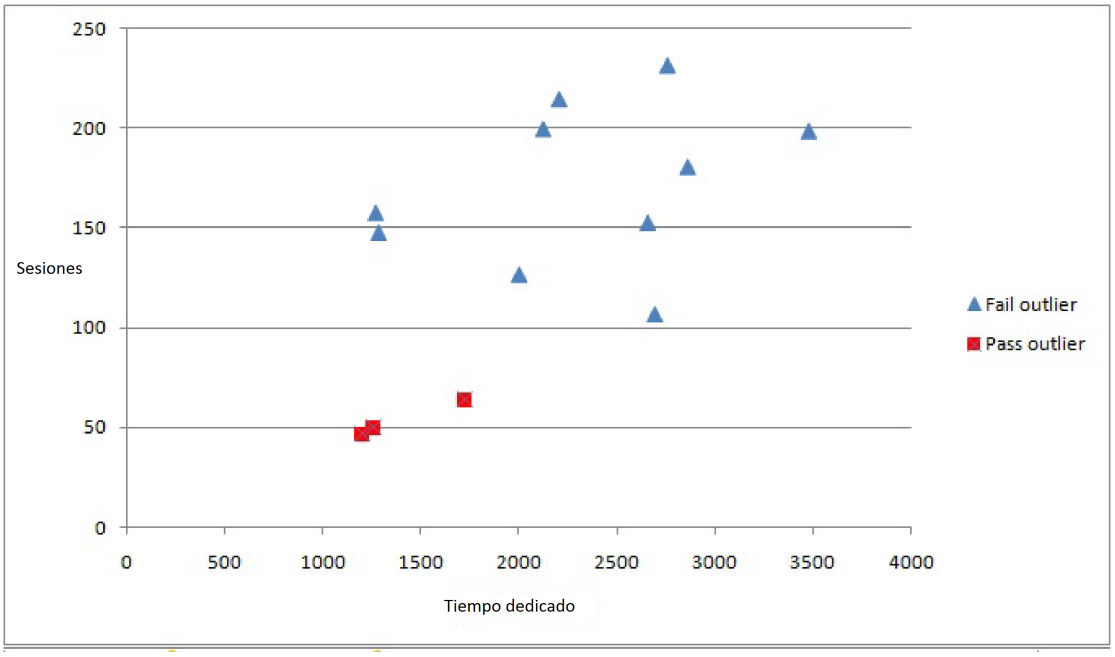


FIGURA 5.24: Comportamientos anómalos detectados por DARIM

(A) Distribución de los estudiantes, marcando comportamientos anómalos con DARIM



(B) Distribución de los estudiantes con comportamientos anómalos en la Figura 5.24a



5.4.5. Estudio 5. DARIM: otra aproximación basada en densidad

El proceso DARIM sigue requiriendo de dos parámetros para su funcionamiento, lo que a priori puede requerir de la intervención del usuario para configurarlo: (1) el número de clusters que han de ser generados para cada conjunto de instancias mal clasificadas

de cada clase, y (2) el número de estos clusters cuyas instancias han de ser consideradas *outliers*. El primero de estos problemas puede solventarse haciendo uso de diferentes técnicas existentes en la literatura que retornan valores o índices informando sobre la calidad de los modelos de *clustering*, como es por ejemplo el *Davies Bouldin's index* [257]. Diversos estudios realizados con DARIM utilizando estos índices han dado como resultados diferentes recomendaciones sobre el número de clusters a generar, según el índice que se use, si bien en una gran parte de los casos suele coincidir que, debido al bajo número de instancias a tratar, se suelan recomendar dos clusters. Esta recomendación solventaría así mismo el segundo de los problemas, ya que al generarse dos clusters, automáticamente las instancias de uno de ellos, el de mayor distancia a la media de su clase, son consideradas *outliers*. Sumado al hecho de que, tal como se ha mostrado en el estudio 4, esta configuración tiene buenos resultados, DARIM podría perfectamente ofrecerse a profesores de cursos virtuales como un proceso que, sin ser libre de parámetros, no necesita de intervención por parte del usuario para mejorar los modelos de predicción del rendimiento de los estudiantes al tener una configuración prefijada. No obstante, esta configuración sólo sería válida en el contexto de los cursos virtuales, en los que el número de instancias a tratar es reducido.

Aún con los buenos resultados de DARIM utilizando el algoritmo k-Means, existen otros algoritmos de *clustering* que podrían sustituirle o complementarle, adaptándose mejor a la distribución de las instancias que van a tratarse. En la Tabla 5.17 se muestran las mejoras obtenidas al sustituir el algoritmo k-Means por DBSCAN en DARIM, con valores del parámetro *Epsilon* que van del 0,5 al 1, teniendo el parámetro que indica el *Número Mínimo de Puntos* un valor de 1/10 del número de instancias a tratar. Al igual que con la configuración con k-Means, DARIM sigue obteniendo en todos los casos una mejoría de los modelos de predicción. No obstante, estas ejecuciones tienen el mismo problema que el proceso de Voto Mayoritario, y es que sacrifican la mejora en *TPrate* para mejorar aún más en *TNrate*.

En la Tabla 5.18 se muestran las resultados por conjunto de datos, y en las Figuras 5.25 y 5.26 se expone una comparativa con el DARIM del estudio 4. El *accuracy* es la única medida que no empeora en ninguno de los casos, si bien en uno de ellos la mejora es del 0 %. Además, mientras que DARIM con k-Means empeora en *accuracy* con los conjuntos "dataset2" y "dataset22", con DBSCAN consigue mejoras, que en el segundo caso son bastante notables. Sin embargo, el rendimiento en *TPrate* y *f-Measure*

Tabla 5.17: Mejoras obtenidas en los modelos de predicción al aplicar DARIM con DBSCAN, por ejecución (número mínimo de puntos = 1/10 del número de instancias), en el estudio 5

	Mejora media por conjunto:					**Mejora media de la desviación estándar:**					
Epsilon	Acc.	TPrate	TNrate	f-Measure	AUC	Acc.	TPrate	TNrate	f-Measure	AUC	Tiempo
0,5	2,176	1,020	3,431	2,008	2,597	0,869	1,606	3,140	0,609	1,430	0,612
0,6	1,912	0,611	3,264	1,709	2,353	0,866	1,541	3,206	0,575	1,373	0,714
0,7	1,738	0,709	2,829	1,550	2,232	0,852	1,511	3,221	0,546	1,351	0,711
0,8	1,682	0,877	2,676	1,571	2,172	0,798	1,401	2,907	0,574	1,283	0,725
0,9	1,630	0,862	2,658	1,576	2,049	0,641	1,265	2,591	0,607	1,040	0,578
1	1,463	0,815	2,511	1,455	1,789	0,504	1,096	2,211	0,497	0,875	0,933
Estadísticas:											
Media	1,767	0,815	2,895	1,645	2,199	0,755	1,403	2,879	0,568	1,225	0,712
Mediana	1,710	0,838	2,752	1,574	2,202	0,825	1,456	3,024	0,575	1,317	0,713
Desv. Est.	0,226	0,130	0,336	0,179	0,250	0,137	0,176	0,370	0,038	0,200	0,113
Máx.	2,176	1,020	3,431	2,008	2,597	0,869	1,606	3,221	0,609	1,430	0,933
Mín.	1,463	0,611	2,511	1,455	1,789	0,504	1,096	2,211	0,497	0,875	0,578

es comparativamente más bajo, tanto en porcentaje como en número de veces que se mejora.

Tabla 5.18: Mejoras obtenidas en los modelos de predicción al aplicar DARIM con DBSCAN, por conjunto de datos, en el estudio 5

	Mejora media por conjunto:					**Mejora media de la desviación estándar:**					
Conjunto	Acc.	TPrate	TNrate	f-Measure	AUC	Acc.	TPrate	TNrate	f-Measure	AUC	Tiempo
dataset1	0,851	-2,199	5,645	-0,255	3,897	0,635	1,336	-2,544	0,828	1,224	0,000
dataset2	0,386	1,636	0,186	1,351	-0,147	0,188	-0,045	0,878	-0,234	-0,580	0,484
dataset3	4,473	2,299	6,961	3,793	4,198	-1,062	0,393	3,715	-1,066	-0,620	0,484
dataset4	1,139	1,543	-1,075	0,743	1,053	1,933	2,898	4,735	1,284	7,404	0,000
dataset5	1,489	6,317	-0,241	5,087	1,240	-0,697	0,463	2,348	-1,689	-1,331	1,000
dataset6	0,937	0,744	0,987	1,152	1,296	1,889	6,030	3,957	4,778	4,429	0,000
dataset7	0,000	0,190	-0,147	0,079	0,285	-0,031	1,093	0,973	1,171	0,917	0,484
dataset8	4,499	2,688	5,335	5,590	3,614	2,331	4,475	7,046	3,199	-0,115	0,484
dataset9	4,225	1,008	5,161	6,792	4,661	0,716	1,730	4,767	-0,502	1,827	1,032
dataset10	1,662	-1,613	2,113	-1,306	1,474	0,616	3,070	2,951	3,916	5,454	0,000
dataset11	3,226	6,229	0,872	4,620	4,630	1,238	-1,959	1,802	-2,560	0,478	1,548
dataset12	1,466	1,843	0,806	1,345	2,117	0,912	0,089	0,946	-0,513	-0,161	0,000
dataset13	3,461	4,473	-0,154	2,710	5,870	3,532	5,588	3,404	3,216	7,117	0,516
dataset14	4,153	2,031	8,561	2,748	5,505	0,541	2,940	6,908	0,828	2,782	1,032
dataset15	1,047	0,427	4,201	0,603	0,813	0,998	2,697	0,948	0,882	1,797	0,613
dataset16	2,926	6,855	2,028	6,383	1,181	2,047	2,561	1,710	1,720	-0,108	0,516
dataset17	2,978	-0,645	5,242	-1,751	0,161	0,577	3,907	1,374	2,807	-1,558	0,323
dataset18	6,452	1,613	10,753	5,766	7,975	1,329	0,815	8,966	-0,498	1,682	0,387
dataset19	2,419	-3,226	6,072	1,472	6,745	2,518	0,134	9,418	-0,488	2,828	0,387
dataset20	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,387
dataset21	1,557	-3,763	3,952	0,680	1,987	2,409	-0,049	4,692	-2,034	3,004	0,548
dataset22	1,437	-1,526	5,536	0,936	1,401	0,345	0,216	4,332	-0,178	0,383	0,936
dataset23	1,613	1,466	1,769	1,634	2,340	1,500	3,507	3,367	2,762	1,391	0,452
dataset24	1,042	-2,248	4,435	0,609	2,459	-0,507	-1,403	4,288	-1,031	0,562	0,323
dataset25	2,823	0,281	9,319	1,583	3,569	-1,019	0,412	3,979	-1,278	-0,622	0,742
dataset26	1,333	1,367	1,283	1,085	0,753	0,484	1,415	1,012	0,648	0,306	3,032
dataset27	1,154	-0,237	3,027	0,771	1,071	0,038	1,047	-1,186	0,463	0,129	0,807
Estadísticas:											
Media	2,176	1,020	3,431	2,008	2,598	0,869	1,606	3,140	0,609	1,430	0,612
Mediana	1,557	1,008	3,027	1,345	1,987	0,635	1,093	3,367	0,463	0,562	0,484
Desv. Est.	1,585	2,711	3,208	2,325	2,199	1,140	1,981	2,825	1,876	2,355	0,611
Máx.	6,452	6,855	10,753	6,792	7,975	3,532	6,030	9,418	4,778	7,404	3,032
Mín.	0,000	-3,763	-1,075	-1,751	-0,147	-1,062	-1,959	-2,544	-2,560	-1,558	0,000
N# de veces que empeora, obtiene el mismo resultado, o mejora:											
Empeora	0	8	4	3	1	5	4	2	12	8	-
No mejora	1	1	1	1	1	1	1	1	1	1	-
Mejora	26	18	22	23	25	21	22	24	14	18	-
N# de veces que la mejora es mayor que 0, 1 o 2 %:											
Entre 0 y 1	4	4	4	7	4	11	7	4	5	6	-
Entre 1 y 2	11	7	2	7	8	5	5	4	3	5	-
Mayor de 2	11	7	16	9	13	5	10	16	6	7	-

5.4.6. Estudio 6. Detección de comportamientos anómalos para evitar el bajo rendimiento o el abandono

En los estudios contenidos en los anteriores apartados se demostró como la correcta detección y posterior tratamiento de los comportamientos anómalos de los estudiantes

FIGURA 5.25: Comparativa de mejoras obtenidas entre DARIM original y DARIM con DBScan

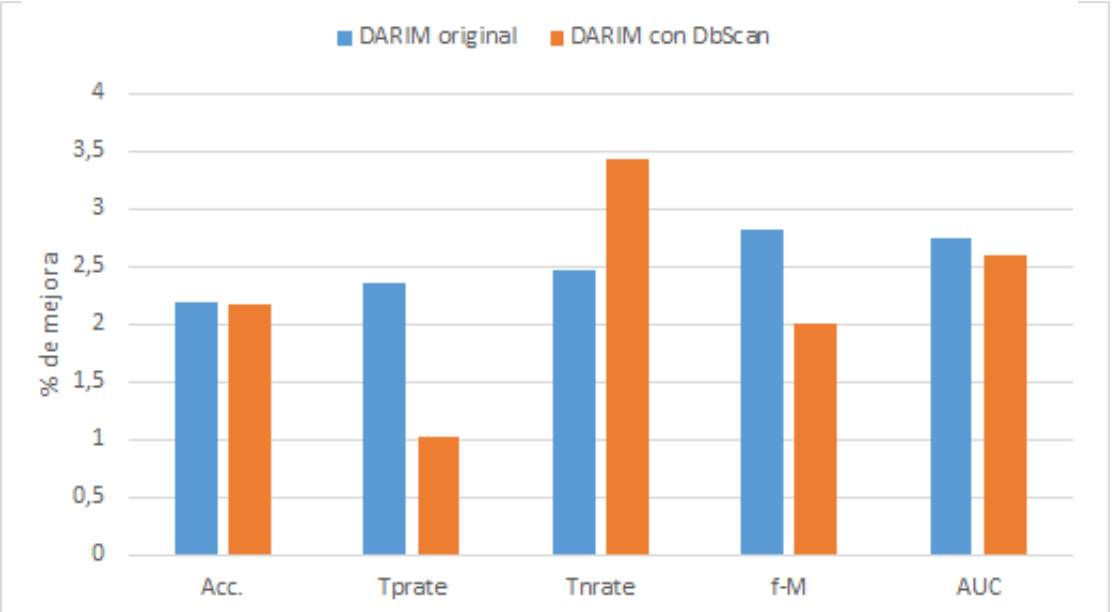
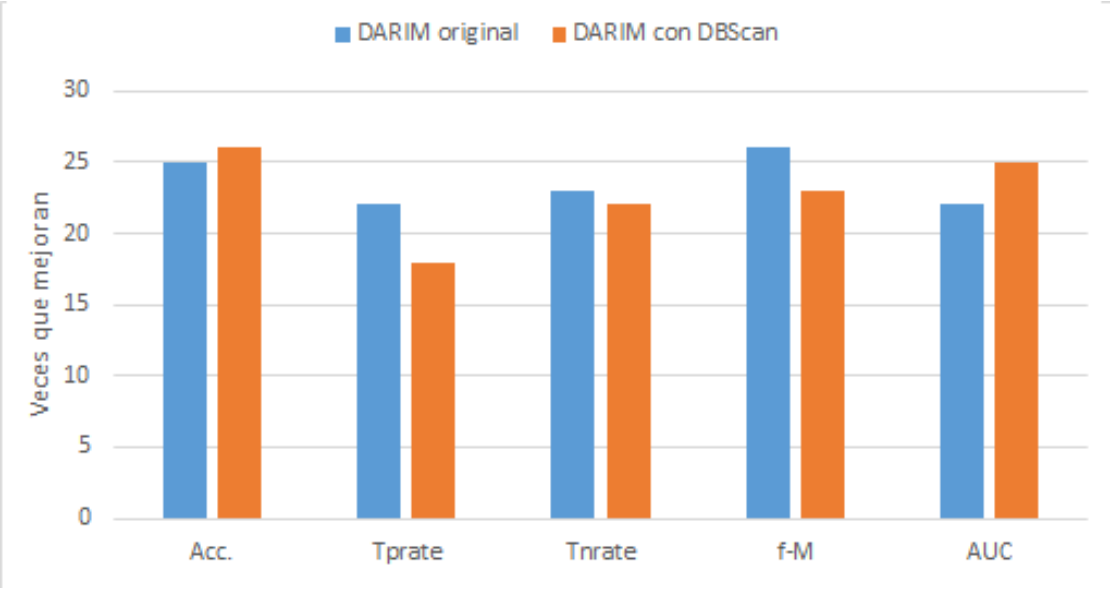


FIGURA 5.26: Comparativa de las veces que mejoran DARIM original y DARIM con DBScan



puede ayudar a mejorar la calidad de los modelos de predicción del rendimiento.

En este nuevo estudio, el proceso de detección de estudiantes con comportamientos anómalos se ejecutó con un nuevo objetivo: informar al profesor de aquellos estudiantes que, pese a haber realizado un esfuerzo notable, habían suspendido. Para ello, la detección se realizó no ya con la actividad y rendimiento globales, sino en base al rendimiento de los estudiantes en los diferentes ejercicios parciales y evaluables de los cursos, con lo que los profesores podrían recibir los avisos sobre los estudiantes suspensos con comportamientos anómalos durante el periodo de impartición de los cursos y, por tanto, serían capaces de actuar en consecuencia, posibilitando así que este tipo de estudiantes recibiese una realimentación personalizada que les ayude a mejorar y evitase que siguiesen suspendiendo o que abandonen el curso.

En la Figura 5.27 se muestran los tipos de estudiantes que pueden encontrarse en el proceso de detección de comportamientos anómalos, tras aplicar un algoritmo de clasificación teniendo como atributos la actividad de los estudiantes en el curso. En el eje X se refleja la clasificación del rendimiento de los estudiantes, mientras que en el eje Y se tiene su rendimiento real. Por un lado, se considera que los estudiantes que habiendo aprobado sean bien clasificados no necesitan de una realimentación especial por parte del profesor, ya que son estudiantes que con una alta actividad en el curso y que obtienen un rendimiento acorde a ello. Por otro lado, los estudiantes que, si bien aprobaron, se clasificaron como suspensos, no fueron objetivo de este estudio, ya que estos estudiantes no tienen un alto riesgo de abandono, pese a que su actividad sea más bien baja en comparación con la del resto de estudiantes con alto rendimiento. Los estudiantes suspensos clasificados como tales, si bien si podrían estar en riesgo de abandono, no necesitan a priori que el profesor les proporcione una retroalimentación personalizada, ya que el principal motivo de su bajo rendimiento es una baja actividad en el curso. Por último, se tiene a aquellos estudiantes que, pese a haber sido clasificados como aprobados por su alta actividad, suspendieron, y se encuentran por tanto en una situación de riesgo de abandonar o suspender el curso completo. Algunos de los estudiantes incluidos en este grupo pueden necesitar de un *feedback* más personalizado ya que la actividad dedicada al curso puede estar más cercana a los estudiantes aprobados que al resto de estudiantes suspensos. Sin esta retroalimentación personalizada, estos estudiantes pueden fácilmente desmotivarse y abandonar el curso.

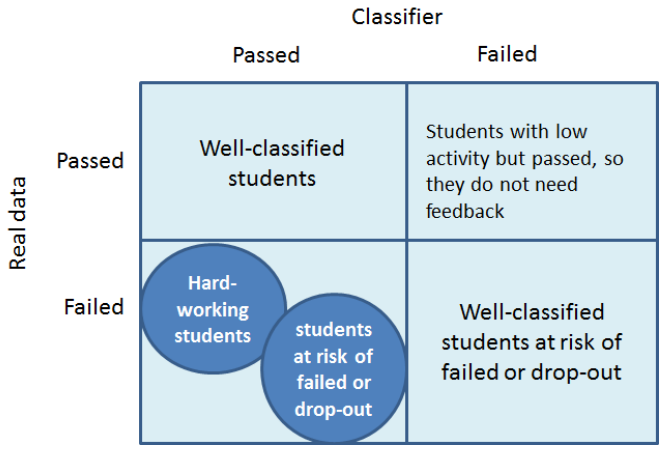


FIGURA 5.27: Estudiantes objetivo del estudio 6

El objetivo de este estudio, por tanto, consistió en detectar a estos estudiantes e informar al profesor sobre ello para que pueda actuar en consecuencia y evitar que abandonen.

Configuración de la experimentación: datos, técnicas y proceso seguido

1. Características de los cursos utilizados en el estudio

Se utilizó la actividad y rendimiento en los entregables de dos cursos virtuales. Uno de estos cursos es el que tiene identificador 23, en el que existen 3 entregables diferentes a lo largo de su desarrollo. El otro de los cursos no aparece en la Tabla del anexo A.2 debido a que únicamente se dispuso de él en el presente estudio. El número de estudiantes matriculados en este nuevo curso es de 119, habiendo 5 tareas entregables a lo largo de su impartición.

En ambos cursos, los profesores realizaron un *feedback* a los estudiantes tras cada entregable, informándolos de las fortalezas y debilidades de los trabajos entregados, así como del motivo de las calificaciones obtenidas.

2. Medidas de actividad y de rendimiento de los estudiantes

Dado que en este estudio la predicción del rendimiento y posterior detección de comportamientos anómalos no se realizó en base a todo el curso completo, si no en base a cada periodo en el que los estudiantes habían de realizar una entrega, la actividad extraída se correspondía con la realizada en dicho periodo, teniendo como rendimiento a predecir la calificación obtenida en el entregable.

Los atributos de actividad utilizados en este estudio fueron los siguientes: número total de acciones ("act"), número de visitas al temario del curso ("v-re"), visitas a los recursos

TABLA 5.19: Características principales de los dos conjuntos de datos utilizados, conteniendo la actividad de los estudiantes en el periodo comprendido entre el inicio del curso y la fecha del primer entregable para los cursos 23 y 26

Conjunto	Periodo (días)	Apro.	Susp.	No entregaron
d1	7	20	19	4
d2	7	85	34	0

SCORM ("v-sc"), vistas a la páginas de estadísticas ("v-da"), visitas a los comentarios de *feedback* realizados por el profesor ("v-fe"), visitas a las páginas html del curso ("v-co"), y cantidad de mensajes leídos ("v-fo"), escritos ("a-di") y respondidos en el foro ("p-fo"), además de la suma de los atributos "a-di" y "p-fo"("pa-fo").

Como atributo de clase, se indicó si el estudiante aprobó o suspendió cada uno de los entregables.

3. Conjuntos de datos generados

Al observar los datos, se constató en ambos cursos que la gran mayoría de los abandonos se produjeron tras superar la fecha del primer entregable, no habiendo ningún estudiante que, habiendo aprobado este trabajo, abandonase el curso. Por ello, en este estudio, el proceso de detección de estudiantes suspensos con comportamiento anómalo se aplicó sobre la actividad realizada por todos los estudiantes en el periodo comprendido entre el inicio del curso y la fecha del primer entregable. Se generaron, por tanto, dos conjuntos de datos con esta actividad y el rendimiento de los estudiantes en el primer entregable para cada uno de los cursos, cuyas características principales se resumen en la Tabla 5.19.

4. Método utilizado para la detección de comportamientos anómalos

Dados los buenos resultados obtenidos con DARIM en el estudio del apartado 5.4.4, en este nuevo estudio se aplicó un proceso similar para la detección de los comportamientos anómalos, al que se le añadieron algunas modificaciones para mejorarlo y adaptarlo al objetivo actual. A continuación, se muestran los pasos dados por el nuevo proceso:

1. *OutlierInstances:= conjunto de datos que almacena las instancias detectadas como anómalas;*

2. *RelevanceOfAttibures[] := array con la relevancia (peso) de cada uno de los atributos;*

3. *BadClassifiedInstances := se ejecuta algoritmo de clasificación y se almacenan las instancias mal clasificadas de los estudiantes suspensos.*

4. *AveragePerClass[] := array que almacena el valor medio de cada atributo con los estudiantes suspensos bien clasificados.*

5. *Cluster[] := almacena el valor medio de los atributos en cada uno de los clusters generados al aplicar un algoritmo de clustering sobre BadClassifiedInstances.*

6. *Distances[] = almacena la distancia euclídea de cada cluster en Cluster[] con respecto a los valores medios almacenados en AveragePerClass[], teniendo en cuenta el peso de cada atributo almacenado en RelevanceOfAttibures[];*

7. *Para cada cluster "clus" en Cluster[]*

   a) *si((Distances[clus] a Max[Distances[]]) menor que (Distances[clus] a Min[Distances[]]) entonces se añaden a outlierInstances todas las instancias contenidas en el cluster "clus";*

Como puede observarse, existen 3 modificaciones principales con respecto al proceso del apartado 5.4.4:

1. Dado que el objetivo es detectar estudiantes con alta actividad en el curso que suspenden los entregables y, posteriormente, abandonan, el proceso solamente realiza la detección de comportamientos anómalos sobre los estudiantes suspensos mal clasificados.

2. Se tiene en cuenta la relevancia de cada uno de los atributos de actividad utilizados, medida en base a los resultados de aplicar un algoritmo de selección de atributos. Cada atributo, por tanto, tendrá un peso específico en el momento de calcular la distancia entre cada cluster y el valor medio de las instancias de los estudiantes suspensos, utilizando por tanto la fórmula de la distancia euclídea con pesos (*weighed euclidean distance*), tal y como se muestra en la ecuación 5.1, siendo "c_i" el valor medio del atributo "i" en el cluster "c" y "m_i" el valor medio del mismo atributo para todas las instancias de los estudiantes suspensos:

$$dist(c, m) = \sqrt{\sum_{i=1}^{n}((w_i * (c_i - m_i))^2)} \tag{5.1}$$

3. El proceso no se limita a generar únicamente dos clusters sobre los estudiantes suspensos mal clasificados sino que, al igual que en la experimentación mostrada en el estudio del apartado anterior, el número de clusters puede ser mayor.

5. Técnicas de minería de datos utilizadas en el proceso

En este estudio, se utilizó el algoritmo J48 para realizar la clasificación de los estudiantes indicada en el paso 3 del proceso, siguiendo una estrategia de validación cruzada con 10 *folds*.

Como técnica de *clustering* para generar los grupos, se usó el algoritmo k-Means, al igual que en el proceso original, DARIM.

Por último, con objeto de dar peso a los atributos de actividad, previo paso a utilizar la distancia Euclídea con Pesos, se aplicó el algoritmo de selección de atributos *ClassifierSubSetEval*, con J48 como clasificador base y siguiendo una estrategia de evaluación cruzada con 10 folds, de forma tal que cada atributo tuviese un peso con valores entre 0 (nada relevante) a 10 (máxima relevancia).

Resultados de la experimentación

Al aplicar el algoritmo J48 sobre los conjuntos "d1" y "d2", se obtuvieron 2 árboles de decisión. En "d1", se tienen 7 estudiantes suspensos mal clasificados, mientras que en "d2" esta cifra asciende a 13.

Para detectar cuáles de esos estudiantes tienen una actividad similar a los estudiantes aprobados, el proceso descrito en el apartado anterior fue ejecutado varias veces, modificando manualmente en cada ejecución el valor k de k-Means en un rango de 2 a 5, concluyendo que, para el conjunto "d1", el mejor modelo de agrupamiento se obtiene con k=2, y para "d2", con k=4.

Los centroides de los clusters obtenidos para cada conjunto, así como la relevancia o peso de los atributos (columna "Peso") y el valor medio de los atributos para los estudiantes suspensos bien clasificados (columna "Media suspensos"), son mostrados respectivamente en las Tablas 5.20 y 5.21. En la penúltima de las filas, además, se indica el número de instancias que contiene el cluster y, en la última de las filas, se muestra la distancia de cada cluster con respecto a la media de los suspensos bien clasificados.

Tabla 5.20: Proceso de clustering y peso de los atributos en el conjunto "d1"del estudio 6

Atributo	Peso	C1	C2	Media suspensos
act	9	0.1835	0.4639	0.1245
v-re	4	0.093	0.438	0.1270
v-co	1	0.1017	0.3785	0.1239
v-fe	1	0	0.1667	0.0385
v-da	6	0.0435	0.3732	0.0920
a-di	2	1	0.1667	0.0769
p-fo	4	0	0.2	0.0308
pa-fo	1	0.1667	0.1944	0.0385
v-fo	2	0.3061	0.3299	0.0597
v-sc	3	0.075	0.3417	0.0952
N# ins.	-	1	6	-
dist. to avg.	-	2.0183	3.8936	-

Tabla 5.21: Proceso de clustering y peso de los atributos en el conjunto "d2"del estudio 6

Atributo	Peso	C1	C2	C3	C4	Media suspensos
act	10	0.679	0.049	0.163	0.137	0.07
N# ins.	-	2	5	2	4	-
dist. to avg.	-	0.609	0.021	0,093	0,067	-

Como puede observarse, en "d1", el cluster C1 sólo contiene una instancia, que representa al estudiante con la menor actividad de entre todos los mal clasificados, siendo esta bastante similar al resto de estudiantes suspensos bien clasificados. Por otro lado, los valores de actividad medios de los estudiantes del cluster C2 están notablemente más alejados de la media, por lo que son marcados como estudiantes con comportamientos anómalos.

En el conjunto "d2", el único atributo relevante es el número de acciones ejecutadas en el curso, siendo por tanto la única medida de actividad utilizada para detectar a los estudiantes suspensos con comportamientos anómalos. En este caso, son detectados dos estudiantes, pertenecientes al cluster C1, que es el que tiene una mayor distancia con respecto a la media. La distancia de los otros clusters con respecto a la media es notablemente más baja que la distancia que tienen a este cluster C1, por lo que se considera que, pese a haber sido mal clasificados, no tienen un comportamiento anómalo.

La Tabla 5.22 contiene los valores de actividad de los atributos más relevantes para respectivamente los 6 y 2 estudiantes detectados en cada curso, así como las calificaciones

obtenidas en el primer ("q1") y segundo entregable ("q2"), indicando qué estudiantes no realizaron la segunda entrega y, por tanto, abandonaron el curso.

En "d1", el valor de la mayor parte de los atributos para cada estudiante es notablemente mayor que el valor medio que pudo observarse en la Tabla 5.20, siendo esta diferencia destacable en el número de acciones ejecutadas en el curso ("act").

Analizando los datos, podemos observar cómo, por un lado, los estudiantes con identificadores d1s3, d1s4, d1s5 y d1s6 tienen unos altos indicadores de actividad y, pese a ello, obtuvieron una baja calificación en el entregable "q1". Sin embargo, estos mismos estudiantes aprobaron, posteriormente, el entregable "q2" con una alta calificación de 9 sobre 10. Como ya se mencionó en el apartado anterior, el profesor de este curso realizó un *feedback* a los estudiantes tras la corrección de los trabajos. Esto demuestra que la intervención del profesor en los casos de bajo rendimiento puede ayudar a los estudiantes en esa situación a mejorar.

No obstante, en el caso del estudiante d1s2, incluso teniendo una considerable actividad, encontramos que, tras suspender el entregable "q1", abandonó el curso. En este caso, el *feedback* proporcionado por el profesor no fue útil. Dado que el profesor desconocía que este estudiante tenía una alta actividad en el curso, no pudo personalizar los comentarios enviados como *feedback* en base a esta información, tratando de identificar mejor los problemas que dicho estudiante pudo tener en el desarrollo del entregable y redactando, por tanto, un mensaje en un tono más "motivador" y adaptado a este caso concreto.

Por último, el estudiante d1s1, pese a tener una alta actividad, no realizó la primera entrega, por lo que el profesor perdió la oportunidad de enviarle cualquier tipo de *feedback* y, por tanto, rescatarlo. Con respecto al conjunto "d2", el número de acciones realizadas por ambos estudiantes es muy alto en comparación con el valor medio de los estudiantes suspensos bien clasificados. De hecho, uno de estos estudiantes, d2s2, tiene una calificación de 4.5 sobre 10, muy próxima al aprobado. En este caso, el *feedback* proporcionado por el profesor tuvo éxito, ya que en el siguiente entregable, este estudiante alcanzó una calificación de 8.5 sobre 10. Igualmente, el estudiante d2s1, con una calificación en "q1"de 3 sobre 10, mejoró su rendimiento notablemente en "q2".

Tabla 5.22: Actividad de los estudiantes y calificaciones en los 2 primeros entregables de los cursos ("q1" y "q2" respectivamente) del estudio 6

Estudiante	act	v-re	v-da	p-fo	v-sc	q1	q2
Conjunto d1							
d1s1	0.23	0.30	0.24	0.00	0.19	0	0(dropout)
d1s2	0.20	0.16	0.15	0.00	0.34	3	0(dropout)
d1s3	0.91	1.00	0.72	0.60	0.63	4	9
d1s4	0.50	0.44	0.20	0.20	0.23	0	9
d1s5	0.58	0.42	0.43	0.40	0.31	3	9
d1s6	0.37	0.30	0.50	0.00	0.36	0	9
Conjunto d2							
d2s1	0.84					3	8
d2s2	0.51					4.5	8.5

5.5. Conclusiones y trabajo futuro

El proceso de tratamiento de comportamientos anómalos de los estudiantes de cursos virtuales puede mejorar los modelos de predicción del rendimiento de estos estudiantes. No obstante, para obtener esta mejora, es necesario escoger el proceso adecuado para la tarea. Por una parte, técnicas bien conocidas de detección de *outliers*, como LOF y ECODB, no han resultado ser especialmente eficientes. En el caso de LOF, al no tener en cuenta la clase de las instancias, no detecta correctamente a aquellos estudiantes que tienen comportamientos anómalos en comparación al resto de los de su misma clase, esto es, suspensos o aprobados. ECODB, que sí detecta *class-outliers*, si bien mejora en términos de *accuracy*, tiene dificultades para mejorar a la vez en la predicción de suspensos y aprobados, obteniendo mejoras o empeorando los resultados según el conjunto de datos utilizados.

Los algoritmos utilizados en el estudio 2, Adaboost, Multiboost, Bagging y RandomForest, muestran que no se puede afirmar de forma generalizada que todos ellos sean robustos o tolerantes ante conjuntos de datos con *outliers* o comportamientos anómalos, siendo bastantes irregulares en las mejoras obtenidas, y llegando a empeorar en numerosos conjuntos de datos al igual que ECODB. De hecho, es en la predicción de los estudiantes suspensos, *TPrate*, en dónde todos fallan con más frecuencia a la hora de mejorar los modelos. Siendo estos estudiantes los más relevantes de cara a ser bien clasificados, se concluye que los mencionados algoritmos tienen un resultado mediocre, llegando en casos como el de Multiboost a un empeoramiento de esta medida.

El proceso de Voto Mayoritario del estudio 3 obtiene unos notablemente buenos resultados, obteniendo notables mejoras en un gran porcentaje de los conjuntos de datos y clasificadores utilizados en términos de *accuracy* y *TNrate*, y también en la reducción de la desviación estándar de los modelos obtenidos con el proceso 10-CV. No obstante, las mejoras obtenidas en la clasificación de suspensos, *TPrate*, son bastante bajas, mostrando en esta medida un comportamiento algo irregular al empeorar su rendimiento en una notable cantidad de conjuntos de datos. Más aún, en los modelos que inicialmente tienen un *accuracy* por encima del 80 %, el proceso de Voto Mayoritario empeora esta medida.

En cuanto a DARIM, cuyos resultados se exponen en el estudio 4, esta nuevo proceso propuesto ha mostrado ser el más efectivo de cara a mejorar los modelos de predicción. Si bien porcentualmente el proceso de Voto Mayoritario alcanza mayores mejoras en términos como *accuracy* o *TNrate*, DARIM obtiene un mayor equilibrio en las mejoras al no sacrificar el *TPrate* en aras de mejorar aún más el *TNrate*. Esto sucede debido a que DARIM, al contrario que el proceso de Voto Mayoritario, fuerza a buscar comportamientos anómalos en las instancias mal clasificadas de ambas clases. De hecho, DARIM obtiene mayores mejoras en la clasificación de los estudiantes suspensos que las obtenidas por el proceso de Voto Mayoritario. Además, las mejoras de DARIM en el resto de medidas son más estables, ya que el proceso de Voto Mayoritario tiene unas estadísticas de mejoras mínimas por conjunto de datos mucho más bajas que las alcanzadas con DARIM.

Por otra parte, para conjuntos de datos con pocos instancias como son los del entorno educativo, DARIM puede ser fácilmente utilizado como proceso automático o de caja negra, en el que el usuario no tiene necesidad de configurar sus parámetros, generando siempre dos clusters para cada conjunto de instancias mal clasificadas.

También se ha mostrado una primera aproximación a la modificación de DARIM en la que se usa otro algoritmo de *clustering* diferente, DBSCAN, concluyendo que dependiendo del conjunto de datos y de la distribución de los mismos, puede ser más conveniente utilizar esta nueva versión. No obstante, esto necesitará de un estudio más profundo en un futuro.

Como trabajo futuro, también se puede valorar la utilización de algoritmos que devuelvan índices recomendando el número de clusters a generar con DARIM, siempre

que el conjunto de datos sea lo suficientemente grande como para que la configuración propuesta de dos clusters se muestre insuficiente. En base a ello, se podrá extender la experimentación a otras áreas y conjuntos de datos fuera del campo educativo, y constatar así si DARIM obtiene los mismos buenos resultados.

Por otra parte, el estudio 6 demuestra que el proceso de selección de estudiantes cuya actividad en cursos *e-learning* no se corresponde con su rendimiento puede ayudar al profesor a detectar a este tipo de estudiantes con objeto de poder personalizar los comentarios de realimentación, y de esta manera mejorar su rendimiento y evitar que abandonen.

El proceso propuesto, además, muestra su utilidad al detectar tanto estudiantes a los que el *feedback* del profesor les sirvió para mejorar sus calificaciones como, más importante aún, a estudiantes que abandonaron el curso debido a que los comentarios del profesor no fueron adecuadamente personalizados, dado que éste desconocía la información al respecto del alto grado de actividad que habían desempeñado. Este tipo de información, por tanto, podría ayudar al profesor a mejorar el *feedback* de estos estudiantes y, de esta forma, tratar de conseguir que se mantengan en el curso y mejoren su rendimiento.

Capítulo 6

Estudio y aplicación de otro tipo de técnicas para mejorar los modelos ofrecidos por ElWM

En este capítulo, se presentan los resultados obtenidos hasta ahora en dos trabajos de investigación desarrollados en la Universidad de Cantabria, en colaboración con otros investigadores, y relativos a los objetivos de esta tesis.

En el primero de ellos, descrito en el apartado 6.1, se realizó un estudio del comportamiento social de los estudiantes en los foros de los cursos. En base a ello, se pudieron determinar nuevos atributos de los estudiantes que puedan ser usados como predictores de su rendimiento. Esta investigación se desarrolló en colaboración con otro investigador en formación de la Universidad de Cantabria, Camilo Palazuelos, a quién correspondió la tarea de aplicación de técnicas de análisis de redes sociales y la extracción de los atributos que definen el comportamiento social de los estudiantes. El resto del proceso de minería de datos, incluyendo el análisis de los atributos, pre-procesado, extracción de modelos y evaluación de resultados corresponden a las tareas realizadas por el autor de esta tesis.

El segundo de estos estudios, descrito en el apartado 6.2, tuvo como objetivo la mejora en la extracción de reglas por parte de E-learning WebMiner, de forma que se redujese el número de reglas eliminando redundancia, con objeto de facilitar al profesor que utilice la herramienta la comprensión y análisis del modelo. En este estudio se colaboró con

diversos investigadores de la Universidad de Cantabria en la mejora de un algoritmo de reglas de asociación que eliminase la redundancia entre reglas, correspondiendo al autor de esta tesis el trabajo de extracción de los datos, pre-procesado, aplicación de las técnicas de reglas de asociación, así como la comparativa y evaluación de los resultados.

6.1. El poder predictivo del Análisis de Redes Sociales en la educación

En su trabajo, Siemens [258] afirmó que durante los últimos 20 años, la tecnología ha redefinido cómo (los seres humanos) vivimos, nos comunicamos, e incluso cómo aprendemos. En la actualidad, es bastante frecuente encontrarse con cursos virtuales en los que los profesores requieren y hacen uso de tecnologías Web 2.0 para desarrollar los contenidos, y a su vez promocionar el aprendizaje social y colaborativo de los estudiantes, fomentando así la interacción entre ellos. Algunos ejemplos de actividades colaborativas que se encuentran comúnmente en el ámbito educativo son la búsqueda de contenidos, la escritura colaborativa o los foros de discusión. Este tipo de actividades pueden ser desarrolladas gracias a las herramientas que ofrecen las plataformas de aprendizaje *e-learning* como Moodle o Blackboard, como son los foros, blogs y wikis.

En este escenario, en donde el nivel de interacción entre los diferentes actores del curso (estudiantes, profesores y recursos) puede ser muy alto, surge la posibilidad de analizar cómo se produce esta interacción y cuál es el grado de participación de los estudiantes en las actividades propuestas en estas herramientas. Este tipo de análisis puede resultar útil para el profesor, con objeto de poder analizar si el comportamiento social de sus estudiantes tiene repercusión en su rendimiento en el curso. El conocimiento extraído de este tipo de análisis puede ser también utilizado para conocer, por ejemplo, que estudiantes se encuentran aislados o quiénes tienen un mayor número de conexiones con el resto.

En el presente apartado, se muestra cómo las técnicas de Análisis de Redes Sociales (*Social Network Analysis* en inglés, SNA por sus siglas) pueden ser aplicadas sobre los foros de cursos virtuales, de cara a extraer medidas que caractericen el comportamiento social de los estudiantes, y cómo estas medidas pueden utilizarse en la tarea de predicción de rendimiento de los estudiantes.

Este apartado se organiza de la siguiente forma: en la sección 6.1.1 se expone un resumen de los principales trabajos en los que se abordan la aplicación de técnicas SNA dentro del área educativa y *e-learning*. En la sección 6.1.2 se resume brevemente cómo se ha realizado el proceso de modelado de redes sociales respecto de los foros en cursos virtuales, y se explican los tipos de redes modeladas. La sección 6.1.3 muestra las hipótesis de partida y la organización de dos estudios realizados. En la sección 6.1.4 se explica el significado de cada una de las medidas de centralidad utilizadas en los estudios, además de indicar en cuáles de ellos se incluyen. La sección 6.1.5 se divide en dos subsecciones, en cada una de las cuáles se muestran la configuración, desarrollo, resultados y conclusiones de los dos estudios mencionados en la sección 6.1.3. Por último, en la sección 6.1.6 se establecen las conclusiones finales en base a los resultados obtenidos en los dos estudios mostrados.

6.1.1. Estado del arte: Análisis de Redes Sociales en entornos e-learning

El Análisis de Redes Sociales es un proceso que consiste en estudiar las relaciones entre actores conectados y que interactúan de alguna forma entre ellos, representando a estos actores y sus relaciones como una red o grafo, en el que cada nodo representa a un actor concreto, y cada enlace simboliza una conexión o interacción entre pares de actores, como puede ser, por ejemplo, una relación de amistad.

En su libro [259], Moreno estableció los conceptos fundacionales de la sociometría, precursora del SNA. Desde entonces, se han venido aplicando una gran variedad de técnicas SNA en diversos campos, y con muy diferentes objetivos. Así, por ejemplo, actualmente se pueden encontrar estudios acerca de diferentes tipos de interacciones y conexiones, y que han dado solución a problemas como la detección de patrones entre criminales y terroristas [260] o la identificación de actores de especial relevancia en las redes sociales [261, 262].

Existen diferentes tipos de medidas de SNA que permiten modelar y extraer información sobre como interaccionan los usuarios conectados en una red social o cuál es su rol dentro de la misma. Un tipo de estas medidas, denominadas medidas de centralidad, permiten determinar cuáles son los nodos (usuarios) más relevantes en la red. Estas medidas, utilizadas en los estudios que componen el presente capítulo, son presentadas en el apartado 6.1.4.

En el campo educativo, las técnicas de SNA han sido aplicadas e incluidas en estudios con diversos objetivos. Un ejemplo lo podemos encontrar en [263], en donde los autores utilizaron SNA para analizar las relaciones de amistad entre estudiantes de primaria con objeto de determinar el efecto que tienen estas conexiones de amistad en su educación.

No obstante, fue con el desarrollo de las tecnologías de educación virtual cuando se facilitó la posibilidad de extraer redes sociales que modelen las relaciones sociales entres los estudiantes, gracias a que este tipo de plataformas almacenan la actividad que los estudiantes realizan en los cursos, ofreciendo herramientas que permiten que los estudiantes interaccionen entre ellos. Sin embargo, aún en la actualidad no existe un gran número de estudios en este área, tal y como concluyeron Karina et al. [264] en su revisión sistemática de trabajos en los que se aplica SNA a entornos *e-learning*, publicada en 2014. Los autores de este estudio identificaron únicamente 37 trabajos publicados entre los años 1999 y 2012, y de entre ellos, solamente en 27 se utilizan medidas de centralidad. Los objetivos de estos trabajos son bastante variados. Así, se pueden encontrar trabajos en los que los autores utilizaron SNA para ayudar al profesor a optimizar el curso de cara a mejorar el proceso de aprendizaje [265], en los que se estudió la influencia del estilo de tutoría en la interacción de los estudiantes [266], o incluso estudios en los que se combinaron técnicas de análisis de contenidos y SNA [267–269].

En esta revisión sistemática también se encuentran trabajos enfocados en modelar la relación existente entre las medidas SNA y el rendimiento de los estudiantes. Dawn et al. [270] concluyeron en su estudio que existen diferencias significativas en las redes sociales construidas con estudiantes de alto y bajo rendimiento respectivamente. En otro estudio [271], los autores concluyeron que el nivel de actividad en el foro de un curso *e-learning* está altamente relacionado con el rendimiento de los estudiantes, algo en lo que también coincidieron Stepanyan et al. [272] al afirmar que los estudiantes con una alta interacción recíproca obtienen mejores calificaciones.

En estos trabajos, las redes sociales fueron construidas en base a la interacción de los usuarios en herramientas como los microBlogs de Twitter [272], en entornos enfocados al aprendizaje colaborativo [266, 273–275], o incluso utilizando recursos que pueden encontrarse en entornos *e-learning* del tipo de Moodle y Blackboard, como son las wikis del curso [276] y los foros de discusión [277].

Desde la publicación de la mencionada revisión sistemática, han surgido nuevos trabajos que aplican SNA en entornos *e-learning*, denotando un creciente interés en el área, si bien una búsqueda en las bases de datos científicas *Scopus* o *Web of Knowledge* muestra que aún es un área poco explorada y con mucho camino por recorrer. Utilizando un marco de aprendizaje colaborativo, Gewerc-Barujel et al. [278] realizaron un estudio en el que, haciendo uso para construir las redes sociales de los mensajes en foros, los comentarios en blogs, los correos electrónicos y los ficheros subidos, analizan las interacciones de los estudiantes de un curso de nivel universitario con objeto de determinar la intensidad y pertinencia de las aportaciones de los estudiantes. En [279] se analizó no solamente la interacción entre estudiantes, sino también entre los estudiantes y el profesor, concluyendo que la asistencia de los profesores a los estudiantes tiene un notable impacto en el progreso de estos últimos. Romero-Moreno [280] utilizó la actividad de los estudiantes en los foros de Moodle y Blackboard para analizar sus interacciones, proponiendo finalmente una metodología que evaluase la colaboración existente entre los estudiantes en estos foros. En Fishpaw et al.[281] se analizó la frecuencia y tipo de uso de tecnologías que permitieran la interacción social con el objetivo de mejorar el diseño de la educación virtual.

En los dos últimos años, 2014 y 2015, también han visto la luz algunos estudios en los que se relaciona el SNA con el rendimiento de los estudiantes. Carceller et al. [282] estudiaron la relación que existe entre las medidas de centralidad y la nota final obtenida por estudiantes de cursos tanto completamente virtuales como semipresenciales, concluyendo que hay una alta correlación entre algunas de estas medidas y la nota, como son el *Degree* o el *Eigenvector*. Las mismas conclusiones se extraen en otro estudio de Putnik et al. [283], publicado a mediados de 2015.

Con respecto a la combinación de SNA con técnicas de minería de datos para la predicción del rendimiento de los estudiantes, hay que destacar un trabajo ya mencionado en en el apartado 3, de Romero et al. [135], en el que se combinaron dos medidas de centralidad con otras medidas de actividad de los estudiantes en los foros de cursos alojados en Moodle, como el número de mensajes escritos y leídos, para predecir dicho rendimiento.

En conclusión, los estudios encontrados en la literatura prueban que las técnicas SNA son especialmente útiles para modelar el comportamiento de los estudiantes en cursos de educación virtual en los que se realizan actividades de forma colaborativa. En los

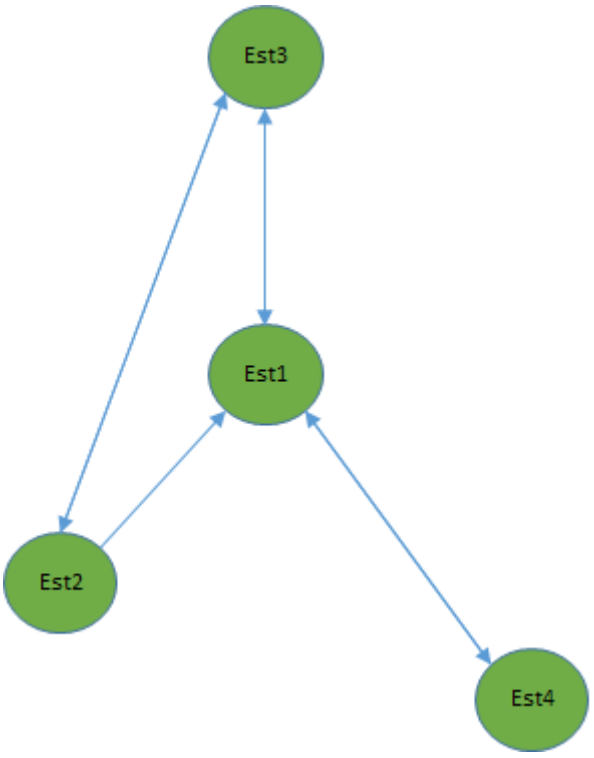
LCMS como Moodle, esta interacción puede ser recogida y analizada en herramientas como los foros, en los que los estudiantes escriben mensajes y responden a los de otros compañeros.

En la predicción del rendimiento de los estudiantes en cursos virtuales, existen algunos trabajos que sugieren la utilidad que las medidas de centralidad pueden tener como atributos predictores, si bien el bajo número de estos estudios hace surgir la necesidad de realizar nuevos estudios para contrastar este hecho.

6.1.2. Modelado de las redes sociales basadas en foros de cursos e-learning

Tras realizar un análisis de todos los cursos virtuales disponibles, se identificaron finamente 5 con una actividad en los foros que pudiera permitir el modelado de estas redes y su posterior análisis. Para cada uno de estos cursos, se construyó un modelo de red social. Para ello, se modelaron las interacciones de forma tal que los nodos de la red social representaran a cada uno de los estudiantes, mientras que las conexiones entre los estudiantes informaban de que ha habido una interacción o respuesta en el foro por parte de un estudiante a otro, o por parte de un estudiante al profesor y viceversa. Esto puede entenderse mejor al observar la Figura 6.1, en la que se muestra un ejemplo de interacción en el foro entre 4 estudiantes, identificados como “Est1”, “Est2”, “Est3” y “Est4”. Estos estudiantes son los nodos de la red, representados con un círculo. Las flechas simbolizan las conexiones entre los nodos, e informan de que el estudiante representado por el nodo del que sale la flecha o conexión ha escrito una respuesta en el foro a un mensaje escrito por el estudiante al que va dirigida dicha conexión. En el ejemplo mostrado, el estudiante “Est1”ha recibido, al menos, un mensaje de cada uno de los otros 3 estudiantes, lo que se representa con las flechas direccionales que van desde cada uno de estos estudiantes hacia “Est1”. Este mismo estudiante “Est1”habría enviado al menos un mensaje en el foro a “Est3” y “Est4”, existiendo por tanto una comunicación bidireccional entre ellos, al igual que ocurre entre “Est3” y “Est2”. La falta de interacción entre, por un lado, “Est4”, y por otro, los estudiantes “Est2” y “Est3”, indica que no ha habido una interacción directa entre ellos, es decir, “Est4” no les ha enviado ningún mensaje ni ha recibido ninguno por parte de estos dos estudiantes.

FIGURA 6.1: Ejemplo de red social con las interacciones entre estudiantes de un foro en un curso e-learning



Al extraer los modelos de red social de los cursos, nos encontramos con 2 tipos de redes con una estructura diferente: centralizadas y distribuidas. La particularidad de las redes centralizadas consiste en que existe un nodo central que recibe y realiza la mayor parte de las interacciones con respecto a los demás nodos de la red, existiendo muy poca interacción entre los demás nodos. Por el contrario, las redes distribuidas se caracterizan por el hecho de que en ellas no existe un sólo nodo central como el descrito, sino que existe un mayor nivel de interacción entre todos los nodos, si bien pueden existir un pequeño subconjunto de nodos a través de los cuales se realicen la mayor parte de las interacciones.

6.1.3. Hipótesis de partida, organización y resumen de los estudios

La hipótesis principal de partida fue la siguiente: **¿Pueden las medidas de centralidad, extraídas con técnicas SNA, ayudar a mejorar la predicción del rendimiento de los estudiantes de cursos virtuales, utilizando para ello técnicas de minería de datos?**. Para responder a esta pregunta, se realizaron dos estudios diferentes:

1. **Estudio 1. Redes sociales centralizadas en los foros de cursos *e-learning***
   En este primer estudio, mostrado en la sección 6.1.5.1, se extrajeron las medidas de centralidad del foro de un curso en el que la red social obtenida está fuertemente centralizada en el profesor del curso, esto es, el profesor es la principal vía de comunicación en la red, siendo el usuario que envía y recibe gran parte de los mensajes. En este estudio, se utilizaron las medidas de centralidad tanto por sí solas, como junto a otras medidas de actividad en el curso, para tratar de predecir el rendimiento o la posibilidad de abandono de los estudiantes. Como conclusión, se estableció que las medidas de centralidad son notablemente útiles, por sí solas, para realizar la tarea de predicción, además de mejorar los modelos cuando se utilizan conjuntamente con otras medidas de actividad.

2. **Estudio 2. Redes sociales distribuidas en los foros de cursos *e-learning***
   Se realizó un estudio similar al del estudio 1, pero esta vez utilizando la interacción en los foros de dos cursos *e-learning* en el que la red es distribuida. Las conclusiones fueron similares a las del estudio 1, estableciendo además que las medidas de centralidad tienen aún más utilidad en la tarea de predicción cuando la red es distribuida.

En el siguiente apartado 6.1.4, se indica que medidas de centralidad fueron utilizadas en cada estudio, además del significado que adquieren en las redes construidas con la interacción en los foros.

6.1.4. Medidas de SNA utilizadas en los estudios: las medidas centralidad

Este apartado se divide en dos secciones. En la sección 6.1.5.1 se muestra un caso de estudio en el que la red obtenida del foro es centralizada. Por otra parte, en el apartado 6.1.5.2 se realiza el mismo estudio en el que las redes son distribuidas.

De entre las medidas de centralidad existentes, en este trabajo se utilizan aquellas más relevantes y que tienen un uso más extendido en la literatura, con objeto de medir la interacción social de los estudiantes en los foros de los cursos virtuales estudiados. En la Tabla 6.1 se define el significado de estas medidas al construir la red social con los foros, en donde los nodos representan a los estudiantes y las conexiones son los mensajes

enviados o respondidos entre ellos. En la última columna de la tabla se indica en cuál de los dos estudios realizados han sido utilizadas:

6.1.5. Configuración, proceso, resultados y conclusiones de los estudios

En este apartado, se muestran la configuración utilizada, el proceso seguido, así como los resultados y conclusiones obtenidas en los dos estudios descritos en el apartado 6.1.3.

6.1.5.1. Estudio 1. Redes sociales centralizadas en los foros de cursos *e-learning*

En este primer experimento, se realizó un estudio de la potencialidad predictiva de las medidas de centralidad, extraídas al aplicar técnicas de SNA sobre foros de cursos en los que la red social está fuertemente centralizada, esto es, en dónde existe un usuario que recibe y/o envía la mayor parte de los mensajes en el foro, el cuál por tanto canaliza la mayor parte de la interacción entre usuarios.

Configuración y proceso seguido

En el desarrollo de este estudio se siguió el siguiente proceso:

1. Selección de cursos con interacción fuertemente centralizada

De entre los 5 cursos mencionados en la sección 6.1.2, en este experimento se utilizaron 3 de ellos en los que la actividad en los foros está fuertemente centralizada en un solo usuario. Estos 3 cursos se corresponden con los cursos con identificador 1, 2 y 3 del Anexo A.2.

En estos cursos, por tanto, nos encontramos con que la mecánica de uso de los foros consiste en que el profesor realiza anuncios relacionados sobre el curso y sus contenidos y los estudiantes responden a dichos anuncios consultando dudas sobre los mismos, o bien los estudiantes plantean dudas sobre los contenidos y es principalmente el profesor quien responde a dichas dudas, existiendo poca interacción entre los propios estudiantes.

2. Construcción de la red social, extracción y análisis de las medidas de centralidad

Para cada uno de los 3 cursos, se construyó la red social en base a la interacción en los foros, se realizó un análisis de las características de la red y, finalmente, se extrajeron y analizaron las medidas de centralidad, según se indica en la sección 6.1.4, que definen

Tabla 6.1: Medidas de centralidad

Medida	Significado	Estudios
Degree	Nº de conexiones entrantes y salientes del nodo (mensajes respondidos y recibidos)	1,2
Outdegree	Nº de conexiones salientes (mensajes respondidos)	1,2
Indegree	Nº de conexiones entrantes (mensajes recibidos)	1,2
Closeness	Grado de interés de los mensajes publicados por el estudiante	2
Betweenness	Grado de intermediación del nodo (cuantas veces el estudiante actúa como intermediario en la comunicación entre otros estudiantes)	1,2
Eigenvector	Influencia del nodo en la red. Valores altos representan a estudiantes con conexiones a otros estudiantes también influyentes en la red.	2
Hub	Generalización del Eigenvector. Un estudiante tiene un alto Hub si tiene conexiones salientes (envía mensajes) a estudiantes con gran cantidad de conexiones entrantes	1,2
Authority	Generalización del Eigenvector. Un estudiante tiene un alto Authority si tiene conexiones entrantes (recibe mensajes) de nodos con gran cantidad de conexiones salientes.	1,2
Information centrality	Medida introducida en Sthepenson et al. [284] y basada en la información que puede ser transmitida entre 2 nodos	2
Clique membership count	Nº de "cliques" a los que pertenece el estudiante, siendo un "clique" un grupo de 3 o más estudiantes que tienen muchas conexiones entre sí y muy pocas con el resto	2
Local clustering coefficient	Identifica a los estudiantes a través de los cuáles pasa la información en el foro	2
ClusteringDegree	Mide el grado en el que un estudiante tiende a forma grupo con otros estudiantes	2

la interacción social de cada uno de los estudiantes. A estas medidas se añadió un atributo predictor más, llamado en el estudio "top 3", que indica en cuantas medidas de centralidad de las utilizadas el estudiante obtiene uno de los tres valores más altos.

3. Generación de conjuntos de datos con la actividad y rendimiento de los estudiantes

En este estudio se generaron y utilizaron 3 conjuntos de datos diferentes, todos ellos conteniendo la información relativa a los estudiantes de los 3 cursos mencionados en el paso 1:

- **performance.dat**: contiene como atributos predictores, para cada estudiante, los valores de las medidas de centralidad extraídas en el apartado 2, y como clase a predecir el rendimiento obtenido en el curso (suspende/aprueba).
- **dropout.dat**: misma información que performance.dat, excepto porque la clase a predecir no indica el rendimiento de los estudiantes, sino si abandonaron o no el curso.
- **mixed.dat**: contiene, para cada estudiante, además de los valores de las medidas de centralidad, 6 atributos predictores que miden su actividad en el curso en base al tiempo y sesiones totales y medias dedicadas a la asignatura y el número de mensajes escritos y leídos tanto en el foro como en la herramienta de correo interno que tiene implementada el curso. En este caso, el nuevo conjunto de datos tiene 3 valores de clase, indicando si el estudiante aprobó, suspendió o abandonó la asignatura.

4. Evaluación de la potencialidad predictora de las medidas SNA

Sobre los conjuntos de datos performance.dat y mixed.dat se aplicaron varios algoritmos de selección de características, con objeto de determinar la utilidad de las medidas de centralidad como predictoras del rendimiento de los estudiantes. En este apartado se comentarán los resultados obtenidos por las técnicas *CfsSubSetEval* y *ClassifierSubSetEval*.

5. Aplicación y evaluación de clasificadores

Se utilizaron 6 clasificadores de diferente paradigma, sobre los conjuntos de datos performance.dat y dropout.dat, para determinar tanto si era posible obtener modelos de predicción del rendimiento o abandono de los estudiantes fiables utilizando únicamente

medidas SNA como atributos predictores. Estos clasificadores son JRip, OneR, Naïve-Bayes, BayesianNetwork, J48 y RandomForest.

Además, se aplicaron los mismos clasificadores sobre el conjunto "mixed.dat", tanto utilizando conjuntamente las medidas de SNA y las de actividad, y solamente las de actividad, con objeto de determinar si las medidas SNA mejoraban la fiabilidad de los modelos de predicción al añadirlas a las de actividad de los estudiantes.

Resultados de la experimentación

En la Figura 6.2 se muestra la red social construida tomando la interacción en el foro del curso 1. Como puede observarse, es una red fuertemente centralizada, en la que el nodo 1, que representa al profesor de la asignatura, realiza y recibe la mayor parte de las interacciones. Como ya se ha comentado anteriormente en esta sección del capítulo, al realizar un análisis de los mensajes, y en base a lo observado en esta red, se puede concluir que en este curso el foro se ha utilizado expresamente como herramienta con la cuál el profesor realiza anuncios importantes y resuelve las dudas de los estudiantes sobre los contenidos, existiendo una interacción menor entre los propios estudiantes para resolver entre ellos las dudas o trabajar de forma colaborativa. Este tipo de red, con el profesor como nodo central, es el mismo que se obtiene al analizar los cursos 2 y 3.

En la Tabla 6.2 se muestran las medidas de centralidad *Degree*, *Indegree*, *Outdegree*, *Betweeness*, *Authority* y *Hub* para los 3 nodos con un mayor valor en cada una de dichas medidas. Como era de esperar al observar la red, el nodo con un mayor *Degree*, esto es, con un mayor número de interacciones tanto de entrada como de salida, es el nodo 1, que representa al profesor. Su valor de *Degree*, además, está bastante alejado del obtenido por los nodos 3 y 5, que representan a los dos estudiantes con un mayor *Degree* tras el propio profesor. Más aún, el *Outdegree* del profesor, que nos informa del número de respuestas dadas en el foro por el usuario (interacciones de salida de un nodo), tiene un valor aún mucho más alto y distante, 157, que el del segundo nodo con mayor valor, que es de sólo 6. Igualmente, es el profesor el que tiene un mayor *Betweeness*, lo que indica que este usuario actúa frecuentemente de intermediador entre los estudiantes.

Por otro lado, se observa que los valores de *Authority* e *Indegree* son más altos para 3 de los estudiantes del curso, además de no existir una diferencia tan grande entre ellos como la que existía con respecto al profesor al analizar las otras medidas de centralidad.

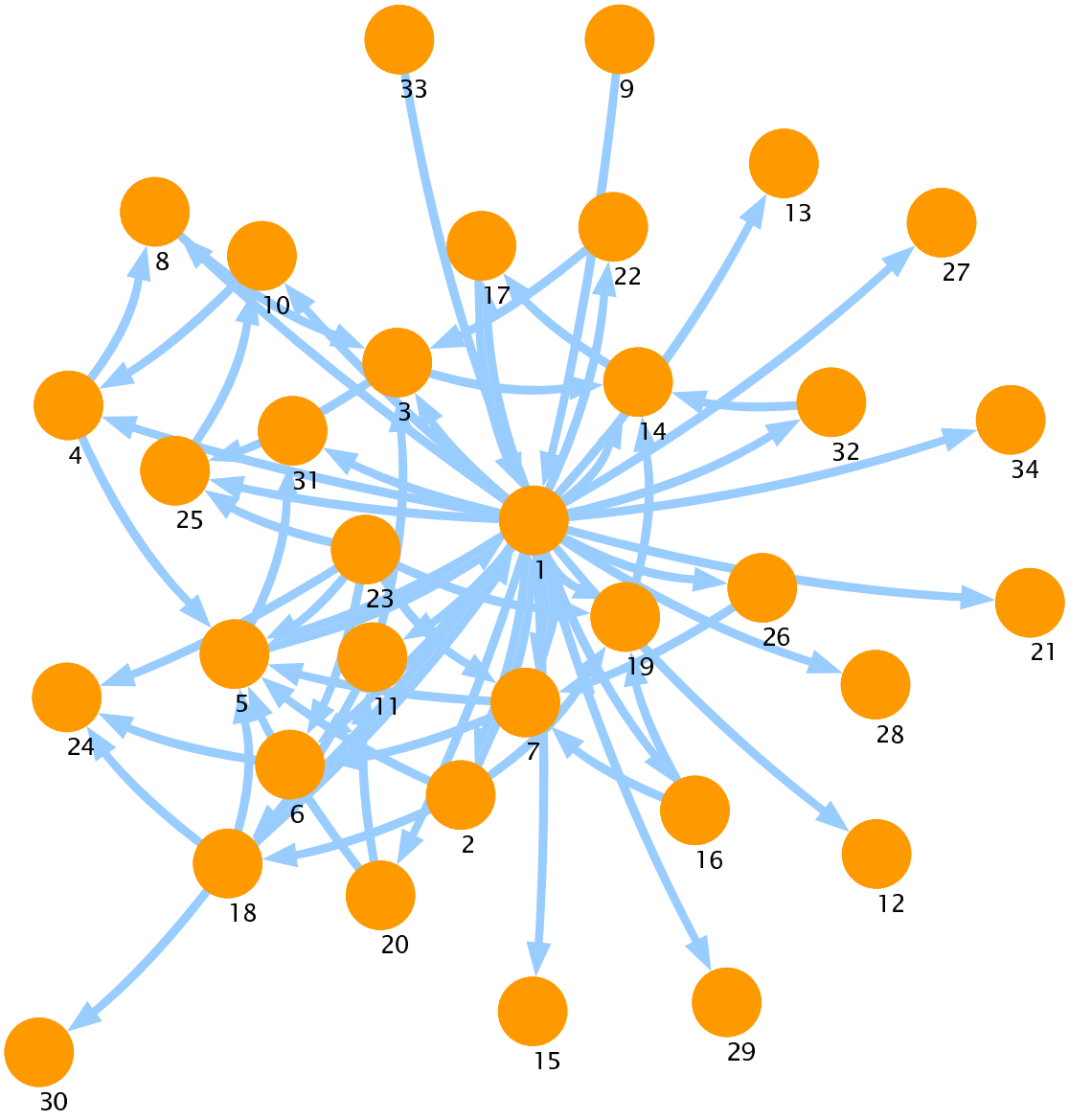


FIGURA 6.2: Red centralizada obtenida en el curso 1 del estudio 1

TABLA 6.2: Valor de cada medida de centralidad para los 3 nodos con mayor valor del estudio 1

Medida	Top 1		Top 2		Top 3	
Degree	1	166	3	39	5	35
Indegree	3	36	5	33	6	17
Outdegree	1	157	2	6	23	6
Betweenness	1	505	17	151	14	142
Authority	3	0.93	5	0.76	6	0.41
Hub	1	1.41	23	0.03	2	0.03

Los resultados en los otros 2 cursos de la misma asignatura, los "cursos 2 y 3", son muy similares, no aportando nada nuevo al análisis de redes sociales. Con estos análisis, el profesor podría tener una mejor visión de cómo se desarrolla el proceso de enseñaza/aprendizaje en el foro, cuyo uso por parte de los estudiantes parece estar casi exclusivamente enfocado en obtener respuestas por parte del profesor a dudas sobre los contenidos de la asignatura. A su vez, los 3 estudiantes con un mayor valor de *Authority* y *Hub* son los que obtuvieron una mejor calificación en el curso, por lo que ya con este análisis de redes sociales se comienza a ver que puede existir una relación entre la influencia de los estudiantes en el foro y su rendimiento, algo de lo que también se podría informar al profesor para ayudarle en el proceso de toma de decisiones sobre la estructura del propio curso y sobre cómo implementar actividades colaborativas.

Aplicando el algoritmo de selección de atributos *CfsSubSetEval* sobre ambos conjuntos, siguiendo un proceso de validación cruzada, en ambos casos el atributo más relevante es el "top 3". No obstante, al aplicar la técnica de selección *ClassifierSubSetEval*, utilizando como algoritmo base J48, aparecen no sólo el "top 3", sino también el *Betweeness* y el *Authority* como atributos predictivos relevantes. En cambio, si aplicamos la misma técnica tomando como base el algoritmo NaïveBayes, las medidas de centralidad destacadas varían, siendo estas el *Degree*, el *Authority* y el *Hub*. Lo mismo ocurre al utilizar otros algoritmos de clasificación como base. Se puede concluir, por tanto, que dependiendo del algoritmo de clasificación a utilizar, las medidas más relevantes para la tarea de predicción varían y, en todo caso, la conjunción de todas ellas (atributo "top 3") ofrece un rendimiento superior, en media, que cada una de las medidas por separado.

La Tabla 6.3 contiene el *accuracy* obtenido dos de los conjuntos de datos con los 6 clasificadores usados en el estudio. Como puede observarse, las medidas de SNA muestran tener un potencial predictivo razonablemente alto por sí solas, obteniendo todos los modelos, salvo uno, un *accuracy* con un rango de entre un 70 % y un 75 %.

Con objeto de constatar hasta que punto las medidas de centralidad no sólo son útiles por sí mismas, sino que pueden ayudar a mejorar la predicción del rendimiento y abandono de los estudiantes cuando sólo se utilizan las medidas de actividad, en la Tabla 6.4 se muestra el *accuracy* obtenido al aplicar los mismos clasificadores que en el caso anterior al conjunto de datos "mixed.dat". Como puede observarse, añadir las medidas

Tabla 6.3: Accuracy obtenido por cada dataset y clasificador, utilizando sólo como predictoras las medidas SNA en el estudio 1

Clasificador	Conjunto performance.dat	Conjunto dropout.dat
J48	71.10	71.00
RandomForest	71.90	73.10
NaïveBayes	70.32	71.00
BayesNetwork	71.10	74.61
JRip	70.10	73.58
Ridor	68.00	74.01

Tabla 6.4: Accuracy obtenido con el dataset "mixed.dat" utilizando sólo medidas de actividad ("No SNA") y añadiendo las medidas de centralidad ("SNA") en el estudio 1

Clasificador	No SNA	SNA
J48	77.20	79.79
RandomForest	78.75	80.31
NaïveBayes	65.29	65.29
BayesNetwork	80.83	81.87
JRip	83.42	77.20
Ridor	78.23	79.79

de centralidad ha supuesto una mejora en el *accuracy* de 4 de los 6 clasificadores obtenidos. Más aún, en estos 4 casos, esta mejora está por encima del 1 %, e incluso con el algoritmo J48 se ha conseguido mejorar en más de un 2 %. Solamente en 1 de los casos, aplicando el algoritmo JRip, se obtiene un rendimiento inferior al añadir las medidas de SNA. Por último, NaïveBayes obtiene el mismo *accuracy* al añadirse estas medidas al modelo, no experimentando ninguna mejora ni empeoramiento.

En la Figura 6.3 se muestra un trozo del árbol generado con el algoritmo J48 al utilizar las medidas de actividad junto con las medidas de centralidad. Como puede observarse, en este ejemplo la mejora en *accuracy* ha venido propiciada, fundamentalmente, por la inclusión en el modelo de dos medidas de centralidad: *Indegree* y *Authority*, denotando la relevancia que este tipo de atributos pueden llegar a tener para predecir el rendimiento de los estudiantes.

6.1.5.2. Estudio 2. Redes sociales distribuidas en los foros de cursos *e-learning*

Al contrario de las redes centralizadas, las redes distribuidas se caracterizan por no tener un único nodo central que aglutina la mayor parte de las interacciones que se producen

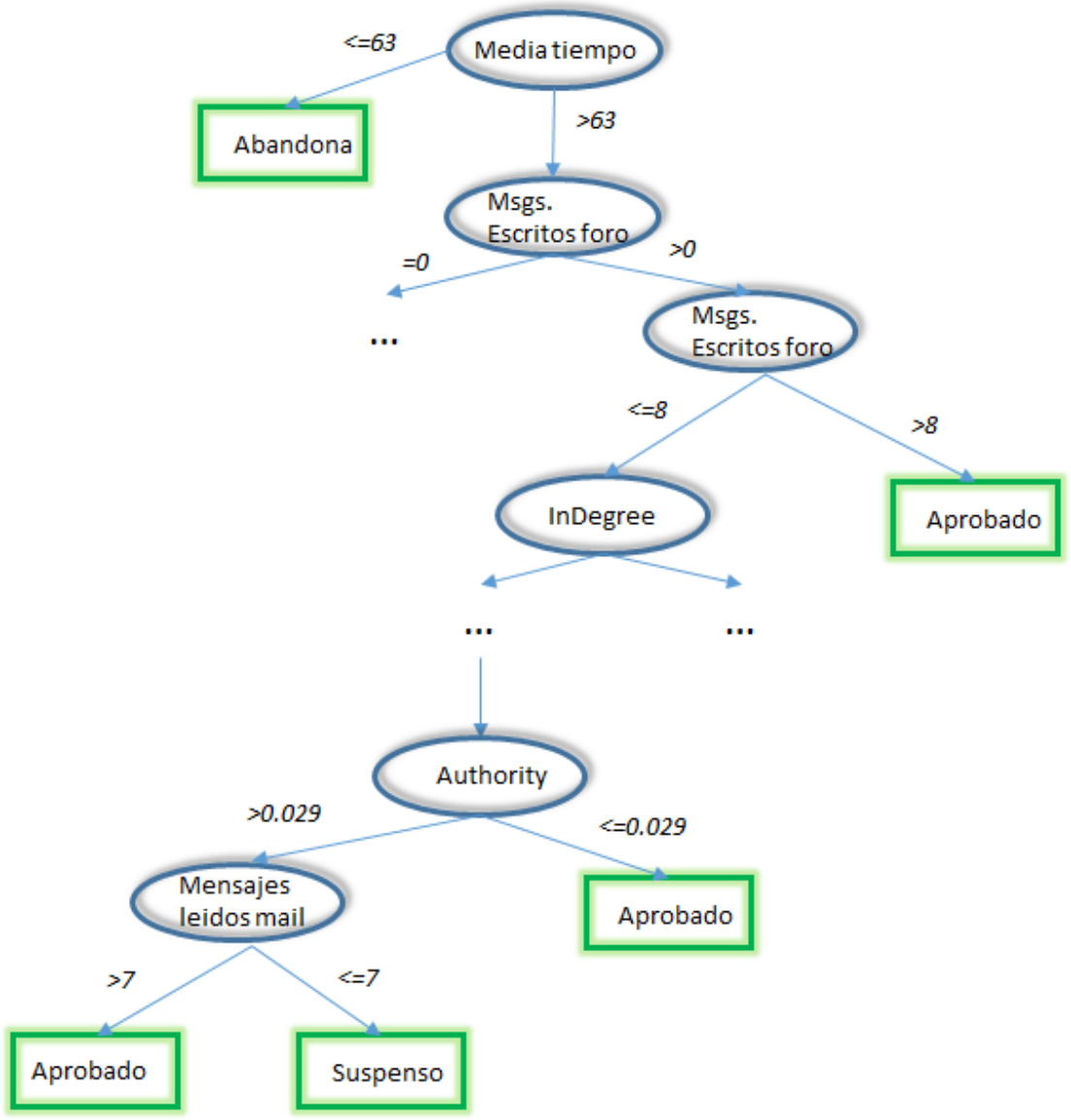


FIGURA 6.3: Árbol de decisión J48 con el conjunto "mixed.dat", utilizando medidas SNA en el estudio 1

en la red. En estas redes existen multitud de nodos que, a su vez, tienen multitud de conexiones con los otros nodos.

En esta sección, se muestran los resultados de un segundo estudio, similar al mostrado en la anterior sección 6.1.5.1, en el que se utilizaron foros de cursos en dónde la interacción no está fuertemente centralizada en un sólo individuo.

Configuración y proceso seguido

En el desarrollo de este estudio se siguió el siguiente proceso:

1. Selección de cursos con interacción fuertemente centralizada

En este estudio se utilizó la interacción en los foros de dos nuevos cursos, que fueron utilizados expresamente para el presente estudio de predicción del rendimiento con medidas SNA. Uno de estos cursos tenía 119 estudiantes matriculados, era de carácter específico y su impartición fue semipresencial. El otro de los cursos tenía 48 estudiantes matriculados y, al contrario que el primero, era de carácter transversal y fue impartido de forma completamente virtual.

En ambos, la interacción de los estudiantes en los foros respondía a un paradigma descentralizado o distribuido, en el que los estudiantes se comunicaban notablemente entre sí, en vez de comunicarse casi únicamente a través de un usuario principal, como ocurría en los cursos del estudio anterior. Por tanto, en este estudio, si bien el profesor tenía cierto protagonismo en los foros, los estudiantes también interaccionaban de forma notable entre sí para resolver dudas acerca de los contenidos o de la organización del curso.

2. Construcción de la red social, extracción y análisis de las medidas de centralidad

Para cada uno de los 2 cursos, se construyó la red social en base a la interacción en los foros, se realizó un análisis de las características de la red y, finalmente, se extrajeron y analizaron las medidas de centralidad, según se indica en la sección 6.1.4, que definen la interacción social de cada uno de los estudiantes.

3. Generación de conjuntos de datos con la actividad y rendimiento de los estudiantes

En este estudio se generaron y utilizaron 2 conjuntos de datos diferentes, todos ellos conteniendo la información relativa a los estudiantes de los 2 cursos mencionados en el paso 1:

- **dataset1**: contiene como atributos predictores, para cada estudiante, los valores de las medidas de centralidad extraídas en el apartado 2 y la actividad del curso de carácter específico. Como clase a predecir se tiene el rendimiento obtenido en el curso (suspende/aprueba).
- **dataset2**: misma información que el dataset2, pero con los estudiantes del curso de carácter transversal.

4. Evaluación de la potencialidad predictora de las medidas SNA

Utilizando únicamente los atributos SNA de los conjuntos de datos dataset1 y dataset2,

se aplicaron varios algoritmos de selección de características, con objeto de determinar la utilidad de las medidas de centralidad como predictoras del rendimiento de los estudiantes, mostrando los resultados de las técnicas *CfsSubSetEval* y *ClassifierSubSetEval* con J48 y NaïveBayes como algoritmos de clasificación base.

5. Aplicación y evaluación de clasificadores

Se utilizaron 2 clasificadores de diferente paradigma, J48 y NaïveBayes, sobre los 2 conjuntos de datos, para determinar si era posible obtener modelos de predicción del rendimiento o abandono de los estudiantes fiables utilizando únicamente medidas SNA como atributos predictores. La comparativa de rendimiento de los modelos de predicción obtenidos se realizó en base al *accuracy*, *TPrate* (% suspensos bien clasificados) y *TNrate* (% aprobados bien clasificados).

Resultados de la experimentación

En las Figuras 6.4 y 6.5 se muestran las redes obtenidas para ambos cursos respectivamente. Si bien existen nodos que, sólo con observar la red, muestran tener un gran número de conexiones en comparación con el resto de nodos, en ambas redes se ve claramente que existe una mayor interacción entre todos los nodos que la que existía en la red centralizada mostrada en el apartado 6.1.5.1. Además, ya no es un único nodo en el que, en ambos casos, monopoliza las interacciones, sino que hay varios que muestran un alto número de éstas.

La Tabla 6.5 muestra que 5 nodos tienen un mayor valor en cada una de las medidas de centralidad utilizadas en este estudio en el dataset2: *Degree*, *Indegree* ("Indeg."), *Outdegree* ("Outdeg."), *Eigenvector* ("Eigen."), *Closeness* ("Close."), *Information centrality* ("Info."), *Betweeness* ("Betw."), *Hub* y *Authority* ("Auth.") . Como puede observarse, el nodo 3254 lidera el ranking en casi todos los casos. Este nodo, al igual que ocurría en la red centralizada del anterior estudio, se corresponde con el profesor principal del curso, que suele ser el usuario que más hilos inicia en el foro, y el que más veces responde a los hilos de los estudiantes. No obstante, en este caso también se identifican otros nodos relevantes. Es el caso del nodo 5036 que, sin llegar a tener una actividad tan alta como el nodo 3254, representa a un usuario que abre hilos y los responde habitualmente (altos valores de *Indegree* y *Outdegree*), actúa como intermediario del resto de estudiantes (alto *Betweeness*) y parece tener un alto perfil tanto de *Authority* como de *Hub*. Por otra parte, el nodo 4046 también tiene un comportamiento bastante interesante, y que combina

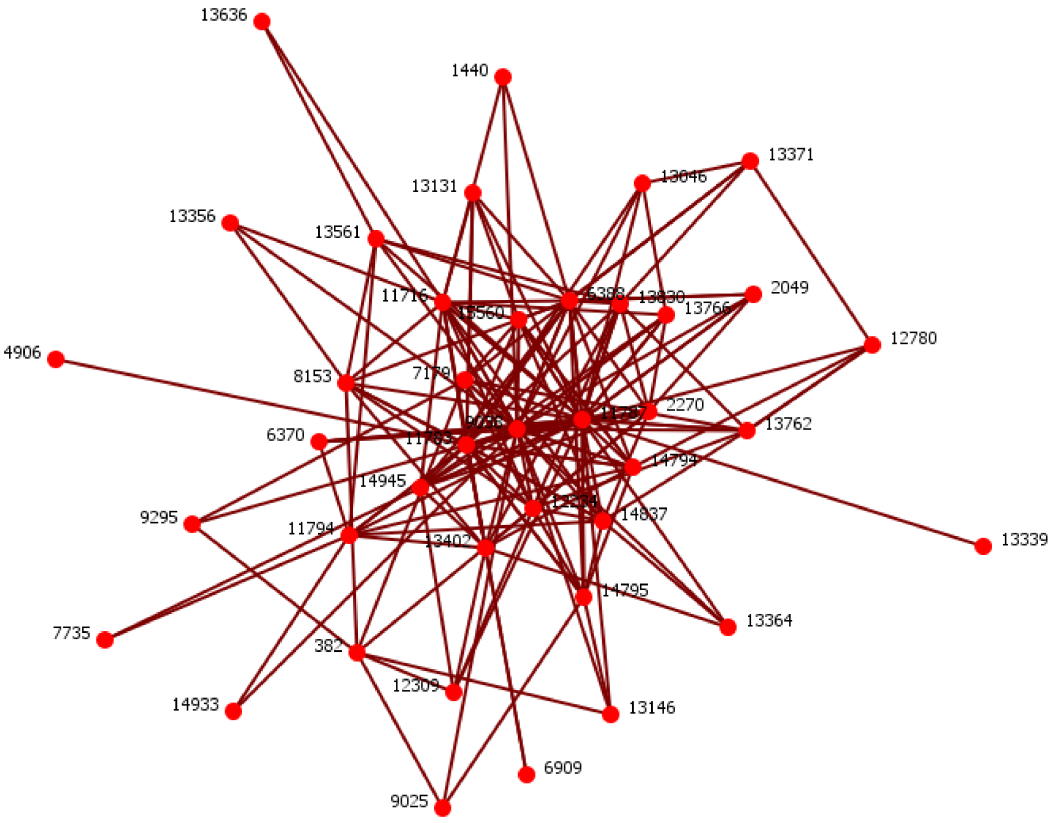


FIGURA 6.4: Red social del foro para el dataset1 del estudio 2

TABLA 6.5: Top 5 de los estudiantes con los valores más altos en cada medida de centralidad en el estudio 2

Degree	Indeg.	Outdeg.	Eigen.	Close.	Info.	Betw.	Hub	Auth.
3254	3254	3254	3254	3254	3254	3254	12722	3254
5036	4046	5036	12722	5748	5036	4046	6837	6826
4046	5036	12722	6837	5036	12722	5036	5036	5036
12722	6837	4046	5036	4046	4046	6685	5077	6821
6837	5630	6837	5077	6821	6837	5630	6821	6837

la actividad típica de los profesores y de los estudiantes: tiene un alto *Degree*, pero su reputación no es demasiado alta (bajos *Eigenvector*, *Hub* y *Authority*). Este nodo podría estar representando o bien a un co-profesor poco activo en comparación con el resto de profesores, o a un estudiante muy colaborativo en el foro.

El nodo 12722, que representa a un estudiante con una alta actividad en el foro, también

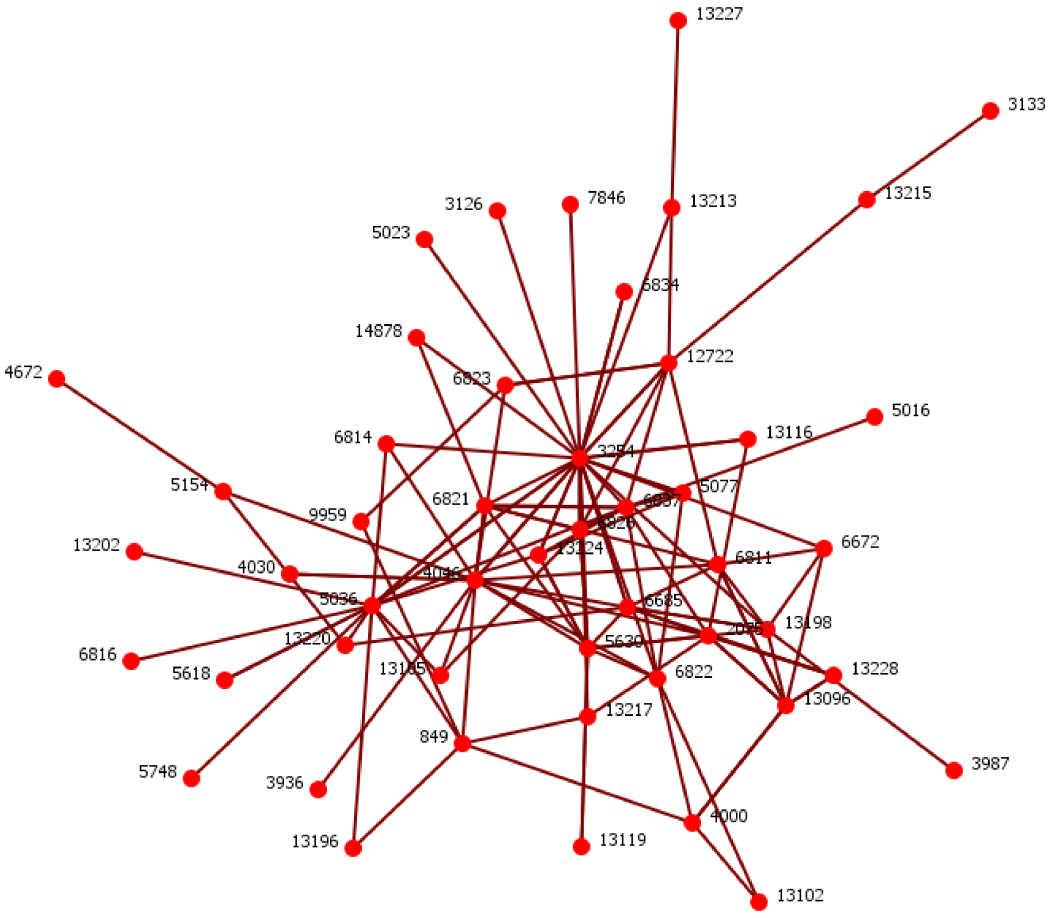


FIGURA 6.5: Red social del foro para el dataset2 del estudio 2

destaca en su comportamiento social, respondiendo a una gran cantidad de preguntas (alto *Outdegree*) en el foro. Por otra parte, no parece que la actitud de este estudiante sea la de realizar preguntas en el foro, ya que no aparece entre los "top 5" en relación a su *Indegree*. Este nodo, por tanto, responde a un perfil en el que el estudiante dedica su actividad social a resolver dudas o responder a otros estudiantes. Por otro lado, el nodo 6837 representa a un estudiante cuyo comportamiento es diametralmente opuesto al del nodo 12722, con una mayor tendencia a recibir respuestas (alto *Indegree*) que a responderlas.

Las medidas de centralidad, tal y como se extrae de las conclusiones anteriores, muestran ser útiles de cara a identificar perfiles de usuario en base a su interacción social con el resto. Esta información puede serle de utilidad al profesor de un curso *e-learning*, por

ejemplo, para detectar aquellos estudiantes que colaboran en el proceso de enseñanza y aprendizaje del curso involucrándose de forma activa en la resolución de dudas y problemas de otros de sus compañeros, o bien identificar aquellos estudiantes que, si bien utilizan el foro para pedir ayuda, no suelen prestarla cuando son otros compañeros los que la piden.

En la Tabla 6.6 se muestran cuáles son los atributos más relevantes para realizar la tarea de predicción en el dataset1. Se incluyen aquellos atributos cuya relevancia es igual o superior a 2 sobre 10, esto es, los atributos que han sido seleccionado en al menos 2 de las 10 iteraciones del proceso de validación cruzada. Como puede verse, sólo 2 atributos son devueltos en el proceso de selección, y ambos son atributos de centralidad: *Degree* y *Hub*. Como primera conclusión, a falta de observar los resultados obtenidos con el resto de técnicas y con el otro curso, podemos ya concluir el gran poder predictivo que pueden tener las medidas de SNA. El *Degree*, esto es, la suma de mensajes escritos y leídos en el foro, parece tener además una especial relevancia. Esto da una idea de que, para predecir el rendimiento de los estudiantes, su actividad en el foro tiene una notable importancia. El otro atributo seleccionado es la medida de centralidad *Hub*, lo que denota que no solamente la actividad "per se" es suficientemente importante para la tarea de predicción, sino que además aquellos estudiantes que reciben más respuestas a sus preguntas son aquellos que finalmente más aprenden y, por tanto, más posibilidades tienen de aprobar el curso. De esto se puede extraer, a su vez, que el foro, y la interacción social que allí se realiza, tiene especial importancia en el rendimiento de los estudiantes, y que la resolución de dudas por parte del profesor, o de otros estudiantes, tiene un gran peso en el rendimiento final del estudiante.

TABLA 6.6: Atributos seleccionados con la técnica CfsSubSetEval para el "curso 4"del estudio 2

Atributo	Relevancia (0-10)
Degree	7
Hub	3

Los resultados de aplicar la técnica de selección *ClassifierSubSetEval*, mostrados en las Tablas 6.7 y 6.8, arrojan conclusiones similares al respecto de la importancia de las métricas SNA para predecir. No obstante, otros atributos de actividad al margen de las métricas SNA adquieren especial relevancia, según el algoritmo de clasificación utilizado como base. En ambos casos, tanto para J48 como para NaïveBayes, el *Hub* muestra ser la

medida con más relevancia para predecir el rendimiento de los estudiantes. Sin embargo, el *Degree*, que con *CfsSubSetEval* era el atributo más relevante, no aperece en ninguno de los casos. En su lugar, cobran importancia otras medidas SNA, como son el *Authority* o el *Information centrality* con NaïveBayes, o el *Betweeness* y el *Click Membership* con J48. No obstante, el número de veces que el estudiante ha accedido a los recursos, una medida no SNA, resulta ser la segunda más relevante para NaïveBayes, y una de las destacadas para J48. Para este último algoritmo, además, destacan notablemente otras medidas de actividad en el foro al margen de las de centralidad, como son el número de hilos iniciados y leídos. Con estos resultados, se alcanzan las mismas conclusiones que con la red social centralizada del apartado 6.1.5.1, y es que las medidas de centralidad tienen un gran peso de cara a predecir el rendimiento de los estudiantes, y parecen mostrar un gran potencial como complemento a las otras medidas de actividad. Más aún, dependiendo del clasificador a utilizar, las medidas SNA más relevantes pueden variar, si bien aquellas que un alto peso, como es el caso del *Hub* para este curso, mantienen una alta relevancia para todos los algoritmos.

Tabla 6.7: Atributos seleccionados con la técnica ClassifierSubSetEval para el dataset1, utilizando NaïveBayes como clasificador base en el estudio 2

Atributos seleccionados	Relevancia (0-10)
Hub	10
Authority	3
InformationCentrality	6
ClickMembership	4
DegreeClustering	3
N_Initiated_discussions	3
N_subscriptions_forum	2
N_views_resources	9

Tabla 6.8: Atributos seleccionados con la técnica ClassifierSubSetEval para el dataset1, utilizando J48 como clasificador base en el estudio 2

Atributos seleccionados	Relevancia (0-10)
OutDegree	2
Hub	5
ClickMembership	3
Closeness Inverted	2
Betweeness	3
ClusteringDegree	2
N_Initiated_discussions	4
N_readen_discussions	3
N_views_forum	2
N_views_resources	2

En la Tabla 6.13 se muestran el *accuracy*, *TPrate* (porcentaje de aprobados bien clasificados) y *TNrate* (porcentaje de suspensos bien clasificados) obtenidos al utilizar los algoritmos de clasificación J48 y Naïve Bayes, tanto al incluir las métricas SNA (columna "SNA"), como utilizando únicamente los otros 32 atributos de actividad de los estudiantes en solitario (columna "No SNA"). En la última columna, "Improv.", se muestra el porcentaje de mejora obtenido al incluir los atributos SNA en el modelo predictivo. Como puede verse, al utilizar J48, la mejora en *accuracy* es notablemente alta, alcanzando casi el 14 %. Igualmente, el rendimiento del modelo tanto en términos de *TPrate* como de *TNrate* es notablemente alto, con una mejora del 11.5 % y 17.7 % respectivamente. En el caso de NaïveBayes, el *accuracy* también se ve mejorado de forma notable, en más de un 2 %, fruto de haber mejorado la clasificación de los estudiantes aprobados en un 3.9 % (*TPrate*). Estos resultados refuerzan las conclusiones extraídas en el proceso de selección de atributos, y es que las medidas de centralidad tienen un peso importante a la hora de determinar el rendimiento de los estudiantes. De hecho, el peso de estas medidas es más evidente en un foro cuya red social no es centralizada. Esto se debe fundamentalmente a que, al contrario que en una red centralizada, en una red distribuida existe una mayor implicación de los estudiantes en la interacción, con lo que consecuentemente cobra mayor relevancia.

Tabla 6.9: Accuracy, TPrate y TNrate obtenidos con J48 y NaïveBayes en el dataset1 del estudio 2

		SNA	No SNA	Improv.
J48	Acc.	76.74 %	62.79 %	13.95 %
	TPr.	76.9 %	65.4 %	11.5 %
	TNr.	76.5 %	58.8 %	17.7 %
NaïveBayes	Acc.	51.16 %	48.84 %	2.32 %
	TPr.	57.7 %	53.8 %	3.9 %
	TNr.	41.2 %	41.2 %	0.0 %

En la Figura 6.6 se muestra un trozo del árbol de clasificación obtenido con J48 para el dataset1. Como puede observarse, el atributo dominante es el *Hub*, que se encuentra en la raíz del árbol, confirmando así las conclusiones sobre la potencialidad predictora de las medidas de centralidad SNA.

En el dataset2, sin embargo, las medidas de centralidad ven mermado su poder predictor, si bien siguen mostrándose útiles para la tarea, tal y como se muestra en las Tablas 6.10, 6.11 y 6.12. Como puede observarse, el *Degree* es una de las medidas más relevantes para la predicción, al igual que ocurría con el dataset2. Igualmente, el *Hub*, el *Authority*,

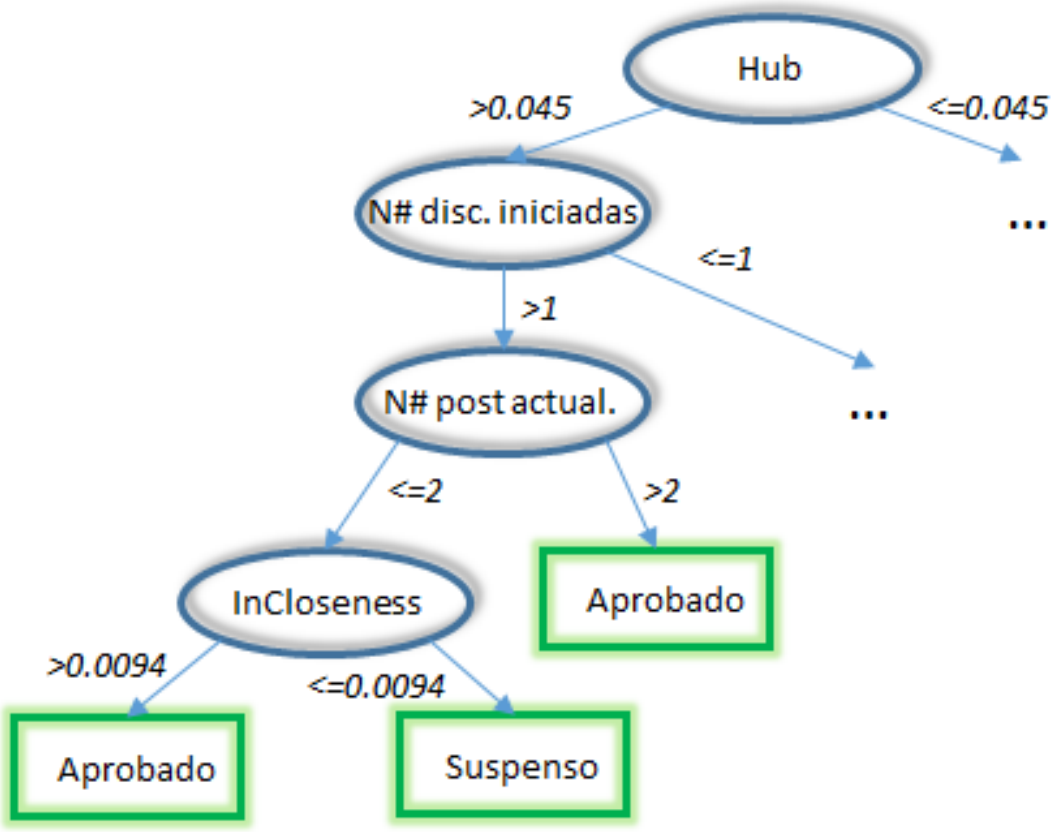


FIGURA 6.6: Árbol de decisión J48 para el dataset1, utilizando medidas SNA en el estudio 2

el *Eigenvector* y el *Betweeness* tienen un moderado peso dependiendo del algoritmo de clasificación utilizado. No obstante, en este curso existen otras medidas de actividad al margen de las SNA que tiene notablemente más relevancia, y entre ellas destaca el número de veces que el estudiante ha realizado los auto-tests del curso. Este hecho, tiene que ver con la propia estructura del curso, en el cual existe una gran actividad en los auto-test, denotando que en el proceso de aprendizaje de los estudiantes tienen mucho más peso que el hecho de poder resolver dudas en el foro.

TABLA 6.10: Atributos seleccionados con la técnica CfsSubSetEval para el dataset2 del estudio 2

Atributos	Relevancia (0-10)
Degree	4
N_attempts_quizzes	6

No obstante, tal y como puede verse en la Tabla 6.13, la inclusión de medidas SNA supone una mejora más que notable de cara a generar el modelo predictivo con J48, con una mejora en *accuracy* de más del 6 %, y del 5.8 % y 6.9 % en *TPrate* y *TNrate*

Tabla 6.11: Atributos seleccionados con la técnica ClassifierSubSetEval para el dataset2, utilizando NaïveBayes como clasificador base en el estudio 2

Atributos	Relevancia (0-10)
Eigenvector	3
Hub	2
Authority	4
ClusteringDegree	3
N_readen_discussions	8
N_view_resources	8
N_attempt_quizzes	9

Tabla 6.12: Atributos seleccionados con la técnica ClassifierSubSetEval para el dataset2, utilizando J48 como clasificador base en el estudio 2

Atributos	Relevancia (0-10)
OutDegree	2
EigenVector	2
Betweeness	3
Authority	2
ClusteringDegree	2
N_view_resources	7
N_read_discussions	7
N_attempt_quizzes	8
N_actions	2

respectivamente. Por el contrario, NaïveBayes ve mermado su rendimiento al utilizar medidas SNA, aunque muy ligeramente, empeorando por debajo del 1 % en *accuracy* y en un 1.1 % en *TPrate*. De ello, se puede concluir que, si bien las medidas de SNA muestran un gran potencial para la predicción del rendimiento de los estudiantes cuando el foro es utilizado como una de las principales herramientas de aprendizaje, este potencial disminuye notablemente en cursos en los que, como sucede en el dataset2, existen otras herramientas, como los auto-test, a las que los estudiantes dan más prioridad en su proceso de aprendizaje que al foro. No obstante, aún en ese caso, las medidas SNA siguen mostrando poder ser útiles, lo que se corrobora observando la mejora obtenida con J48.

En la Figura 6.7 se muestra un trozo del árbol obtenido con J48, en donde se ve que tanto el *Authority* como el *Betweeness* aparecen en el tercer nivel del árbol, y el *Clustering Degree* en el cuarto.

TABLA 6.13: Accuracy, TPrate y TNrate obtenidos con J48 y NaïveBayes en el dataset2 del estudio 2

		SNA	No SNA	Improv.
J48	Acc.	76.52 %	70.43 %	6.09 %
	TPr.	86.0 %	80.2 %	5.8 %
	TNr.	48.3 %	41.4 %	6.9 %
NB	Acc.	64.34 %	65.21 %	-0.87 %
	TPr.	82.6 %	83.7 %	-1.1 %
	TNr.	10.3 %	10.3 %	0.0 %

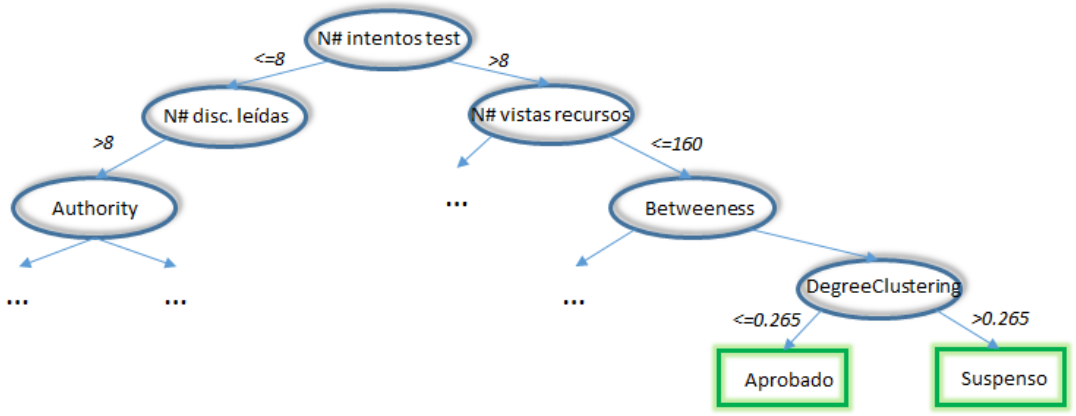


FIGURA 6.7: Árbol de decisión J48 para el dataset2, utilizando medidas SNA en el estudio 2

6.1.6. Conclusiones y trabajo futuro

El análisis de las interacciones sociales con SNA en los foros de cursos virtuales muestra un gran potencial, pudiendo analizar el comportamiento social de los estudiantes en base a los mensajes que envían y reciben, tanto entre ellos como con respecto al profesor. En los cursos en los que los foros se modelan como una red centralizada, las medidas de centralidad muestran ser moderadamente útiles para predecir el rendimiento de los estudiantes. No obstante, es en los cursos en los que se obtienen redes sociales distribuidas en los que estas medidas cobran más relevancia. Aunque por sí solas, las medidas de centralidad pueden obtener modelos de predicción accionables, cuando realmente muestran su potencial es al usarlas en combinación con otras medidas de actividad de los estudiantes, mejorando los modelos de predicción con respecto a los obtenidos si no utilizamos medidas SNA.

En suma, el comportamiento social de los estudiantes está notablemente ligado a su rendimiento en los cursos, si bien el hecho de que haya habido pocos conjuntos de datos disponibles en el desarrollo de esta tesis para este estudio hace que sea necesario en

un futuro más estudios que corroboren esta afirmación. Estos estudios, como trabajo futuro, deberán ir enfocados no solamente a obtener medidas SNA en foros, si no en otras herramientas de los LCMS en lo que exista interacción social, como son las wikis o los blogs. También como trabajo futuro, se valorará la posibilidad de extender el análisis SNA a entornos de aprendizaje diferentes a los ofrecidos por los LCMS tradicionales y en los que existe un mayor número de usuarios, como son los MOOCs.

6.2. Eliminando redundancia para facilitar la visualización de las reglas de asociación

Diversos estudios han mostrado la utilidad de las técnicas de reglas de asociación en el campo educativo con objetivos variados: extraer patrones de interés sobre el comportamiento de los estudiantes para dar una mejor idea al profesor de como interaccionan con su curso [79], encontrar errores que los estudiantes cometen frecuentemente de forma conjunta [80], modelar el rendimiento de los estudiantes en base a su actividad [81], extraer patrones denominados "raros"(poco frecuentes) [132] utilizando técnicas como Apriori-rare [285], o incluso optimizar la organización del curso en base a los contenidos que más interesan [286]

En E-learning WebMiner ya existe un servicio, mostrado en el anexo A.1, que ofrece al profesor patrones de comportamiento de los estudiantes en formas de reglas de asociación. No obstante, el principal problema del modelo de reglas retornado por ElWM es que, al usar un algoritmo tan común en la literatura como es Apriori [287], ocurre que para la mayor parte de conjuntos de datos se genera una cantidad ingente de reglas, que en el mejor de los casos pueden ser de cientos y, en el peor, de miles. Evidentemente, es difícil incluso para un experto en minería de datos el poder extraer, con facilidad, información útil de entre estas reglas, por lo que este inconveniente es aún más acuciado por los usuarios no expertos.

El principal problema de algoritmos como Apriori es que generan una alta cantidad de reglas redundantes, esto es, que contienen una información casi idéntica entre ellas. En la Tabla 6.14 se expone un ejemplo de este problema de redundancia, en el que las Regla 3 "resume"la información contenida en las Reglas 1 y 2, teniendo las tres una confianza similar. En este ejemplo concreto, podría bastar con mostrar al usuario, por tanto, la

Regla 3, eliminando del conjunto las Reglas 1 y 2, que no aportan ninguna información nueva.

TABLA 6.14: Ejemplo de rendundancia entre reglas

Regla 1	A ⇒ B	Confianza : 91.6 %
Regla 2	A ⇒ C	Confianza : 91.3 %
Regla 3	A ⇒ B & C	Confianza : 90.9 %

Otro de los problemas que puede encontrarse un usuario no experto es la complejidad de la configuración inicial de algunos de los algoritmos de reglas de asociación, con numerosos parámetros de entrada. En [288] proponen una solución a este problema mediante el uso de Predictive Apriori [289], ya que este algoritmo sólo requiere de un parámetro de configuración para funcionar. No obstante, este algoritmo no soluciona el problema de la redundancia.

También la interpretabilidad de los modelos está ligada a las medidas de calidad o rendiminento de las reglas. En [290], por ejemplo, los autores argumentan que medidas como el *lift* son bastante útiles para extraer reglas en el campo educativo. No obstante, este tipo de medidas son difícilmente interpretables, sobre todo por un usuario no experto, por lo que dificultan la toma de decisiones por parte de éste. Si queremos que un profesor de cursos virtuales puede interpretar la reglas que se le muestran, se han de utilizar medidas más simples como la confianza y el soporte.

En el estudio realizado como parte de esta tesis, con la motivación de mejorar los resultados ofrecidos por ElWM, se compararon los resultados obtenidos por 5 algoritmos de reglas de asociación: la implementación de Apriori en Weka, Predictive Apriori, Apriori de Borglet [291], ChARM [292] y Yacaree [33]. La implementación de Apriori de Borglet, a diferencia de la de Weka, genera reglas en las que el consecuente tiene solamente un elemento. Además, difiere en la forma de generar los conjuntos frecuentes y, posteriormente, las reglas, siendo un algoritmo más eficiente en términos de rendimiento temporal. Tanto Apriori de Borglet como la implementación de Weka, así como Predictive Apriori, utilizan conjuntos frecuentes para construir las reglas.

Los otros dos algoritmos, ChARM y Yacaree, utilizan conjuntos cerrados. En el caso de Yacaree, además, este algoritmo trata de eliminar las reglas redundantes, filtrando aquellas que son rendundantes entre sí. Este proceso se hace mediante la comparación

de la confianza de las reglas: si dos reglas tienen una información similar, y la diferencia de confianza es baja (ver *closure-based confidence boost* [293]) la regla menos informativa es eliminada, quedándonos solamente con la otra regla.

6.2.1. Comparativa de algoritmos de reglas de asociación

Para la comparativa de los 5 algoritmos de reglas, se utilizaron 5 conjuntos de datos provenientes de 2 cursos *e-learning* con una estructura diferente, y alojados en la plataforma Blackboard de la Universidad de Cantabria. El primero de estos cursos, que en adelante será denominado "curso1", tiene un diseño basado en páginas Web con contenidos, en las que además incluye video-tutoriales, animaciones flash y otros elementos interactivos. Los estudiantes de este curso habían de realizar y entregar 4 ejercicios evaluables, además de 2 proyectos, y realizar un examen final. El perfil de los estudiantes era heterogéneo, estando este curso abierto a toda la comunidad universitaria independientemente de la carrera que se encontrasen estudiando. El número total de alumnos era de 79. Al contrario, el segundo curso, al que llamaremos "curso2", sólo admitía estudiantes que se encontrasen matriculados en el grado de telecomunicaciones de la UC, siendo el número total de alumnos de 24. Todos los materiales de este curso estaban disponibles desde el primer día en formato *.pdf*, habiendo un total de 11 documentos, 10 de ellos correspondientes a cada uno de los 10 temas de la asignatura, y un último documento que contenía todos los apuntes de una edición anterior del mismo curso.

Las características de los 5 conjuntos de datos se muestran en la Tabla 6.15, en dónde se pueden observar el número de instancias e items de cada conjunto. En el caso de los items, se consideran los diferentes valores que pueden encontrarse en una instancia.

Tabla 6.15: Descripción de los conjuntos de datos

Name	Transactions	Items
Dataset1 (materiales y tests del curso 1)	407	22
Dataset2 (recursos del curso 1)	2486	27
Dataset3 (recursos sin "Organizer" del curso 1)	2346	26
Dataset4 (recursos del curso 2)	5892	27
Dataset5 (recursos sin "Organizer" del curso 2)	5643	26

Tabla 6.16: Ejemplo de instancias en el dataset2

Organizer;Forum;Tests;Assigments
Organizer;Resources;Forum
Organizer;Resources
Organizer;Forum
Forum;Resources;Mail
...

El dataset1, generado con respecto al curso1, contiene la información acerca de los ficheros *.pdf* que fueron accedidos por un estudiante en una sesión de conexión al curso concreta, esto es, qué ficheros *.pdf* visitó en un momento dado, desde el momento en que se conectó hasta el momento en el que se desconectó del curso. A esta información se añadió también la relativa a los accesos que el estudiante realizó, en la misma sesión, a los auto-test del curso.

Los dataset2 y dataset4 corresponden a las herramientas visitadas en una sesión por un estudiante en los curso1 y curso2 respectivamente. Ejemplos de estas herramientas, ofrecidas por Blackboard, son el foro, el correo electrónico interno, las páginas de contenidos o las wikis del curso. Como ejemplo, en la Tabla 6.16 se incluye un fragmento del dataset2. Algo que se muestra evidente observando las instancias de este conjunto de datos, incluso antes de extraer las reglas de asociación, es que la herramienta "Organizer" aparece en un gran número de instancias. De hecho, al analizar los datos se constata que el soporte de este elemento es del 84 %. Este alto valor es debido a que la herramienta "Organizer" es través de la cual un estudiante accede al curso cuando inicia sesión, algo que ocurre también en el dataset4 (85 % de soporte). Este hecho puede provocar, y de hecho provoca, que se generen pares de reglas con similar soporte y confianza entre las que, la única diferencia, es el hecho de que la herramienta "Organizer" aparezca o no en ellas. Por ello, se crearon los dataset3 y datset5, que contienen la misma información que los dataset2 y dataset3 pero habiendo previamente eliminado la herramienta "Organizer".

La Tabla 6.17 contiene el número de reglas generadas por cada uno de los 5 algoritmos de reglas de asociación mencionados en el anterior apartado. Una de las principales dificultades en la experimentación fue el establecimiento de los parámetros de configuración de estos algoritmos, destacando el soporte, ya que pequeñas variaciones en el valor de este parámetro podrían favorecer a unos algoritmos por encima de otros. Esta dificultad se ve agravada por el hecho de que uno de los algoritmos, Yacaree, no requiere de

TABLA 6.17: Número de reglas generadas al aplicar cada algoritmo sobre los datasets

Dataset	Número de reglas s=1 % c=66 %				
	Weka Apriori	Predictive Apriori	Borgelt Apriori	ChARM	*yacaree*
Dataset1	2272	1730	524	366	40
Dataset2	7523	over 10000	3751	5610	255
Dataset3	4249	over 10000	1876	2586	93
Dataset4	1442	—	1023	1427	182
Dataset5	488	—	404	469	46

parámetros de entrada, auto-configurando el valor de soporte. Es por ello que, previamente a la obtención de los resultados mostrados en la Tabla 6.17, se realizó una extensa experimentación, en la que se pudo observar que, con valores muy bajos de soporte, determinados algoritmos eran incapaces de arrojar resultados. Por poner dos ejemplos, al utilizar un soporte de 0.02 %, Predictive Apriori requería más de 40 minutos para retornar la reglas, mientras que con Apriori de Weka se producía una sobre-saturación de memoria RAM, incluso cuando se le "cedía"para su ejecución 2GB. Estas observaciones llevaron, finalmente, a la conclusión de utilizar un soporte del 1 % y una confianza del 66 % como valores de entrada.

Con respecto a los conjuntos de datos sin la herramienta "Organizer", en el caso del dataset3 el número de reglas devueltas por Apriori es de 4249, un número demasiado alto como para que puedan ser manejables por el usuario. De entre ellas, las primeras 243 son reglas con una confianza del 100 %, es decir implicaciones, con muy bajo soporte y muy redundantes entre sí, como puede verse en la Tabla 6.18. Las reglas 2 y 3, por ejemplo, pese a tener una confianza del 100 %, tienen un soporte extremadamente bajo, por debajo del 2 %. Otro ejemplo similar lo encontramos con las reglas 235 y 236: en este caso, la única diferencia entre ambas es que en la primera aparece como elemento del antecedente la herramienta "assignments", mientras que en la regla 236 aparece la herramienta "asssessment". Ambas dos tienen la misma confianza, y el mismo bajo soporte, 1 %.

Lo mismo ocurre con las reglas obtenidas por Apriori con una confianza inferior al 100 %. En la Tabla 6.19 se muestran 2 ejemplos de ello. Las reglas 2523 y 2524 difieren entre sí en un elemento del consecuente, teniendo ambas la misma confianza, 78 %, y el mismo bajo soporte, 1.2 %. Algo similar puede observarse entre las reglas 2530 y 2534, en este caso teniendo la regla 2535 un elemento más en el antecedente.

Tabla 6.18: Subconjunto de reglas con confianza de 100 % obtenidas al aplicar Apriori de Weka sobre el dataset3

No.	Association rule	(Sup., Conf.)
2	announcement tracking ⇒ assessment	(1.7, 100)
3	announcement mygrades tracking⇒ assessment	(1.6, 100)
235	assignments calendar contentpage discussion medialibrary syllabus ⇒ assessment	(1.0, 100)
236	assessment calendar contentpage discussion medialibrary syllabus ⇒ assignments	(1.0, 100)

Tabla 6.19: Subconjunto de reglas con confianza menor de 100 % obtenidas al aplicar Apriori de Weka sobre el dataset3

No.	Association rule	(Sup., Conf.)
2523	announcement assessment calendar syllabus ⇒ assignments contentpage	(1.2, 78.0)
2524	announcement assessment calendar syllabus ⇒ assignments discussion	(1.2, 78.0)
2530	announcement calendar mail ⇒ contentpage	(1.0, 78.0)
2534	announcement assignments calendar chat ⇒ contentpage	(1.0, 78.0)

Tabla 6.20: Subconjunto de reglas obtenidas al aplicar PredictiveApriori sobre el dataset3

No.	Association rule	(Support, accuracy)
122	assignments calendar search ⇒ syllabus	(0.85, 0.95439)
123	assignments chat weblinks ⇒ assessment syllabus	(0.85, 0.95439)
124	assignments chat weblinks ⇒ discussion syllabus	(0.85, 0.95439)
125	assignments discussion search ⇒ assessment syllabus	(0.85, 0.95439)

El resultado es el mismo al utilizar el dataset5 del curso 1. Aunque en este caso el número de reglas, 488, es algo más manejable, sigue siendo bastante alto, y el problema de la redundancia persiste. En el caso de utilizar PredictiveApriori para ambos conjuntos de datos, dataset3 y dataset5, el resultado es aún más inmanejable, ya que en el primer caso el número de reglas retornadas es superior a 10.000 y, en el segundo, el algoritmo no terminó de ejecutarse pasados 60 minutos, por lo que fue imposible obtener las reglas. La redundancia entre reglas es también bastante obvia al utilizar este algoritmo, como se muestra en la Tabla 6.20.

En el caso de Apriori de Borglet, el número de reglas también es bastante alto, aunque notablemente menor que con Apriori de Weka: 1876 y 404 para los dataset3 y dataset5, de entre las que 141 y 2 son implicaciones respectivamente. Extraer conclusiones en base a este número de reglas es, por tanto, también bastante difícil y tedioso. En la Tabla 6.21 se muestran dos conjuntos diferentes de reglas en los cuáles se denota claramente

Tabla 6.21: Subconjunto de reglas obtenidas al aplicar Apriori de Borgelt sobre el dataset3

No.	Association rule	(Supp. , Conf.)
11	chat ⇒ discussion	(3.7, 84.9)
12	chat ⇒ assignments	(3.7, 75.6)
13	chat ⇒ assessment	(3.7, 81.4)
99	chat announcement ⇒ discussion	(2.0, 84.8)
100	chat announcement ⇒ assignments	(2.0, 87.0)
101	chat announcement ⇒ assessment	(2.0, 93.5)

Tabla 6.22: Subconjunto de reglas obtenidas al aplicar ChARM sobre el dataset3

No.	Association rule	(Supp. , Conf.)
3	announcement, contentpage, medialibrary, syllabus ⇒ assessment	(1.02, 96.00)
4	announcement, assessment, medialibrary, syllabus ⇒ contentpage	(1.02, 88.89)
5	announcement, assessment, contentpage, medialibrary ⇒ syllabus	(1.02, 70.59)
6	announcement, medialibrary, syllabus ⇒ assessment, contentpage	(1.02, 82.76)
7	announcement, medialibrary, syllabus ⇒ contentpage	(1.07, 86.21)
8	announcement, contentpage, medialibrary ⇒ syllabus	(1.07, 67.57)

Tabla 6.23: Subconjunto de reglas obtenidas al aplicar ChARM sobre el dataset5

No.	Association rule	(Supp. , Conf.)
10	chat, contentpage, discussion ⇒ assessment	(1.13, 81.01)
11	assessment, chat contentpage ⇒ discussion	(1.13, 94.12)
12	chat, contentpage ⇒ assessment, discussion	(1.13, 71.91)
31	contentpage, discussion, syllabus, ⇒ assessment	(1.12, 84.00)
32	assessment, discussion, syllabus, ⇒ contentpage	(1.12, 66.32)
33	assessment, contentpage, syllabus, ⇒ discussion	(1.12, 79.75)

la redundancia existente: con una confianza similar, las reglas con antecedente “chat” están incluidas en aquellas con antencedente “chat” y “annoucement”. Lo mismo que con Apriori de Borglet sucede con Charm, tal y como se puede ver en las Tablas 6.22 y 6.23.

Al utilizar el algoritmo Yacaree, que elimina redundancia, el número de reglas en ambos dataset3 y dataset5 se reduce considerablemente en comparación al resto de algoritmos, con 93 y 46 reglas respectivamente. En la Tabla 6.24 se muestra un subconjunto de las reglas ofrecidas por Yacaree para el dataset3. Puede observarse que el algoritmo devuelve reglas bastante obvias, como la número 1, que indica que cuando en una sesión los estudiantes utilizan la herramienta “filemanager”, han accedido en la misma sesión a la herramienta “assignment”, en dónde se encuentran las tareas evaluables del curso. Esta regla es poco informativa desde el punto de vista del profesor, dado que evidentemente para subir un fichero con las tareas el estudiante tiene que acceder a ellas. Las reglas 6,

Tabla 6.24: Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset3

No.	Association rule	(Supp., Conf., Lift, Cboost)
1	filemanager ⇒ assignments	(4.6, 93.9, 1.908, 1.908)
6	discussion whoisonline ⇒ assessment	(3.0, 75.5, 1.648, 1.379)
18	discussion mail ⇒ assessment	(3.2, 72.1, 1.574, 1.268)
19	announcement mail ⇒ assessment discussion	(1.6, 80.9, 3.381, 1.267)
7	announcement ⇒ assessment	(7.6, 88.1, 1.923, 1.369)
12	calendar ⇒ assessment	(9.1, 75.9, 1.656, 1.337)
36	calendar ⇒ assignments	(8.1, 67.0, 1.362, 1.219)
50	announcement calendar ⇒ assessment assignments	(2.6, 77.2, 2.941, 1.200)
16	tracking ⇒ mygrades	(6.8, 80.3, 2.409, 1.272)
8	contentpage mygrades ⇒ assessment	(3.8, 84.8, 1.850, 1.369)
10	contentpage discussion ⇒ assessment	(7.3, 75.1, 1.639, 1.339)

18 y 19 tampoco ofrecen información relevante al profesor: le están informando de que los estudiantes acceden al foro ("discussion") para consultar acerca de las entregables y sus fechas, algo de lo que el propio profesor es ya plenamente consciente dado que él realiza este tipo de anuncios en el foro.

No obstante, al utilizar Yacaree, es posible encontrar otras reglas que sí son más interesantes y que le dan al profesor información sobre el comportamiento de sus estudiantes, que no conocía previamente. Así, por ejemplo, las reglas 7, 12, 36 y 50 le informan de que existen sesiones en las que el estudiantes se conecta al curso exclusivamente para consultar fechas de entregas y exámenes, mientras que la regla 16 indica que unas pocas sesiones son utilizadas por los estudiantes para conocer su progreso (herramienta "mygrades"). Por otro lado, con las reglas 8 y 10, el profesor puede conocer que los estudiantes del curso suelen acceder al foro y a las páginas de contenidos para responder a las preguntas de los auto-test del curso (herramienta "assessment").

En la Tabla 6.25 se muestran las reglas obtenidas por Yacaree para el dataset5, correspondiente al curso1. También en este caso vuelven a aparecer reglas con información obvia. En el conjunto de reglas, aparece la misma regla 1 que con el dataset3, y por ejemplo, las reglas 2 y 40 informan únicamente de que los estudiantes a veces sólo se conectan para conocer las fechas de los entregables. No obstante, también existen otras reglas que pueden ofrecer al profesor un conocimiento añadido sobre el comportamiento de sus estudiantes, como son las reglas 7, 14 y 36, que establecen que los estudiantes acceden al foro y a las páginas de contenidos para consultar dudas acerca de las tareas entregables. Más aún, la regla 4 le puede informar de que los estudiantes hacen uso de los videotutoriales de la asignatura (herramienta "learningobjectives") al tiempo que

TABLA 6.25: Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset5

No.	Association rule	(Supp., Conf., Lift, Cboost)
1	filemanager ⇒ assignments	(5.1, 71.5, 1.871, 1.871)
2	calendar ⇒ assignments	(6.1, 74.9, 1.961, 1.610)
40	announcement ⇒ assignments	(3.9, 67.2, 1.759, 1.153)
3	weblinks ⇒ contentpage	(3.7, 78.2, 2.105, 1.588)
4	learningobjectives ⇒ contentpage	(4.5, 81.4, 2.192, 1.530)
7	contentpage mygrades ⇒ assignments	(2.7, 66.7, 1.746, 1.421)
14	assignments whoisonline ⇒ discussion	(1.7, 72.5, 1.612, 1.301)
36	discussion weblinks ⇒ assignments	(1.9, 73.4, 1.923, 1.180)

consultan los contenidos propios del curso. Estos videotutoriales requirieron un gran esfuerzo por parte del profesor de cara a elaborarlos, por lo que conocer esta información le resultó extremadamente útil. Adicionalmente, la regla 3 le sirvió al profesor para conocer que los recursos externos que recomienda en la asignatura (herramienta "weblinks") son utilizados en conjunto con los apuntes internos del curso en el proceso de aprendizaje de los estudiantes.

En cuanto a los resultados con los dataset2 y dataset4, en los que no se ha eliminado la herramienta "Organizer", el número de reglas retornadas tanto por Apriori de Weka, como por Apriori de Borglet, Predictive Apriori y ChARM es extremadamente alto, tal y como pudo observarse en la Tabla 6.17. En el caso de Yacaree, el subconjunto de reglas retornado también crece notablemente con respecto a no incluir "Organizer", aunque su número, 255 y 182, sigue siendo bastante más bajo que el alcanzado con los otros algoritmos. Las Tablas 6.26 y 6.27 contienen un subconjunto de reglas obtenidas por este algoritmo para ambos datasets, y en ambos casos podemos observar el mismo problema: una de las reglas que aparece con más confianza es la que indica que, pase lo que pase en la sesión (antecedente vacío), la herramienta "Organizer" es utilizada, lo que es una obviedad dado que esta herramienta es la puerta de entrada usual al curso. Más aún, como puede observarse en las reglas 158 y 287 para el dataset2, y en las reglas 9 y 113 para el dataset4, aparecen reglas con una información similar, en la que la única diferencia es que en el consecuente encontramos la herramienta "Organizer", con una variación notable en la confianza que hace que Yacaree no las considere redundantes.

En el caso en donde sin duda es más notable el beneficio de eliminar reglas redundantes es al aplicar los algoritmos de reglas sobre el dataset1, que contiene la información acerca de a qué ficheros *.pdf* con el temario de la asignatura acceden los estudiantes en una misma sesión. Mientras que Yacaree retorna únicamente 40 reglas, con ChARM este

Tabla 6.26: Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset2

No.	Association rule	(Supp., Conf., Lift, Cboost)
2	⇒ organizer	(83.9, 83.9, 1.000, 1.982)
158	mygrades tracking ⇒ assessment organizer	(4.6, 71.7, 1.888, 1.109)
287	mygrades tracking ⇒ assessment	(5.0, 78.6, 1.818, 1.096)

Tabla 6.27: Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset4

No.	Association rule	(Supp., Conf., Lift, Cboost)
1	⇒ organizer	(84.9, 84.9, 1.000, 2.421)
9	chat ⇒ discussion organizer	(2.0, 77.6, 2.324, 1.283)
113	chat ⇒ discussion	(2.2, 84.2, 1.954, 1.085)

Tabla 6.28: Subconjunto de reglas obtenidas al aplicar ChARM sobre el dataset1

No.	Association rule	(Supp. , Conf.)
6	topic7 topic9 topic10 topic-pdf ⇒ topic8	(1.23, 100.00)
7	topic7 topic8 topic9 topic-pdf ⇒ topic10	(1.23, 100.00)
8	topic7 topic8 topic9 topic10 ⇒ topic-pdf	(1.23, 100.00)
9	topic7 topic9 topic-pdf ⇒ topic8 topic10	(1.23, 100.00)
65	test5 test7 test8 test9 topic10 topic-pdf ⇒ test6	(1.47, 100.00)
66	test5 test6 test8 test9 topic10 topic-pdf ⇒ test7	(1.47, 100.00)
67	test5 test6 test7 test9 topic10 topic-pdf ⇒ test8	(1.47, 100.00)

número crece en una proporción mayor de 9:1, mientras que con Apriori de Borglet el número de reglas llega a ser de 524. Al igual que en casos anteriores, Apriori de Weka y PredictiveApriori sobre pasan las 2000 y 1000 reglas retornadas respectivamente. En la Tabla 6.28 se muestra un subconjunto de ejemplo de las reglas retornadas por ChARM, en dónde se puede observar que existe una alta cantidad de reglas con una confianza del 100 % y un soporte muy bajo, de 1.23 % y 1.47 % en el ejemplo mostrado, que retornan una información muy similar, indicando que los temas ("topic") del 6 al 10 suelen accederse en la misma sesión.

De todo ello se puede concluir que los únicos resultados interpretables, sobre todo de cara a mostrárselos a un usuario no experto, son los ofrecidos por Yacaree. En la Tabla 6.29 se muestran algunas de las reglas que este algoritmo ha retornado, y que le han sido mostradas al profesor del curso. Lo primero que destaca es que todas las reglas indican que el curso está claramente dividido en 2 mitades: una primera, en la que se accede sobre todo a los temas del 1 al 5, y una segunda mitad en la que se accede a los temas del 6 al 10. El profesor pudo observar, además, que muchos de los temas no son revisados por los estudiantes, sino que estos realizan el aprendizaje en base a los auto-test del curso, lo que desde su punto de vista supone una mala aplicación del proceso de aprendizaje.

Tabla 6.29: Subconjunto de reglas obtenidas al aplicar Yacaree sobre el dataset1

No.	Association rule	(Supp., Conf., Lift, Cboost)
1	topic6 ⇒ topic-pdf	(13.3, 1.0, 2.544, 2.544)
2	topic7 ⇒ topic-pdf	(9.8, 1.0, 2.544, 2.500)
3	topic4 topic-pdf ⇒ topic5	(6.4, 76.5, 5.764, 2.266)
18	topic1 topic3 ⇒ topic2	(3.9, 72.7, 4.055, 1.377)
6	topic9 ⇒ topic10 topic-pdf	(0.057, 1.0, 7.537, 1.917)
7	topic10 topic7 ⇒ topic8 topic-pdf	(0.037, 1.0, 14.536, 1.875)
23	topic-pdf topic10 topic6 ⇒ topic8	(2.9, 66.7, 9.690, 1.286)
40	exam2 topic-pdf ⇒ topic10	(1.7, 77.8, 5.862, 1.167)
9	test2 ⇒ test1 test3	(4.9, 71.4, 13.844, 1.667)
10	test9 ⇒ test6 test7 test8 topic-pdf topic10	(2.5, 66.7, 27.133, 1.667)
14	test7 topic-pdf topic10 ⇒ test6 test8 test9	(2.5, 76.9, 31.308, 1.538)
23	test9 ⇒ test8 topic-pdf topic10	(3.4, 93.3, 23.742, 1.273)
28	test3 test4 ⇒ test5 topic-pdf	(2.7, 73.3, 14.213, 1.222)

6.2.2. Conclusiones y trabajo futuro

Los modelos de reglas de asociación obtenidos con algoritmos como Apriori pueden resultar muy difíciles de interpretar, debido a la gran cantidad de reglas que generan, muchas de ellas redundantes. El algoritmo Yacaree soluciona este inconveniente, eliminando la redundancia entre reglas. En el estudio mostrado, se constata que los modelos ofrecidos con Yacaree reducen enormemente el conjunto de reglas con respecto al resto de algoritmos, sin perder información importante, y pudiendo por tanto ser interpretadas de forma sencilla.

Capítulo 7

Extensión de E-learning WebMiner: definición de nuevas plantillas

En el capítulo 1 se presentó E-learning WebMiner (ElWM), una herramienta que, funcionando como aplicación Web, ofrece a los profesores de cursos virtuales la posibilidad de obtener modelos de minería de datos de forma simple e interpretable, posibilitando un mejor entendimiento de la actividad sus estudiantes que les ayudase a mejorar el proceso de enseñanza y aprendizaje. Antes de comenzar el estudio de esta tesis, ElWM utilizaba únicamente técnicas de *clustering* y de reglas de asociación para dar respuesta a 3 preguntas concretas que los profesores de cursos virtuales podrían hacerse en base a la actividad de sus estudiantes: "¿Qué perfil de estudiantes existe en el curso?", "¿Qué herramientas son frecuentemente utilizadas de forma conjunta en una sesión de trabajo?" y "¿Qué perfil de sesiones existe en el curso?".

En el presente capítulo, se detalla un nuevo diseño de ElWM, al que se le añaden nuevas funcionalidades de predicción del rendimiento de los estudiantes y de análisis de redes sociales en los foros, gracias a la investigación desarrollada y mostrada en los capítulos anteriores.

El presente capítulo se estructura de la siguiente forma: el apartado 7.1 contiene un resumen de los trabajos publicados entorno al desarrollo de sistemas que ofrezcan minería de datos a usuarios no expertos. En el apartado 7.2 se describe el diseño de la arquitectura

de ElWM. En el apartado 7.3 se definen los procesos que se ejecutan en ElWM cuando un usuario interactúa con la herramienta. El apartado 7.4 muestra un nuevo diseño de ElWM en el que se incluyen los procesos y sistemas de meta-learning, de detección de comportamientos anómalos, y de Análisis de Redes Sociales de los capítulos anteriores. Seguidamente, en el apartado 7.5 se muestran ejemplos de uso y ejecución de ElWM. Finalmente, el apartado 7.6 contiene los trabajos en curso relacionados con ElWM.

7.1. Estado del Arte: herramientas de minería de datos para usuarios no expertos en el campo educativo

En los últimos años, y debido a la cada vez mayor cantidad de datos publicados en abierto, ha ido crecido el interés en el estudio y diseño de técnicas y herramientas que permitiesen a usuarios no expertos extraer conocimiento mediante técnicas de minería de datos [11]. En lo que respecta al proceso de extracción del conocimiento, la fase selección de los algoritmos de minería de datos adecuados se encuentra en el núcleo del mismo [294], por lo que, con objeto de ofrecer herramientas orientadas a usuarios no expertos, se muestra necesario el desarrollo de procesos de apoyo en esta fase. En este sentido, en la literatura pueden encontrarse diversas ontologías de minería de datos que proveen el adecuado conocimiento para ayudar en esta selección. Como ejemplos, pueden citarse OntoDM [295], enfocada a todo el dominio de minería de datos; o EXPO [296], más enfocada a experimentos científicos. Otras ontologías no solamente se centran en la selección de algoritmos de extracción del conocimiento adecuados, sino que también describen flujos de trabajo KDD, como son KDDONTO [297], DMOP [298] y *the Ontology-Based Meta-Mining of Knowledge Discovery Workflows* [299]. Más aún, en [300], los autores desarrollan una metodología específica para describir experimentos en el campo del *Machine Learning* en la que proponen un proceso colaborativo de análisis de los algoritmos de aprendizaje.

No obstante, existen un bajo número trabajos en la literatura en los que se propongan aplicaciones y herramientas de minería de datos que se orienten exclusivamente a usuarios no expertos, esto es, que permitan a usuarios con este perfil extraer el conocimiento que ofrecen las técnicas de minería de datos. En cuanto a aplicaciones de propósito general, existe en la herramienta RapidMiner un módulo, desarrollado por Reif et al. [159],

que hace uso de técnicas de meta-learning para la recomendación de algoritmos de clasificación. Sin embargo, el uso de este módulo requiere por parte de los usuarios ciertos conocimientos sobre minería de datos. En Campos et al. [301], los autores realizaron una propuesta de sistema orientado a que usuarios no expertos puedan obtener modelos de minería de datos que, aunque no sean tan precisos como los obtenidos por un experto, lo sean "razonablemente". Para ello, establecieron distintas fases del proceso de extracción del conocimiento que habrían de ser automatizadas (selección de atributos, selección de algoritmos, etc.) si bien no especifican en este trabajo como se debería de realizar la automatización de cada una de estas fases. Espinosa et al. [11] propusieron en su trabajo un framework que "democratizase" la minería de datos, de forma que usuarios no expertos en el área puedan obtener información sobre datos abiertos. Por otro lado, en el área de la medicina pueden encontrarse varias propuestas de herramientas para la extracción y análisis de datos enfocadas a usuarios no expertos, si bien en todos estos casos no existen realmente procesos que automaticen las fases KDD, y más concretamente la de selección de algoritmos, sino que existen una serie de algoritmos prefijados para cada una de las tareas, y en muchos casos son necesarios ciertos conocimientos en minería o análisis de datos para manejarlas [21, 22, 302–306].

Con respecto al campo educativo, Yannis et al. [307] mencionaron en su trabajo que los *"LMSs provide a limited set of reporting features and it is very difficult for an educator to extract useful information"*, y expusieron en el mismo el diseño y uso de una aplicación que mostrase a los profesores de cursos virtuales información sobre los hábitos navegacionales de sus estudiantes utilizando para ello técnicas de reglas de asociación. García et al. [17] apuntaron al mismo problema, y destacan que, si bien se han desarrollado y publicado numerosas herramientas de análisis de datos educativos [18–20, 308, 309], ninguna de ellas está enteramente enfocada a usuarios no expertos, requiriendo para ser usadas ciertos conocimientos sobre minería de datos. En este trabajo, los autores propusieron un sistema colaborativo en el que, por una parte, se implementó una aplicación cliente que generaba y mostraba al profesor reglas de asociación con información acerca de la actividad de los estudiantes en sus cursos. Por otro lado, se diseñó una aplicación en el lado del servidor utilizada únicamente por usuarios expertos, que se encargarían de la tarea de puntuar las diferentes reglas de asociación obtenidas por la aplicación cliente. Dado que esta aplicación cliente se orientó a usuarios no expertos, utilizaba el algoritmo Predictive Apriori, requiriendo del usuario como parámetro únicamente el número de

reglas que desease generar. Al igual que en el trabajo de Yannis et al., no se seleccionaba el algoritmo que mejor se adaptase a los datos del usuario, sino que se utilizaba uno ya prefijado. Lo mismo ocurría en Jugo et al [310], en dónde se utilizó un algoritmo de *clustering* prefijado, k-Means, para agrupar a los estudiantes, si bien en este trabajo los autores automatizaron el proceso de selección del número de clústers para evitar que el usuario no experto tuviese que introducirlo.

7.2. Arquitectura y nuevos servicios en ElWM

ElWM fue diseñado siguiendo una Arquitectura Orientada a Servicios (SOA). En la Figura 7.1 se muestra un prototipo de diseño, al que se han añadido dos nuevos servicios: WS-SNA y WS-Classification. El primero de ellos ofrece una nueva funcionalidad en ElWM consistente en mostrar al profesor de cursos virtuales las características de la interacción social de sus estudiantes en los foros de sus cursos. Este servicio, por tanto, se basa en la experimentación mostrada en el capítulo 6 de la presente tesis. El otro servicio, WS-Classification, ofrece al profesor la posibilidad de obtener modelos de predicción del rendimiento de los estudiantes. En este servicio de predicción, por tanto, se añade el proceso de automatización de selección de clasificadores basado en la experimentación descrita en el capítulo 4, y del que se dan más detalles en el apartado 7.4, así como el proceso de eliminación de comportamientos anómalos en las instancias de entrenamiento basado en la experimentación del capítulo 5. Además, el servicio incluye la posibilidad de obtener modelos de predicción del rendimiento en base a las medidas de centralidad de la interacción de los estudiantes en los foros, y que en el capítulo 6 mostraron ser útiles para la tarea.

Los otros servicios incluidos en el nuevo diseño de ElWM son los que ya estaban implementados con anterioridad: WS-Clustering para poder ofrecer modelos basados en *clustering*, WS-Associator para modelos de reglas de asociación, WS-Repository para el acceso a los repositorios de datos de los LCMS cómo Moodle y BlackBoard, y WS-Visualization para poder ofrecer los modelos de minería de datos de forma no sólo textual, sino también gráfica, de manera que se facilite su entendimiento e interpretación por parte de los profesores.

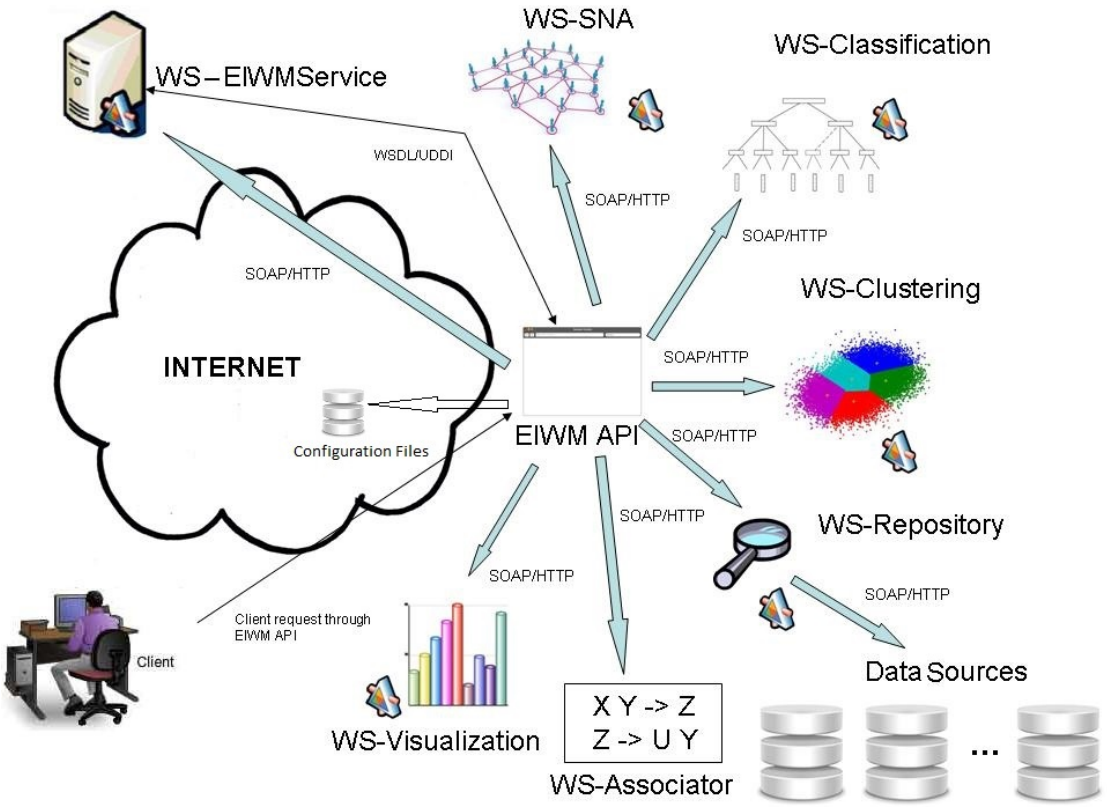


FIGURA 7.1: Arquitectura SOA de ElWM

Para orquestar los servicios anteriormente citados, se incluye el servicio WS-ElWM, descrito en *Web Services Description Language* (WSDL por sus siglas) [311], que expone los servicios que componen ElWM de forma que puedan ser utilizados por cualquier aplicación cliente.

7.3. Flujo de trabajo en ElWM: uso de los servicios

En la Figura 7.2 se muestra cómo se coordinarán los diferentes servicios ofrecidos por ElWM, de cara a mostrar los resultados obtenidos al usuario final. Este usuario, el profesor de un curso virtual, a través de un formulario Web como el mostrado en la Figura 7.3, seleccionaría inicialmente la información que desea obtener sobre su curso, en base a una serie de preguntas predefinidas en la herramienta. Haciendo uso de los nuevos servicios WS-SNA y WS-Classification, basados en la experimentación de esta tesis, se han definido 3 nuevas preguntas que pueden ser respondidas por ELWM: "Predicción del rendimiento o abandono de los estudiantes", "Análisis social del foro" y "Descubrimiento de comunidades sociales en los foros del curso".

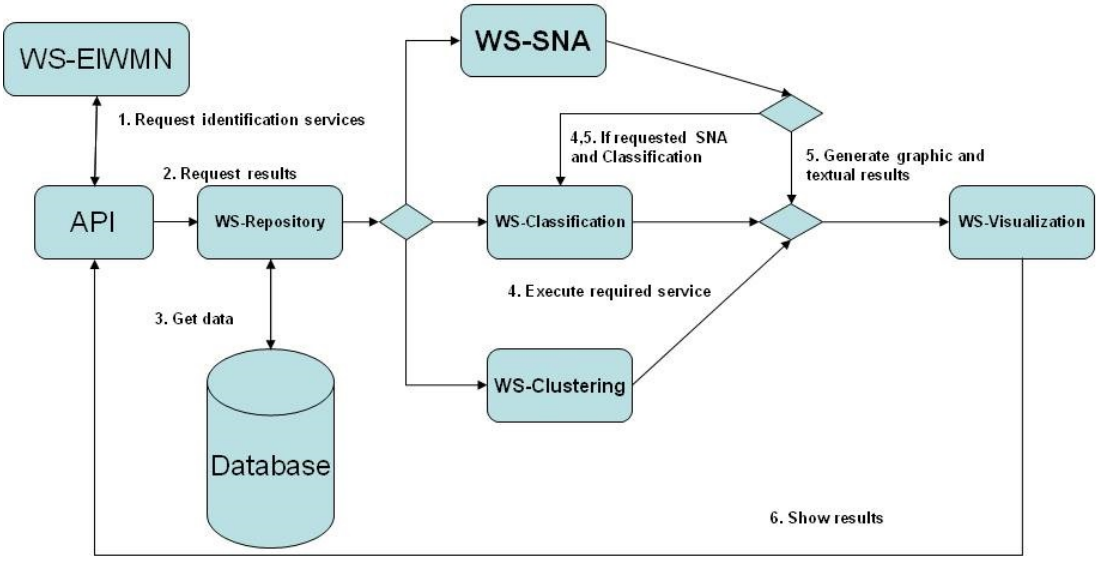


FIGURA 7.2: Flujo de trabajo de ElWM

FIGURA 7.3: Información que el usuario puede obtener con ElWM

Una vez el profesor haya escogido la pregunta, ElWM llama al servicio WS-Repository para acceder a los datos de actividad de los estudiantes, de cuyo curso el profesor requiere la información. Estos datos son obtenidos mediante consulta a la base de datos del LCMS en cuestión. En la Figura 7.4 se muestra un ejemplo de las opciones que se le ofrecerían al profesor si hubiese, previamente, requerido la información acerca de la predicción del rendimiento o abandono de sus estudiantes. El profesor, por tanto, tendría la opción de decidir si quiere realizar la predicción solamente en base al rendimiento, el abandono, o ambos. Seguidamente, también tendrá la opción de decidir si desea realizar

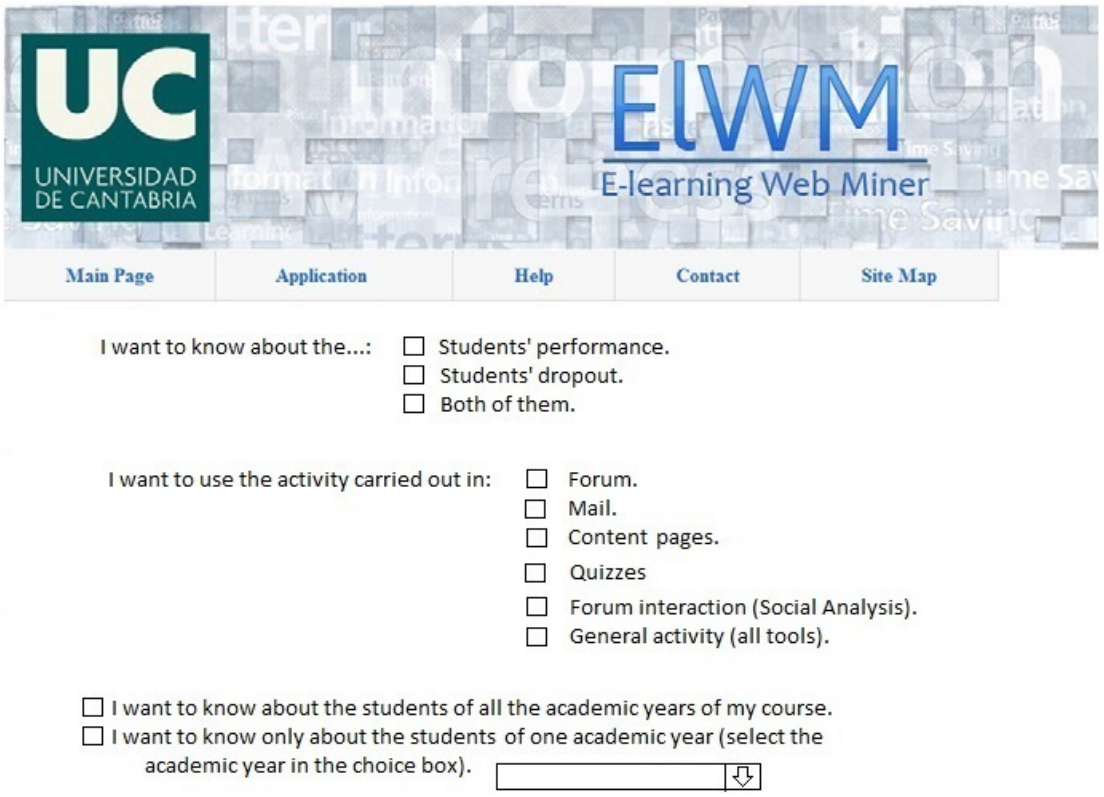


FIGURA 7.4: Opciones que ElWM ofrece al usuario para generar modelos de predicción del rendmiento de los estudiantes

la predicción del rendimiento en base exclusivamente a la actividad de sus estudiantes de unas herramientas concretas (por ejemplo, las páginas de contenidos o el foro), si desea realizarla en base a su actividad social, o si desea utilizar la actividad en todas las herramientas del curso. Adicionalmente, el profesor podrá escoger si desea generar un modelo de predicción en base a los estudiantes de un determinado año académico, o en base a la actividad de los estudiantes de todos los años académicos en los que se ha celebrado el curso.

En el siguiente paso, ElWM llamará a uno de los servicios que ofrecen modelos de minería de datos, dependiendo del tipo de información requerida por el profesor. En caso de requerir un modelo de predicción del rendimiento basado en la actividad social, ElWM llamará primero al servicio WS-SNA para obtener las medidas de centralidad que definan dicha interacción y, posteriormente, llamará al servicio WS-Classification para realizar la mencionada predicción.

Finalmente, ElWM ejecutará el servicio WS-Visualization para obtener una representación tanto gráfica como textual de los modelos de minería de datos que mostrar al

profesor. Estos modelos serán finalmente mostrados al usuario a través de la API.

7.4. Prediciendo el rendimiento de los estudiantes con ElWM: incorporación del proceso de meta-learning

En el apartado 4.4.5 del capítulo 4 se muestra que el proceso de meta-learning basado en la construcción de un ranking, utilizando meta-regresores para predecir el *accuracy* de cada clasificador, es especialmente útil para automatizar la selección de un buen clasificador que prediga el rendimiento de los estudiantes. Por ello, en el servicio WS-Classification de ElWM, se embeberán estos modelos de regresión, tal y como se muestra en la Figura 7.5. Estos modelos, por tanto, se construyen en base a los conjuntos de datos que contienen la actividad de los estudiantes de los cursos virtuales disponibles en la experimentación de esta tesis, de forma tal que, cuando un profesor requiera de ElWM generar un modelo de predicción del rendimiento de los estudiantes, la herramienta haga uso de este proceso de meta-learning para generar un ranking de clasificadores y, finalmente, obtener automáticamente y mostrar al usuario el modelo de clasificación con un mayor valor de *accuracy* predicho.

De esta manera, ElWM garantizará al usuario que se le esté mostrando un modelo de predicción lo más preciso posible, en lugar de tener prefijados un conjunto de algoritmos.

7.5. ¿Cómo ElWM muestra los modelos al usuario?: ejemplos

En el presente apartado, se muestran 2 ejemplos de cómo ElWM retorna los resultados al utilizar los 2 nuevos servicios: WS-Classification y WS-SNA. La Figura 7.6 contiene estos resultados para el primero de los mencionados servicios, asumiendo un caso de uso en el que el profesor requiere obtener una predicción del rendimiento de sus estudiantes en base a la actividad desarrollada en todas las herramientas del curso, e incluyendo las medidas de interacción social en el foro. Tal como puede observarse, al profesor se le muestra, en un primer apartado, información general sobre el curso: número de estudiantes clasificados, número de estudiantes suspensos y aprobados, número de medidas

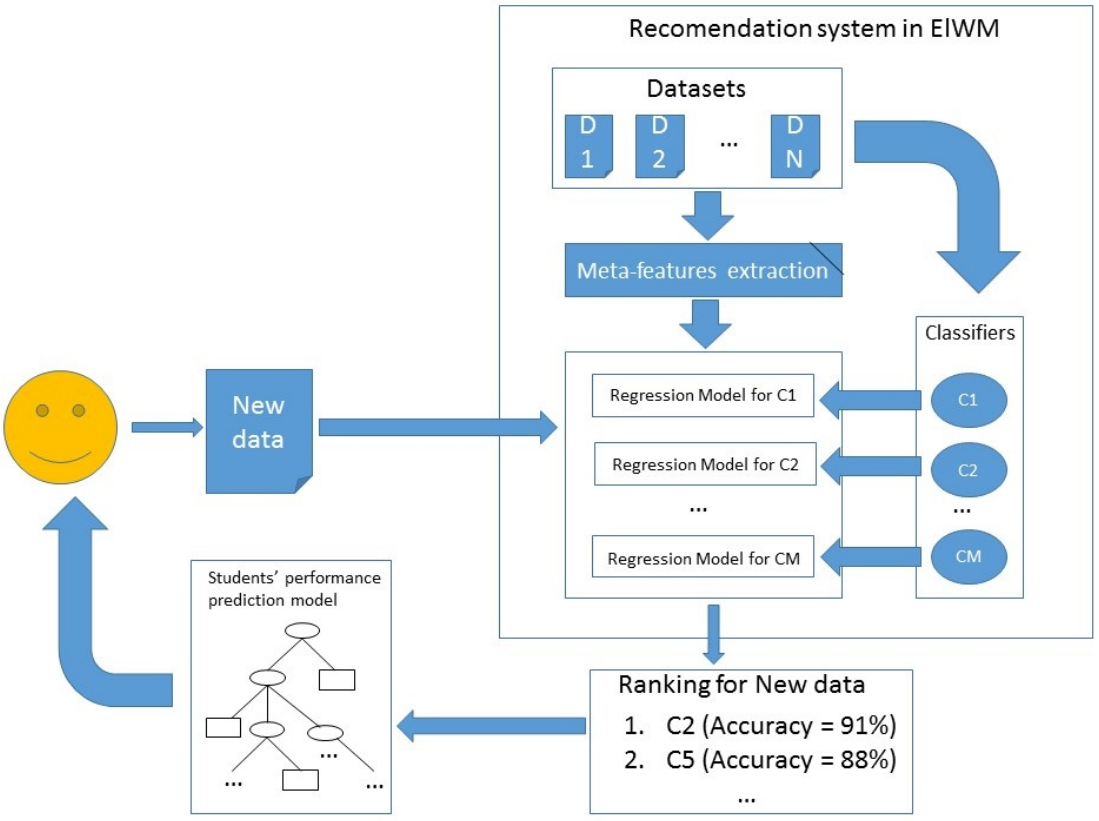


FIGURA 7.5: Despliegue de los meta-modelos para la selección del clasificador

de actividad (atributos predictores) utilizadas, etc. En un segundo apartado, se muestra el modelo de predicción de forma textual, junto con el valor de un conjunto de medidas que determinan la precisión de dicho modelo. Dado que ElWM está enfocado a ser utilizado por profesores de cursos virtuales que no han de tener conocimientos sobre minería de datos, estas medidas han de ser sencillas de interpretar, y por ello las escogidas para mostrarse son el *accuracy*, el *TPrate* (suspensos bien clasificados) y el *TNrate* (aprobados bien clasificados). En este apartado, además, existe un enlace, con el texto "How can i interpret these results? click here for help" en el que, al acceder a él, se le mostrará al usuario una pequeña guía sobre cómo ha de interpretar el modelo de minería de datos, ayudándole así a extraer la información mostrada. Dependiendo del clasificador utilizado por ElWM, esta página de ayuda puede variar. En este caso concreto, se informa al usuario de cómo interpretar un árbol de decisión. En el tercer apartado, se muestra el mismo modelo, pero esta vez de forma gráfica e, igualmente, existe un enlace de ayuda al usuario para interpretar el modelo.

En la Figura 7.7 se muestra un ejemplo de los resultados obtenidos cuando un profesor

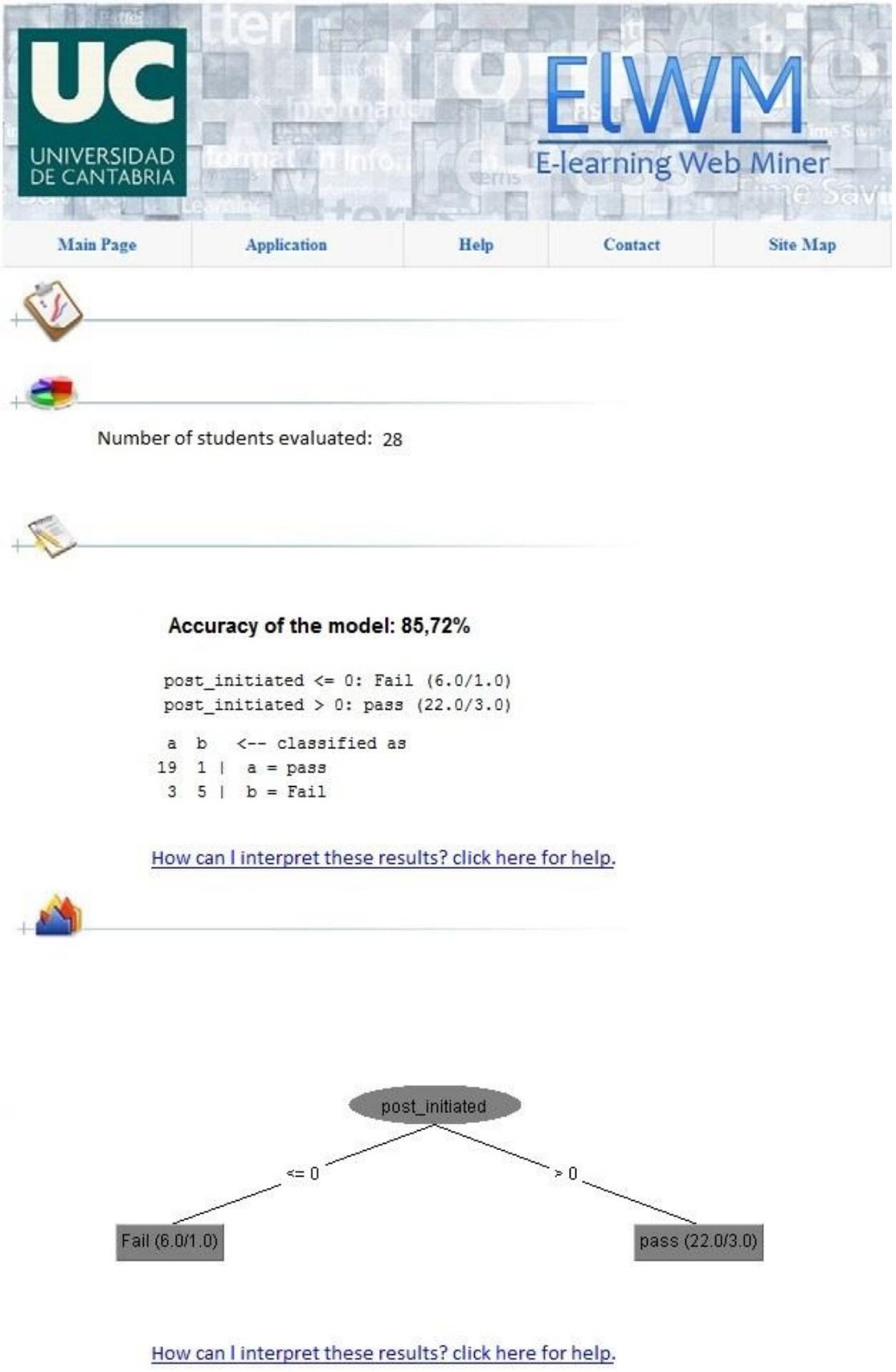


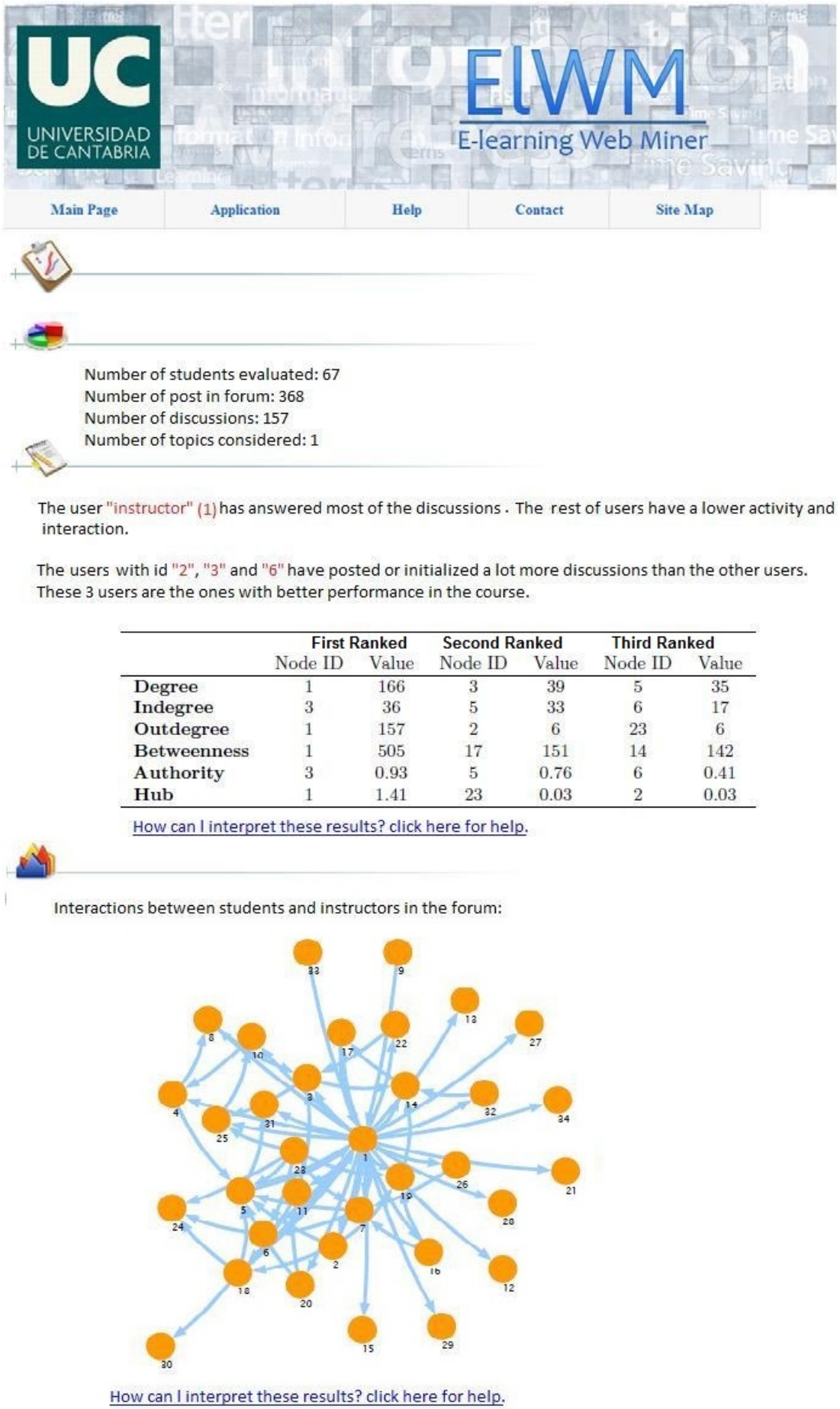
FIGURA 7.6: Predicción del rendimiento de los estudiantes mostrada al usuario por ElWM

requiere conocer la interacción social de sus estudiantes en el foro del curso, para lo que ElWM utilizará el servicio WS-SNA. Al igual que en el caso anterior de predicción del rendimiento de los estudiantes, en un primer apartado se muestra información general sobre los datos, en este caso, sobre la actividad en el foro: número total de mensajes escritos y leídos, número de estudiantes que han interactuado en el foro respecto del total de estudiantes en la asignatura, número de discusiones iniciadas, y número de foros estudiados. En un segundo apartado, se muestra un resumen de la actividad en el foro. En la tabla, el profesor podrá contrastar los valores de las medidas de centralidad de los 3 usuarios más activos. Igualmente, se le muestra un resumen textual y explicativo de lo observado en el foro, en este caso, que el profesor es el usuario con un mayor número de respuestas en el foro, con una gran diferencia con respecto al resto de usuarios (estudiantes). También se le informa textualmente de cuáles son aquellos estudiantes que han inicializado o respondido más veces que el resto, y de su rendimiento en el curso. En el tercer apartado, se muestra la red social del foro de forma gráfica, representando a los estudiantes como nodos, y a sus interacciones como enlaces.

7.6. Colaboraciones en curso

En el transcurso del presente trabajo de tesis, se ha trabajado en dos líneas de investigación relacionadas con E-learning WebMiner y el desarrollo de sistemas que ofrezcan a usuarios no expertos la posibilidad de obtener modelos de minería de datos. Ambas líneas de investigación se encuentran en una fase muy preliminar, con lo que en esta sección se muestran las primeras propuestas y resultados obtenidos, de los que ya existen publicaciones al respecto.

En el apartado 7.6.1 se muestra una propuesta diferente de diseño de ElWM como artefactos software, con objeto de ofrecer la posibilidad de que los usuarios puedan instalar en sus equipos solamente las funcionalidades concretas que necesiten de entre las que ofrece la herramienta. Esta línea de investigación se desarrolló en colaboración con el investigador del área de Ingeniería de Software Pablo Sánchez, de la Universidad de Cantabria.



	First Ranked		Second Ranked		Third Ranked	
	Node ID	Value	Node ID	Value	Node ID	Value
Degree	1	166	3	39	5	35
Indegree	3	36	5	33	6	17
Outdegree	1	157	2	6	23	6
Betweenness	1	505	17	151	14	142
Authority	3	0.93	5	0.76	6	0.41
Hub	1	1.41	23	0.03	2	0.03

FIGURA 7.7: Análisis de la interacción en el foro mostrado al usuario por ElWM

En el siguiente apartado 7.6.2, se realiza una primera propuesta al desarrollo de un sistema que extienda la funcionalidad de ElWM con objeto de que esta pueda ser ofrecida en otros campos diferentes al educativo.

7.6.1. E-learning WebMiner como Línea de Productos Software

En el presente capítulo, se ha mostrado una propuesta de diseño de ElWM basado en Servicios Web. No obstante, ElWM podría ser diseñado siguiendo otros paradigmas diferentes que podrían adaptarse mejor al entorno en particular donde vaya a ser utilizada, como pudieran ser las Universidades enmarcadas en el sistema educativo del Estado Español, o las Universidades de otro estado, con necesidades y problemáticas diferentes.

Así, por ejemplo, nos podemos encontrar con que en cada entorno en el que pudiera utilizarse ElWM, el sistema *e-learning* utilizado sea diferente (el diseño de ElWM contempla la posibilidad de funcionar con las bases de datos de Moodle y Blackboard) o, por ejemplo, existan diferentes estrategias, materias y métodos de organización en la impartición de cursos. En estas situaciones, se hace necesario un estudio de las necesidades concretas de cada usuario o institución, convertido ya en cliente, con objeto de desplegar, de entre todas las funcionalidades que ElWM ofrece, solamente aquellas que satisfagan su demanda.

Aunque existe la posibilidad de adaptar manualmente ElWM a cada entorno en el que pudiera desplegarse, este proceso puede ser tedioso y requerir mucho tiempo para su desarrollo en cada entorno particular. Una posible solución a este problema consiste en utilizar Líneas de Productos Software [27] (*Software Product Lines*, SPL por sus siglas), con el objetivo de automatizar el proceso de despliegue de las diferentes funcionalidades de ElWM. La idea al utilizar SPL en el diseño de ElWM es, por tanto, facilitar su implementación y despliegue en cada entorno particular, permitiendo al usuario seleccionar aquellas características que quiere utilizar, de entre todas las definidas en la herramienta.

El proceso SPL comprende 2 fases, mostradas en la Figura 7.8: (1) Ingeniería de Dominio, en la que se definen el conjunto de elementos que componen el software así como los procesos que permitan automatizar el correcto despliegue de los elementos seleccionados

en un una instalación concreta, y (2) Ingeniería de Aplicación, cuyo objetivo es la construcción de un producto software concreto en base a las características o funcionalidades que se deseen.

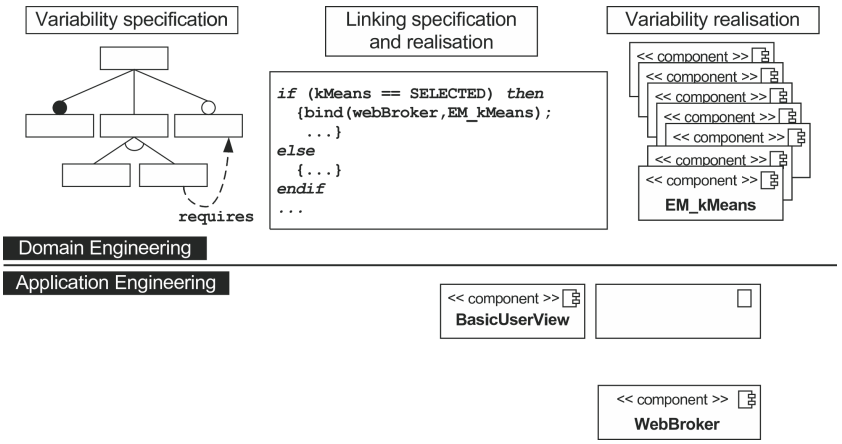


FIGURA 7.8: Proceso SPL

Mientras que la fase de Dominio sólo es ejecutada una vez, la fase de aplicación, por el contrario, es ejecutada cada vez que el software en cuestión, en este caso ElWM, vaya a desplegarse en un nuevo entorno. Es por ello que, cuanto más sencilla y rápida sea la fase de Aplicación, más beneficioso será el utilizar SPL. En la Figura 7.9 se ilustra esta idea, en la que se comparan los respectivos costes de utilizar SPL con respecto a aplicar un proceso de configuración manual cada vez que el software fuese desplegado en un entorno diferente (*Single-System Engineering*, SEE por sus siglas).

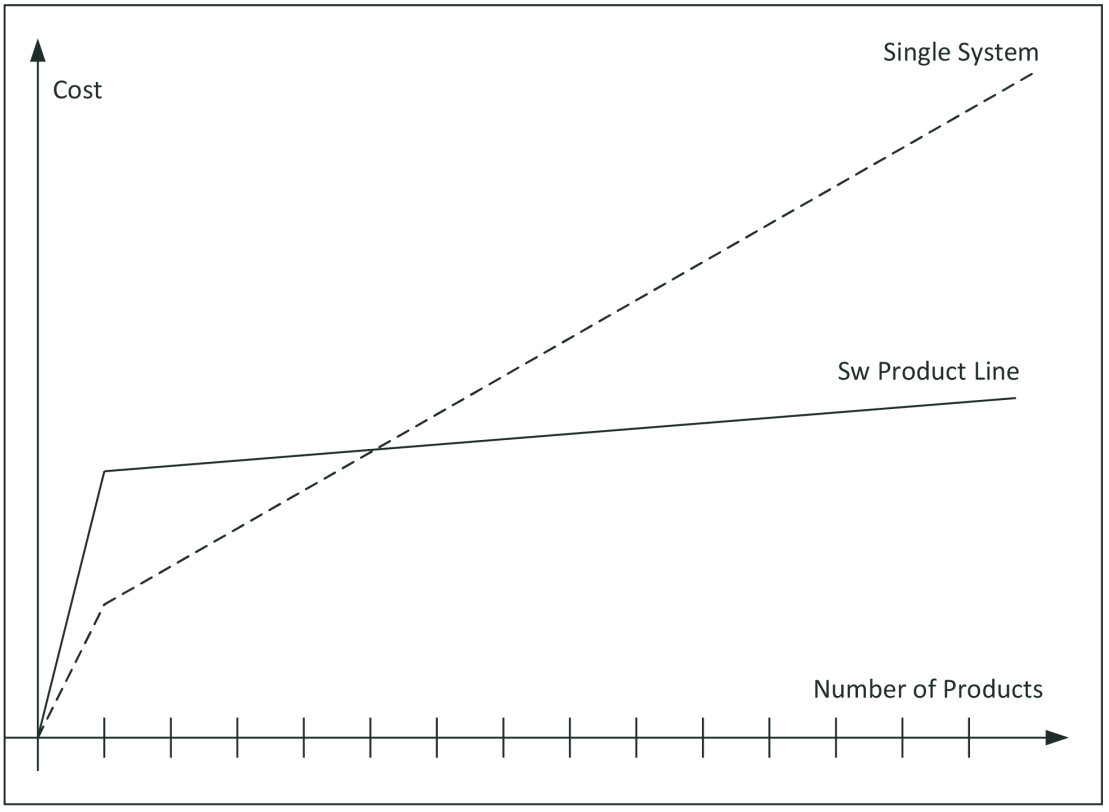


FIGURA 7.9: Costes *Software Product Lines* vs *Single-System Engineering*

En el caso del proceso SEE, inicialmente, una primera versión del software es desplegada en un entorno concreto. En el segundo despliegue que se realice, en un entorno diferente, el coste es más bajo que el del primero, debido a la posibilidad de reutilizar el proceso seguido en el primer despliegue. A partir de este punto, el coste de los siguientes despliegues crece de forma lenta, aunque constante. Al utilizar SPL, el coste inicial para el primer producto de software es más alto que con SEE, debido al gran costo que tiene la fase de Dominio. No obstante, una vez superada esta fase, el coste de desplegar los sucesivos productos software es casi nulo, debido a la automatización en la fase de Aplicación. Por ello, cuanto más se automatice esta fase, menor será el coste, y mayor el beneficio de utilizar SPL.

La fase de dominio se divide, a su vez, en tres fases. En la primera de ellas, denominada Análisis de Variabilidad, se realiza un análisis de las variabilidades inherentes al dominio en el que se trabaja, definiendo las características que obligatoriamente han de ser incorporadas en el producto, las opcionales, etc. En la Figura 7.10 se muestra un modelo de características para ElWM en forma de árbol, construido mediante una técnica de análisis de variabilidad llamada *Feature-Oriented Domain Analysis* (FODA

por sus siglas) [312]. En esta fase, también se definen las restricciones al despliegue como un conjunto de reglas que determinan las dependencias que existen entre los módulos, tal como se muestra en la Figura 7.11.

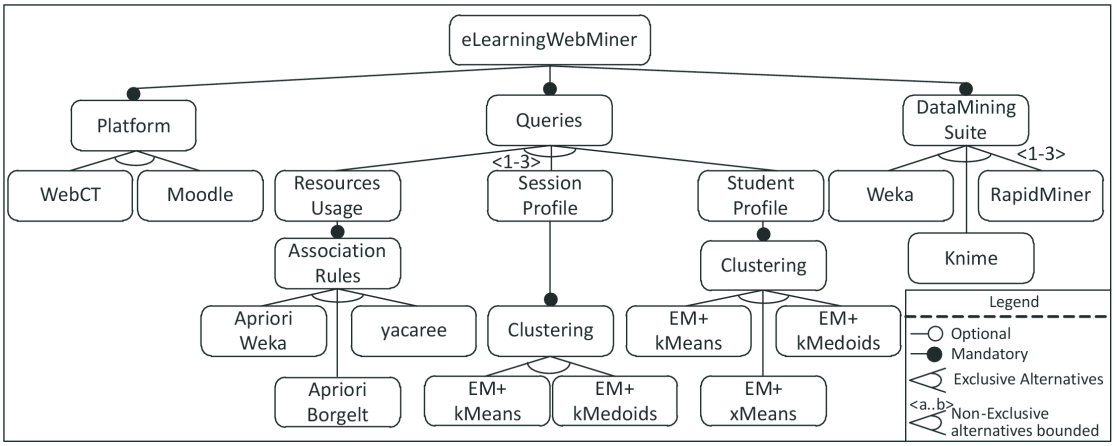


FIGURA 7.10: Modelo de características

1 EM+kMedoids **implies** RapidMiner
2 Apr ior iWeka **implies** Weka
3 EM+xMeans **implies** Weka
4 yacaree **implies** Knime

FIGURA 7.11: Reglas con restricciones al modelo de características

En un segundo paso de la fase de dominio, se define la arquitectura del software de forma que este sea adaptable, tal como se muestra en la Figura 7.12, pudiendo añadir o eliminar componentes según las necesidades.

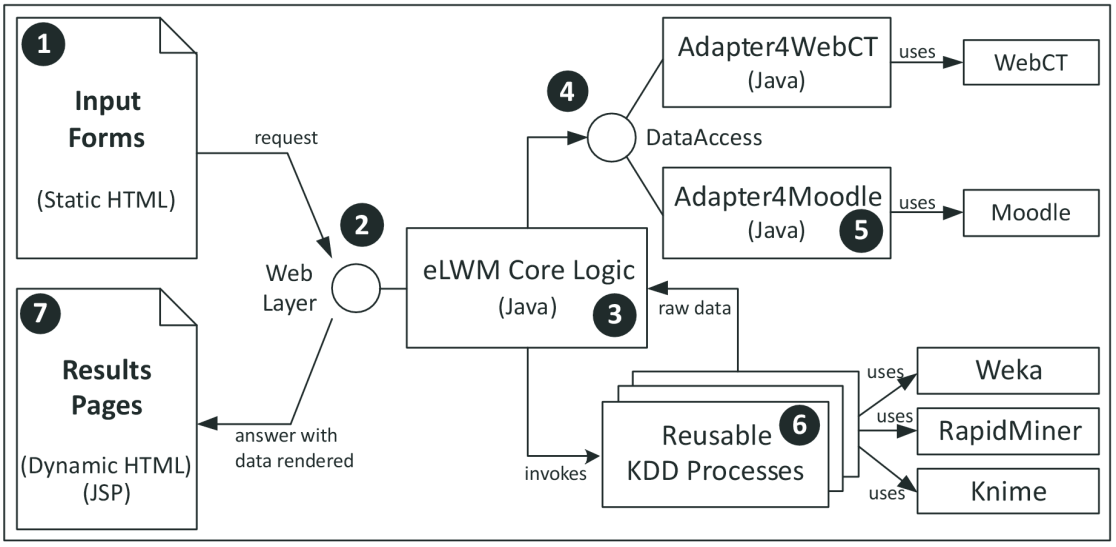


FIGURA 7.12: Ejemplo de arquitectura *Software Product Lines*

El último paso de la fase de dominio, se crean plantillas de generación de código definiendo las reglas de despliegue automático de los componentes software seleccionados por el usuario. En la Figura 7.13 se muestra un ejemplo de estas plantillas en lenguaje Epsilon [313].

```
1 . . .
2 [% if  featureModel.isSelected (“SessionProfile”)  { %]
3 <input  id=”demo” name=”queryKind” value=”Session”
type=”radio” />
4 Sessions Profile
5 [% } %]
6 [%if  featureModel.isSelected (“StudentProfile” ){ %]
7 <input  id=”demo” name=”queryKind” value=” Student”
type=”radio” />
8 Students Profile
9 [% } %]
10 [%if  featureModel . isSelected ( ”ResourceUsage” ) { %]
11 < input  id=”demo” name=”queryKind” value=” Resources”
type=”radio” />
12 Tools used together
13 [% } %]
14 . . .
```

FIGURA 7.13: Plantilla de generación de código con Epsilon

En cuanto a la fase de aplicación, esta se realiza en dos pasos. En un primer paso, se establecen las necesidades del usuario o cliente. En el caso de ElWM, esto involucra a dos actores: por un lado, al propio usuario que debe decidir que es lo que requiere de ElWM; por otro, a un experto en minería de datos, que habría de asistir aconsejar al usuario sobre que módulos de minería se adaptan mejor a sus necesidades. En la Figura 7.14 se muestra un ejemplo de una posible configuración de ElWM para un usuario concreto, con los módulos que éste finalmente ha decidido desplegar. En el segundo paso de la fase de aplicación, finalmente se construiría el sistema en base a la configuración escogida en el paso anterior, utilizando para ello las reglas automáticas de generación de código definidas en la fase de dominio.

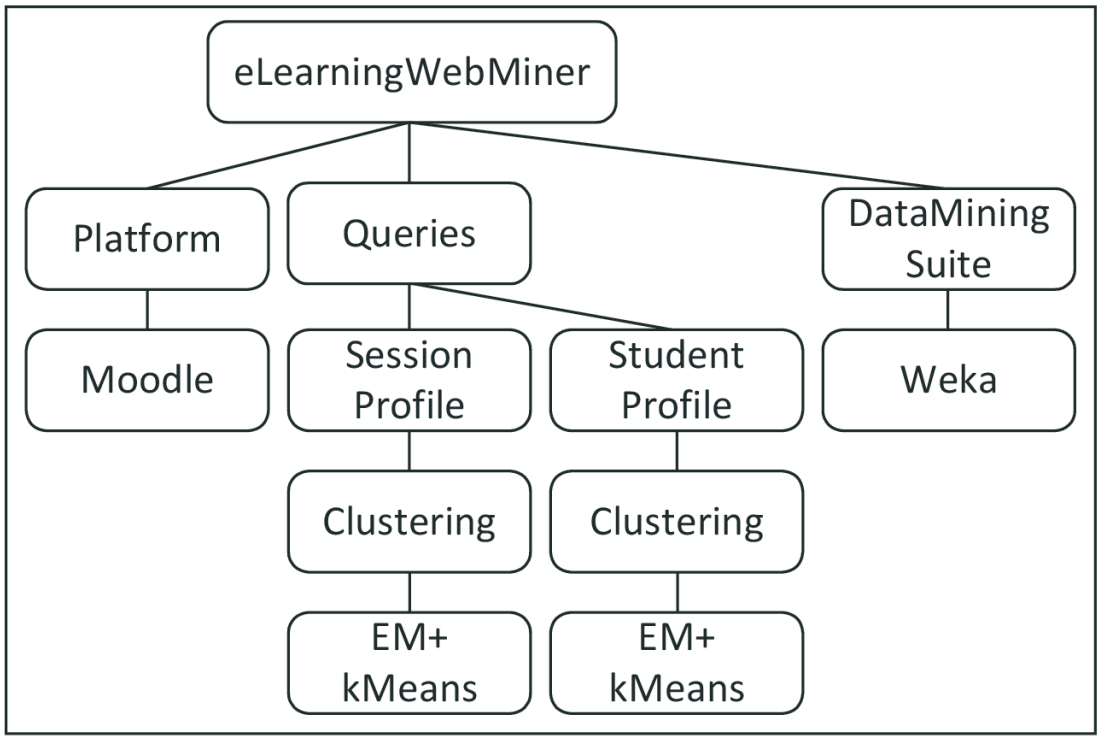


FIGURA 7.14: Ejemplo de configuración

7.6.2. Extendiendo la funcionalidad de ElWM a otras áreas

Movimientos como el de datos abiertos (*Open Data*) posibilitan que cada vez haya una mayor disponibilidad de datos accesibles para su reutilización. A pesar de que el número de herramientas analíticas que están a nuestra disposición crece cada día, lamentablemente, ninguna permite realizar un proceso completo de extracción de conocimiento directo a usuarios con poca o nula experiencia en el uso de la estadística y de algoritmos de minería de datos.

Siguiendo el esquema de E-learning WebMiner, se propone la creación de un marco KaaS (*Knowledge as a Service*) en el cuál un usuario no experto pueda utilizar servicios de minería de manera sencilla, esto es, un servicio que automatice y oculte las fases del proceso KDD. El objetivo de este nuevo marco es permitir al usuario especificar sus requisitos, y que el servicio seleccione el algoritmo que conlleve a la solución más precisa para el conjunto de datos bajo análisis.

La arquitectura propuesta es mostrada en la Figura 7.15. El usuario del sistema habría de proporcionar al mismo el fichero con los datos que desea analizar. Una vez realizada

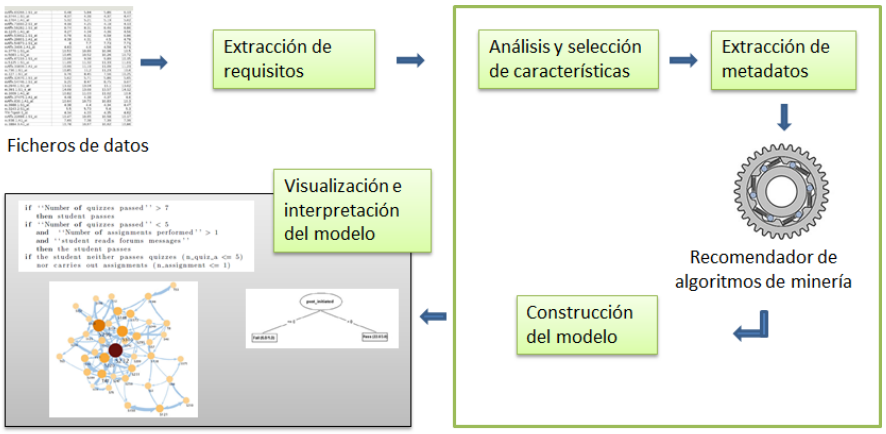


FIGURA 7.15: Arquitectura del servicio de minería propuesto

esta acción, el sistema deberá seleccionar la técnica o técnicas de minería de datos que han de aplicarse para dar respuesta a las necesidades del usuario. Para recoger estas necesidades, existen varias posibilidades. Una de ellas es la construcción de una taxonomía con la cuál, mediante preguntas encadenadas, se pueda determinar la técnica a aplicar en base a las respuestas del usuario. Otra posibilidad para recoger los requisitos del usuario es la utilización de Lenguajes Específicos de Dominio, con el que el usuario, en vez de responder a preguntas, escriba una consulta siguiendo las reglas de un lenguaje sencillo adaptado al dominio del conjunto de datos. En el artículo de Alfonso et al. [47] se expone una primera propuesta de cómo podría ser este lenguaje en el entorno educativo. Otras posibilidades, que se plantean también como reto futuro, es el empleo de lenguajes naturales [314].

En cuanto al módulo de minería, el proceso más importante es el de selección automática de algoritmos, para el que se podrían emplear las técnicas y procesos de meta-learning expuestas en el capítulo 4.

Finalmente, el módulo de visualización habría de garantizar que los modelos retornados al usuario sean sencillos de entender. Esto, a su vez, limita las técnicas que pueden emplearse en el módulo de minería. Aquí se plantean nuevos retos debidos principalmente a la dimensionalidad de las fuentes de datos.

Capítulo 8

Conclusiones

La aplicación de técnicas de minería de datos en el campo educativo es un área de investigación de creciente interés. El uso de estas técnicas cobra aún mayor relevancia en entornos de aprendizaje virtual, como son las plataformas de aprendizaje Moodle y Blackboard, en donde se almacena toda la información relativa a la actividad que los estudiantes desarrollan en cursos virtuales que se imparten tanto de forma semipresencial (*blended learning*) como de forma completamente virtual (*e-learning*). Diversos trabajos del área EDM (*Educational Data Mining*) han venido demostrando cómo la aplicación de este proceso puede ser útil para la extracción de conocimiento que permita a los profesores mejorar el proceso de enseñanza y aprendizaje en sus cursos virtuales, siendo **la predicción del rendimiento de los estudiantes** uno de los problemas más ampliamente demandados y estudiados.

Por otra parte, en la literatura se identifica como uno de los problemas del EDM el hecho de que los profesores de cursos virtuales no son, por norma general, **expertos en minería de datos**, y es por ello que la complejidad del proceso *KDD* dificulta e, incluso, hace que a estos usuarios les sea imposible la extracción de dicho conocimiento, por lo que se hace imperativo desarrollar procesos y ofrecer herramientas que les permitan a estos usuarios obtener estos modelos de minería de datos por sí mismos.

En esta tesis se marcaron dos objetivos: (1) el estudio de aplicación de técnicas de clasificación para la predicción del rendimiento de los estudiantes y el desarrollo de nuevos procesos y líneas de investigación que mejorasen la precisión de los modelos obtenidos, y

(2) el diseño, desarrollo y aplicación de procesos y herramientas que permitiesen a profesores de cursos virtuales la obtención y fácil interpretación de estos modelos, pudiendo así en base al conocimiento adquirido mejorar el proceso de enseñanza y aprendizaje en sus cursos. Con objeto de dar cumplimiento a estos objetivos, en el presente documento de tesis se han presentado los resultados y conclusiones de las diferentes líneas de trabajo desarrolladas.

En el capítulo 4 se han expuesto los resultados alcanzados más relevantes en el área de meta-learning, con objeto de desarrollar un recomendador capaz de seleccionar de forma automática los algoritmos de clasificación que, en base a las meta-características de los conjuntos de datos con la actividad y rendimiento de los estudiantes, tengan una precisión mayor que el resto o, al menos, una de las más altas. En los primeros estudios, las meta-características de tipo simple, de contexto y de complejidad mostraron ser útiles para construir meta-clasificadores, con J48 y NaïveBayes, de recomendación de algoritmos en base al que obtuvo mejor resultado sobre los conjuntos de entrenamiento. No obstante, se constataron las limitaciones que este proceso de selección de algoritmos tiene en el campo del EDM, debido fundamentalmente al bajo número de conjuntos de datos disponibles para la experimentación. Finalmente, en el último estudio, se realizó un cambio de enfoque, construyendo un sistema de recomendación basado en ranking con regresión, y añadiendo nuevas meta-características a la experimentación. Este sistema solventó las mencionadas limitaciones y mostró su utilidad al ser capaz de seleccionar en un 23.33 % de los conjuntos de datos de prueba al clasificador con mejor rendimiento en base al *accuracy*, en un 56.67 % a un clasificador del primer cuartil, y en un 83.34 % a un clasificador del primer o segundo cuartil. De esta forma, un profesor de cursos virtuales que quisiese obtener modelos de predicción del rendimiento de sus estudiantes no tendría la necesidad de conocer, aplicar y comparar diferentes algoritmos de clasificación para obtener un buen modelo, sino que el propio sistema es capaz de seleccionar un clasificador con buena precisión para sus datos.

Durante el desarrollo de esta tesis, se detectó que determinados comportamientos anómalos de los estudiantes, como eran por ejemplo los de aquéllos que pese a su bajo rendimiento dedicaban un alto número de sesiones y gran cantidad de tiempo a los cursos, afectaban negativamente a la precisión de los modelos predictivos. En el capítulo 5 se han mostrado una serie de estudios en los que se aplicaron diversas técnicas de detección de *outliers* y *class-outlier* con objeto de encontrar y eliminar estos comportamientos

anómalos de los conjuntos de entrenamiento, así como técnicas de ensamblado de clasificadores que en la literatura mostraban ser robustas ante la presencia de *outliers*, para tratar de mejorar la precisión de la predicción del rendimiento de los estudiantes. Por un lado, se pudo concluir que las técnicas de detección de *outliers* que no tienen en cuenta el valor de clase, como fue el caso de LOF, no mejoraban con su detección los modelos. Por otra parte, tanto las técnicas de detección de *class-outliers*, como ECODB, así como también las técnicas de ensamblado (Adaboost, Multiboost, RandomForest y Bagging) mostraron un comportamiento altamente irregular, ya que usualmente mejoraban la precisión de la clasificación de una de las clases en detrimento de la otra. Otra de las técnicas estudiadas y ampliamente utilizadas en la literatura, consistente en detectar como *class-outliers* aquellas instancias mal clasificadas por la mayoría de un conjunto de clasificadores (llamado proceso de Voto Mayoritario en el estudio), demostró sin embargo ser bastante efectiva, obteniendo notables mejoras en los modelos de predicción al eliminar las instancias detectadas. No obstante, esta técnica presentaba desventajas para los objetivos de esta tesis, y es que resultaba difícil de automatizar. Finalmente, se presentó el diseño de un nuevo proceso de detección de *class-outliers*, llamado DARIM (acrónimo de *Distance-based Algorithm to Remove Instances that Are Misclassified*) que, si bien obtenía unas mejoras de rendimiento algo más bajas que el proceso de Voto Mayoritario en términos de *accuracy*, alcanzaba una mejora mayor en la clasificación de los estudiantes con bajo rendimiento (suspensos). Además, DARIM tenía la ventaja de poder ser automatizado al aplicarse sobre conjuntos de datos del entorno educativo. Más aún, DARIM demostró su utilidad no solamente de cara a detectar y eliminar *class-outliers* para mejorar los modelos de predicción, sino también para detectar en el transcurso de los cursos virtuales a aquellos estudiantes con actividad anómala en riesgo de abandonar el curso, lo que se mostró de utilidad para que el profesor pueda identificar a estos estudiantes y enviarles realimentación personalizada con el fin de evitar el abandono y poder mejorar su rendimiento.

Con objeto de construir modelos más precisos e informativos, se incluyeron medidas de centralidad como atributos predictores, extraídas mediante aplicación de técnicas de *Social Network Analysis (SNA)* en los foros de los cursos virtuales. Estas medidas, tal como se desprende de los estudios mostrados en el apartado 6.1, mostraron su utilidad de cara a mejorar los modelos de predicción del rendimiento de los estudiantes. También se constató cómo la información extraída del proceso SNA aplicado a los foros puede ser

mostrada al profesor de forma tal que éste comprenda mejor cómo los estudiantes realizan las tareas que requieren de interacción social entre ellos, pudiendo así dinamizarlas y mejorarlas.

En el desarrollo de esta tesis, se realizó también un estudio conducente a mejorar la plantilla de reglas de asociación de ElWM, que inicialmente utilizaba el algoritmo Apriori para extraer los resultados. El principal problema que se encuentra al aplicar este algoritmo es que suele extraer un conjunto demasiado elevado de reglas, por lo que sus resultados se hacen difícilmente interpretables, y por tanto, accionables. En el apartado 6.2 se muestra una comparativa de diversas técnicas de reglas de asociación, y se constata cómo el uso de Yacaree, que elimina redundancia entre reglas, ayuda a mejorar la interpretabilidad del modelo de reglas obtenido.

Los anteriores estudios produjeron como resultado un nuevo diseño de la herramienta ElWM, que se muestra en el capítulo 7, y en la que se incluyeron nuevas plantillas orientadas a la obtención de modelos de predicción del rendimiento de los estudiantes, además de incluir también el proceso de recomendación de algoritmos basado en meta-learning, la técnica DARIM y el proceso SNA, tanto para la extracción de medidas de centralidad que mejoren los modelos de predicción como para el análisis de interacción social de los estudiantes. Asimismo, se propone una generalización de esta herramienta para ser utilizada en otros contextos diferentes al educativo.

En definitiva, gracias a los resultados obtenidos en los diferentes estudios, y las técnicas y procesos desarrollados, los profesores de cursos virtuales no expertos en minería de datos podrán ser capaces, a través de la herramienta ElWM una vez sea refactorizada con el nuevo diseño, de obtener modelos precisos e interpretables de predicción del rendimiento de los estudiantes, y en base a ellos entender mejor el comportamiento de sus estudiantes y tomar decisiones para mejorar la estructura de sus cursos, sus contenidos, sus actividades y otros aspectos que intervienen en el proceso de enseñanza y aprendizaje.

Capítulo 9

Trabajo futuro

Los resultados y conclusiones alcanzadas en los estudios realizados en el desarrollo de esta tesis han propiciado el desarrollo de nuevas líneas de trabajo.

En los últimos meses de realización de esta tesis han surgido nuevos trabajos relacionados con la investigación realizada en el campo del *meta-learning*, en los que se proponen nuevas meta-características que hasta el momento no habían sido apenas incluidas en otros estudios del área, como son las basadas en modelos o en el contexto. La inclusión de este tipo de meta-características en nuevos estudios podría propiciar un refinamiento del sistema de selección automática de clasificadores incluido en el nuevo diseño de ElWM. También en los últimos años, otros autores han abierto nuevas líneas de investigación de meta-learning enfocadas a la recomendación de algoritmos de *clustering* siguiendo procesos similares a los de los trabajos de recomendación de clasificadores. En un futuro, por tanto, se propone la posibilidad de estudiar la aplicación de estos procesos con el objetivo de incluir en el diseño de ElWM nuevos sistemas de recomendación de algoritmos para las plantillas que utilicen técnicas de *clustering* o incluso de reglas de asociación.

En relación a la detección de comportamientos anómalos, actualmente se está trabajando en la ampliación de los resultados del estudio 6 del capítulo 5, con el objetivo final de construir una herramienta software que detecte e informe automáticamente a los profesores de cursos virtuales acerca de aquellos estudiantes que, pese a tener una actividad alta en el curso, se encuentran en riesgo de abandonarlo debido a su bajo rendimiento, de forma tal que el profesor pueda intervenir a tiempo y realizar una realimentación

personalizada al estudiante, posibilitando así el evitar que abandone el curso o persista en su bajo rendimiento en el mismo.

Con respecto a la investigación desarrollada en el apartado 6.1, los buenos resultados obtenidos al utilizar medidas obtenidas con técnicas de SNA para mejorar la predicción del rendimiento de los estudiantes animan a continuar esta línea de investigación. En un futuro, se propone ampliar estos estudios con la utilización de otras medidas no incluidas en los mismos, o con la aplicación de los procesos de SNA en otros contextos de interacción social diferentes a los foros, como pueden ser cursos virtuales en los que se haga uso de redes sociales como Facebook o Twitter.

Por otro lado, si bien los objetivos de esta tesis se han enfocado a la clasificación del rendimiento de los estudiantes en cursos virtuales de educación superior, no se puede obviar que existen otro tipo de cursos, los MOOC, que se desarrollan de forma completamente on-line, y que en la actualidad tienen gran implantación. No obstante, estos cursos suelen tener la particularidad de tener una gran cantidad de estudiantes matriculados, al contrario que los utilizados en esta tesis, además de otras diferencias, como por ejemplo el perfil o procedencia de los estudiantes, las herramientas que ofrecen las plataformas, la actividad que almacenan, etc. Debido a esto, los resultados obtenidos en esta tesis no son extrapolables a este tipo de cursos. Es por ello que, en un futuro, resultaría de interés desarrollar nuevas líneas de trabajo en las que se valore la adaptación y aplicación a los MOOC de los diferentes estudios realizados en esta tesis.

También, tanto los estudios realizados en el campo de meta-learning como en la detección de comportamientos anómalos pueden ser aplicados, en un futuro, en otros contextos diferentes al educativo. En este sentido, en un futuro se propone trabajar en la comparativa de la efectividad de DARIM con respecto a otras técnicas de detección y eliminación de *class-outliers* con objeto de mejorar los modelos de predicción de forma general, es decir, sin enfocarse exclusivamente en conjuntos de datos del entorno educativo. Igualmente, una nueva línea de trabajo futura podría consistir en la aplicación y refinamiento del sistema de selección automática de clasificadores de forma que también pueda ser utilizado con conjuntos de datos provenientes de cualquier campo.

Por último, y tal y como ya se ha expuesto en el apartado 7.6.2, actualmente se está trabajando en el diseño de un sistema basado en ElWM, pero de ámbito generalista,

que permita a usuarios no expertos de cualquier campo obtener modelos de minería de datos de forma sencilla y cuyos modelos sean interpretables y accionables.

Apéndice A

Anexos

A.1. Ejemplos de uso de ElWM previos al desarrollo de esta tesis

Al inicio del desarrollo de esta tesis, ElWM estaba preparado para, utilizando técnicas de minería de datos, responder a tres preguntas diferentes que los profesores podrían hacerse sobre sus estudiantes y su actividad en cursos virtuales.

En el caso de la primera pregunta, con objeto de obtener el perfil de los estudiantes, ElWM hace uso de un algoritmo de *clustering*, k-Means. El motivo por el cual se seleccionó k-Means para ofrecer la información al profesor se basó principalmente en la experimentación previa realizada, que mostró que esta técnica tenía un buen rendimiento para generar este modelo, y en el hecho de es sencillo de interpretar incluso por usuarios no expertos, al contrario de lo que podría suceder con los modelos ofrecidos por otros algoritmos de *clustering*, como pueden ser los jerárquicos o los basados en densidad. En la Figura A.1 se muestra un ejemplo de cómo es mostrada la información al profesor cuando se utiliza un conjunto de datos con la actividad de los estudiantes en un curso *e-learning*, medida en sesiones y tiempo total y medio dedicados, además de los datos demográficos de género y edad. Como puede verse, ElWM ofrece los resultados de forma tanto textual como gráfica, facilitando así al usuario la interpretación y la extracción de conclusiones sobre la información mostrada. En este caso, el profesor puede observar como, en su curso, existen 3 grupos de estudiantes bien diferenciados: los del Cluster 2,

con una alta actividad y una edad superior a la de los otros 2 grupos, que representan a estudiantes con una actividad media (Cluster 0) y baja (Cluster 1).

Attribute	Cluster# Full Data (67)	0 (21)	1 (31)	2 (15)
age	22.3226	21.9662	22.2799	22.9097
gender	Man	Man	Man	Man
totalTime	1138.1791	1394.7143	103.6129	2917.1333
numberOfSessions	73.6418	93.9524	8.2903	180.2667
averageTimePerWeek	56.4776	69.1905	4.8065	145.4667
averageSessionsPerWeek	3.2836	4.1905	0.0645	8.6667

Clustered Instances

0 21 (31%)
1 31 (46%)
2 15 (22%)

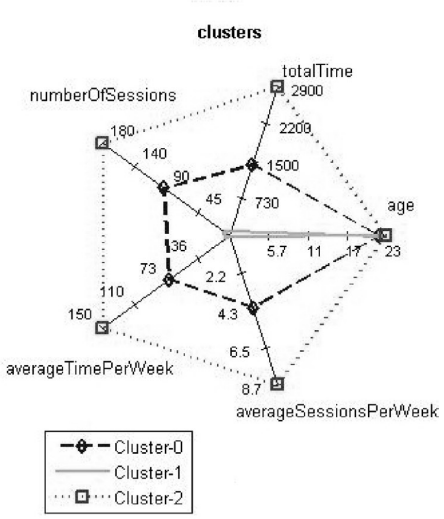


FIGURA A.1: Resultados para la pregunta "¿Cuál es el perfil de estudiantes de mi curso?"

Por los mismos motivos que en el caso anterior, el algoritmo de *clustering* k-Means es el utilizado para dar respuesta a la segunda pregunta, "¿Cuál es el perfil de sesiones del curso?". En este caso, ElWM trata de obtener un modelo que represente qué tipos de sesiones existen por parte de los estudiantes, esto es, identificar grupos de sesiones en los que los estudiantes se conectan al curso para realizar unas tareas concretas. Esto se ve más claramente en la Figura A.2, en la que se muestran los resultados obtenidos por ElWM para el mismo curso que en el anterior ejemplo, con los que el profesor, en este caso, puede observar que existen cuatro tipo de sesiones. En el Cluster 0 se agrupan las sesiones en las que los estudiantes dedican su tiempo a realizar las tareas del curso apoyándose en consultas del temario y del foro del curso, lo que puede concluirse de la alta actividad en las herramientas de tareas (time_assigments), de contenidos del curso (time_content-page y hit_content-page) y en el foro (time_forum y hit_forum). Por otra parte, el Cluster 1 caracteriza a aquellas sesiones utilizadas al estudio, con un tiempo muy alto dedicado al temario (time_content-page), mientras que los Clusters 2 y 3 representan unas sesiones de actividad más general.

Attributes	Cluster 0 (271)	Cluster 1 (256)	Cluster 2 (1051)	Cluster 3 (3356)
time_mail	1.3476	0.4169	0.8051	0.4140
time_forum	4.1622	3.8892	2.0904	0.7084
time_content-page	11.0927	68.6051	1.4502	3.3794
time_organizer	2.6655	6.7933	0.5586	0.7770
time_assignments	20.9470	4.4243	5.2296	0.5603
time_filemanager	0.9801	0.1697	0.1729	0.0128
time_who_is_online	0.1655	0.3321	0.0715	0.0619
time_assessment	4.4437	1.2952	0.6003	0.0566
time_my_grades	0.7185	0.3726	0.4135	0.1552
time_compiler	0.1721	1.9926	0.0347	0.1087
hit_mail	1.0264	0.4760	0.6610	0.3868
hit_forum	10.2682	6.4538	6.5029	1.9123
hit_content-page	3.5033	27.3763	0.6083	1.2411
hit_organizer	5.9304	11.5682	2.0039	1.8634
hit_assignments	5.6490	1.5202	2.4642	0.3511
hit_filemanager	0.5562	0.1217	0.1978	0.0220
hit_who_is_online	0.3973	0.2140	0.1153	0.0837
hit_assessment	2.7152	0.9409	1.1292	0.1642
hit_my_grades	0.6225	0.2546	0.5367	0.2131
hit_compiler	0.0695	0.4280	0.0049	0.0268

Filtered clusters

0	271 (5%)
1	256 (5%)
2	1051 (21%)
3	3356 (68%)

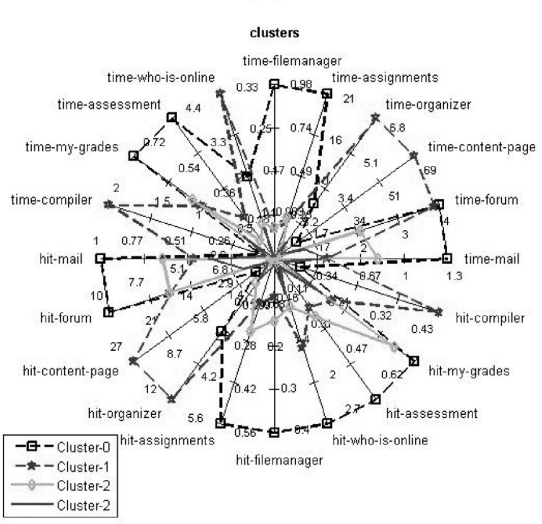
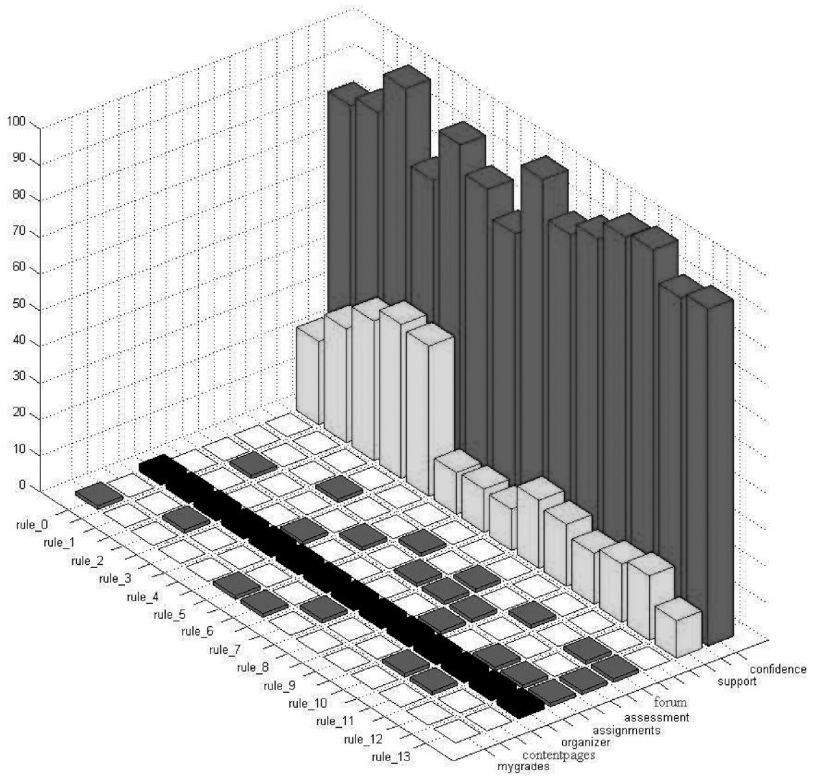


FIGURA A.2: Resultados para la pregunta "¿Cuál es el perfil de sesiones que existen en mi curso?"

Para obtener la información relativa a la tercera pregunta, "¿Qué recursos se suelen utilizar juntos en una sesión de aprendizaje?", ElWM tiene predefinido un algoritmo de reglas de asociación, Apriori. Los resultados que arroja la herramienta para el curso de ejemplo utilizado anteriormente pueden verse en la Figura A.3. Al igual que con los modelos de *clustering*, en este caso también se muestran los resultados tanto de forma textual como gráfica. En este caso, el profesor puede conocer, por ejemplo, que los recursos X e Y son utilizados usualmente juntos, lo que le puede dar una idea de como se desarrolla el proceso de aprendizaje de los estudiantes. Esto, a su vez, puede serle de gran ayuda para mejorar la estructura del curso, tal y como se constató al informarle de los resultados al profesor de este curso.




```
Rule_0: organizer <- my_grades (22.9, 84.4)
Rule_1: organizer <- assessment (31.3, 87.5)
Rule_2: organizer <- content-pages (38.4, 99.0)
Rule_3: organizer <- forum (42.4,78.7)
Rule_4: organizer <- assignments (41.4, 93.4)
Rule_5: organizer <- my_grades assessment (11.5, 86.2)
Rule_6: organizer <- my_grades forum (11.9, 78.8)
Rule_7: organizer <- assessment content-pages (11.3, 98.7)
Rule_8: organizer <- assessment forum (18.7, 88.4)
Rule_9: organizer <- assessment assignments (17.0, 92.2)
Rule_10: organizer <- content-pages forum (14.0, 97.7)
Rule_11: organizer <- content-pages assignments (16.0, 99.1)
Rule_12: organizer <- forum assignments (17.7, 90.8)
Rule_13: organizer <- assessment forum assignments (10.3, 92.7)
```

FIGURA A.3: Resultados para la pregunta "¿Qué herramientas se suelen utilizar habitualmente juntas por los estudiantes en una sesión de aprendizaje?"

A.2. Características de los cursos utilizados en la tesis

En la Tabla A.1 se muestran las características de los cursos virtuales utilizados en los estudios de esta tesis, indicando el número de estudiantes inicialmente matriculados en los mismos, la plataforma virtual en la que se impartieron, el carácter y el porcentaje de estudiantes que suspendieron en cada uno de ellos. En la Tabla A.2 se indican aquellas herramientas más frecuentemente utilizadas en el proceso de enseñanza y aprendizaje de cada curso.

Tabla A.1: Características de los cursos

Id	Plat.	Carácter	Tipo	Nº est.	% susp.
1	BB	Trans.	e-learning	64	51.56
2	BB	Trans.	e-learning	65	50.77
3	BB	Trans.	e-learning	64	71.88
4	MD	Trans.	e-learning	38	31.58
5	MD	Trans.	e-learning	33	12.12
6	MD	Esp.	blended	72	61.11
7	MD	Esp.	blended	136	84.56
8	MD	Esp.	blended	62	20.97
9	MD	Esp.	blended	66	43.94
10	MD	Esp.	blended	80	67.50
11	MD	Esp.	blended	13	38.46
12	MD	Esp.	blended	9	22.22
13	MD	Esp.	blended	262	83.59
14	MD	Trans.	e-learning	17	47.06
15	MD	Trans.	e-learning	28	39.29
16	MD	Trans.	e-learning	30	23.33
17	MD	Esp.	blended	502	13.75
18	MD	Esp.	blended	163	53.37
19	MD	Esp.	blended	182	26.37
20	MD	Trans.	e-learning	39	43.59
21	MD	Trans.	e-learning	22	86.36
22	MD	Trans.	e-learning	96	78.13
23	MD	Trans.	e-learning	43	18.60
24	MD	Trans.	e-learning	89	26.97
25	BB	Esp.	e-learning	20	65.00

Leyenda de la Tabla A.1:

- **Id:** Identificador único del curso con el que se realizan las referencias al mismo en el documento de tesis.

- **Plat.:** Plataforma LCMS en la que se aloja el curso (BB = Blackboard, MD = Moodle)
- **Carácter:** Indica si el curso es Transversal (Trans.) o Específico (Esp.).
- **Tipo:** Curso impartido completamente on-line (*e-learning*), o semipresencial (*blended*).
- **Nº est.:** Número de estudiantes matriculados en el curso.
- **% susp:** Porcentaje de estudiantes suspensos.

Tabla A.2: Herramientas más frecuentemente utilizadas en los cursos

Id	Herramientas más frecuentes*	Otras herramientas**
1	Entregas, Recursos, Foro	Test
2	Entregas, Recursos, Foro, Correo	Test
3	Entregas, Recursos, Foro	Test
4	Recursos, Foro	Blog
5	Recursos, Foro	Blog
6	Entregas, Foro***	-
7	Entregas, Foro***	-
8	Entregas, Foro, Test***	-
9	Entregas, Foro, Test***	-
10	Entregas, Foro, Test***	-
11	Entregas, Foro***	-
12	Entregas, Foro***	-
13	Recursos, Foro, Test	-
14	Recursos, Foro, Test	Blog
15	Recursos, Foro	Blog
16	Recursos, Foro	Blog
17	Recursos, Foro, Test	Blog
18	Recursos, Foro, Test	Blog
19	Recursos, Test, Glosario	Foro, Blog
20	Recursos, Foro	-
21	Recursos, Foro	Blog
22	Recursos, Foro	Test, Data
23	Recursos, Foro	Blog, Test, Wiki
24	Recursos, Foro	Blog, Test
25	Recursos, Foro, Test	-

* Herramientas más frecuentemente utilizadas en el curso.

** Otras herramientas con actividad considerable por parte de los estudiantes.

*** La actividad que se tiene de estos cursos es únicamente la que los profesores de otras universidades proporcionaron.

Bibliografía

[1] B. Frenay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, May 2014.

[2] J. Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 2005.

[3] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–54, 1996.

[4] S. Negash. Business intelligence. *The communications of the Association for Information Systems*, 13(1):117–195, 2004.

[5] M. A. Beyer and D. Laney. *Big Data Now*. O'Reilly, 2012.

[6] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[7] M. J. Berry and G. Linoff. *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc., 1997.

[8] D. Zhang and L. Zhou. Discovering golden nuggets: data mining in financial application. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(4):513–522, 2004.

[9] J. F. Roddick and P. Fule. Exploratory medical knowledge discovery: Experiences and issues. *SIGKDD Exploration*, 5:94–99, 2003.

[10] A. Peña Ayala. Review: Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, 41(4):1432–1462, March 2014.

[11] R. Espinosa, L. Garriga, J. J. Zubcoff, and J. N. Mazón. Linked open data mining for democratization of big data. In *IEEE International Conference on Big Data (Big Data)*, pages 17–19, 2014.

[12] C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618, 2010.

[13] I. Ros Martínez de Lahidalga. Moodle, la plataforma para la enseñanza y organización escolar, 2008. URL http://www.ehu.eus/ikastorratza/2_alea/moodle.pdf. Último acceso: marzo 2016.

[14] P. Bradford, M. Porciello, N. Balkon, and D. Backus. The blackboard learning system: The be all and end all in educational instruction? *Journal of Educational Technology Systems*, 35(3):301–314, 2007.

[15] C. Romero, S.n Ventura, M. Pechenizkiy, and R. S.J.d. Baker. *Handbook of educational data mining*. CRC Press, 2010.

[16] M. Pechenizkiy and J. Stamper. International educational data mining society, 2011. URL http://www.educationaldatamining.org/. Último acceso: marzo 2016.

[17] E. García, C. Romero, S. Ventura, and C. de Castro. A collaborative educational association rule mining tool. *The Internet and Higher Education*, 14(2):77–88, 2011.

[18] A. Hershkovitz and R. Nachmias. Developing a log-based motivation measuring tool. In *1st International Conference on Educational Data Mining*, pages 226–233, 2008.

[19] M. K. Singley and R. B. Lam. The classroom sentinel: supporting data-driven decision-making in the classroom. In *14th international conference on World Wide Web*, pages 315–321. ACM, 2005.

[20] A. Merceron and K. Yacef. Educational data mining: a case study. In *12th International Conference on Artificial Intelligence in Education*, pages 467–474, 2005.

[21] B. Kamsu-Foguem, G. Tchuenté-Foguem, L. Allart, Y. Zennir, C. Vilhelm, H. Mehdaoui, D. Zitouni, H. Hubert, M. Lemdani, and P. Ravaux. User-centered visual

analysis using a hybrid reasoning architecture for intensive care units. *Decision Support Systems*, 54(1):496–509, 2012.

[22] S. Klenk, J. Dippon, P. Fritz, and G. Heidemann. Interactive survival analysis with the ocdm system: From development to application. *Information Systems Frontiers*, 11(4):391–403, 2009.

[23] M. E. Zorrilla and D. García-Saiz. A service oriented architecture to provide data mining services for non-expert data miners. *Decision Support Systems*, 55 (1):399–411, 2013.

[24] K. Channabasavaiah, K. Holley, and E. Tuggle. Migrating to a service-oriented architecture, 2004. URL ftp://service.boulder.ibm.com/s390/audio/pdfs/G224-7298-00_FinalMigratetoSOA.pdf. Último acceso: marzo 2016.

[25] M. Zorrilla and D. García. A data mining service to assist instructors involved in virtual education. In *Business Intelligence Applications and the Web: Models, Systems and Technologies*, pages 222–243. 2011.

[26] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[27] A. Rashid, J. Royer, and A. Rummler. *Aspect-oriented, model-driven software product lines: The AMPLE way*. Cambridge University Press, 2011.

[28] G.Piatetsky. CRISP-DM, still the top methodology for analytics, data mining, or data science projects, 2014. URL http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html. Último acceso: marzo 2016.

[29] R. Wirth and J. Hipp. CRISP-DM: Towards a standard process model for data mining. In *4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, pages 29–39, 2000.

[30] D. Garcıa-Saiz and M. Zorrilla. Comparing classification methods for predicting distance students' performance. In *Proceedings of the International Workshop on Applications of Pattern Analysis*, pages 26–32, 2011.

[31] D. García-Saiz and M. E. Zorrilla. E-learning web miner: A data mining application to help instructors involved in virtual courses. In *4th International Conference on Educational Data Mining*, pages 323–324, 2011.

[32] M. Zorilla, D. García-Saiz, and J. Balcázar. Towards parameter-free data mining: Mining educational data with yacaree. In *4th International Conference on Educational Data Mining*, pages 363–364, 2011.

[33] J. Luis Balcázar, D. García-Sáiz, J. de la Dehesa, et al. Iterator-based algorithms in self-tuning discovery of partial implications. In *International Conference on Formal Concept Analysis*, pages 14–28, 2012.

[34] D. García-Sáiz, M. Zorrilla, J. Balcázar, et al. Closures and partial implications in educational data mining. In *International Conference on Formal Concept Analysis*, pages 98–113, 2012.

[35] D. García-Saiz and M. Zorrilla. A promising classification method for predicting distance students' performance. In *5th International Conference on Educational Data Mining*, pages 206–207, 2012.

[36] P. Sanchez, D. Garcia-Saiz, and M. Zorrilla. Software product line engineering for e-learning applications: A case study. In *International Symposium on Computers in Education (SIIE)*, pages 1–6. IEEE, 2012.

[37] R. Espinosa, D. García-Saiz, J. J. Zubcoff, J. Mazón, and M. Zorrilla. Towards the development of a knowledge base for realizing user-friendly data mining. In *6th International Conference on Metadata and Semantics Research*, pages 121–126. 2012.

[38] D. Garcia-Saiz and M. Zorrilla. Towards the development of a classification service for predicting students' performance. In *6th International Conference on Educational Data Mining*, pages 318–319, 2013.

[39] R. Espinosa, D. García-Saiz, M. Zorrilla, J. J. Zubcoff, and J. Mazón. Development of a knowledge base for enabling non-expert users to apply data mining algorithms. In *3th International Symposium on Data-driven Process Discovery and Analysis*, pages 46–61, 2013.

[40] C. Palazuelos, D. García-Saiz, and M. Zorrilla. Social network analysis and data mining: an application to the e-learning context. In *5th International Conference on Computational Collective Intelligence, Technologies and Applications*, pages 651–660. Springer, 2013.

[41] M. Zorrilla and D. García-Saiz. A service oriented architecture to provide data mining services for non-expert data miners. *Decision Support Systems*, 55(1):399–411, 2013.

[42] D. García-Saiz, C. Palazuelos, and M. Zorrilla. *Data Mining and Social Network Analysis in the Educational Field: An Application for Non-Expert Users*, pages 411–439. Springer, 2014.

[43] P. Sanchez Barreiro, D. Garcia-Saiz, and M. Zorrilla Pantaleon. Building families of software products for e-learning platforms: A case study. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 9(2):64–71, 2014.

[44] M. Zorrilla and D. García-Saiz. Meta-learning: Can it be suitable to automatise the kdd process for the educational domain? In *2nd International Conference on Rough Sets and Intelligent Systems Paradigms*, pages 285–292. Springer, 2014.

[45] D. García-Saiz, C. Palazuelos, and M. Zorrilla. The predictive power of the SNA metrics for education. In *7th International Conference on Educational Data Mining*, pages 419–420, 2014.

[46] A. de la Vega, D. García-Saiz, M. Zorrilla, and P. Sánchez. Domain specific languages for data mining: A case study for educational data mining. In *4th International Symposium on Languages, Applications and Technologies*, 2015.

[47] A. de la Vega, D. García-Saiz, M. Zorrilla, and P. Sánchez. Towards a dsl for educational data mining. In J. Sierra-Rodríguez, J. Leal, and A. Simões, editors, *Languages, Applications and Technologies*, volume 563 of *Communications in Computer and Information Science*, pages 79–90. Springer International Publishing, 2015.

[48] D. García-Saiz and M. Zorrilla. Detection of learners with a performance inconsistent with their effort. In *8th International Conference on Educational Data Mining*, pages 606–607, 2015.

[49] M. Zorrilla and D. García-Saiz. Meta-learning based framework for helping non-expert miners to choice a suitable classification algorithm: An application for the educational field. In *7th International Conference on Computational Collective Intelligence*, pages 431–440. Springer, 2015.

[50] D. García-Saiz, R. Espinosa, M. Zorrilla, J. J. Zubcoff, and J. Mazón. Un marco para democratizar la minería de datos: propuesta inicial y retos. In *XX Jornadas de Ingeniería de Software y Bases de Datos*, 2015.

[51] M. Zorrilla, E. Álvarez, and D. García-Saiz. A parametrisable method for measuring online attendance in e-learning tools. *International Journal of Technology Enhanced Learning*, 7(4):289–308, 2015.

[52] R. Espinosa, D. García-Saiz, M. Zorrilla, J. J. Zubcoff, and J. Mazón. Enabling non-expert users to apply data mining for bridging the big data divide. In *Lecture Notes in Business Information Processing: Data-Driven Process Discovery and Analysis*, pages 65–86. Springer, 2015.

[53] D. Garcıa-Saiz and M. Zorrilla. Metalearning-based recommenders: towards automatic classification algorithm selection. In *Conferencia de la Asociación Española para la Inteligencia Artificial*, pages 749–758, 2015.

[54] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11 (1):10–18, 2009.

[55] J. Luan. Data mining and its applications in higher education. *New Directions for Institutional Research*, 2002(113):17–36, 2002.

[56] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1):135–146, 2007.

[57] A. Drigas and J. Vrettaros. An intelligent tool for building e-learning contend-material using natural language in digital libraries. *WSEAS Transactions on Information Science and Applications*, 1(5):1197–1205, 2004.

[58] K. Hammouda and M. Kamel. Data mining in e-learning. In *E-Learning Networked Environments and Architectures*, pages 374–404. Springer, 2007.

[59] J. Tane, C. Schmitz, and G. Stumme. Semantic resource management for the web: An e-learning application. In *13th International World Wide Web Conference on Alternate Track Papers*, pages 1–10, New York, NY, USA, 2004. ACM.

[60] Z. Su, W. Song, M. Lin, and J. Li. Web text clustering for personalized e-learning based on maximal frequent itemsets. In *International Conference on Computer Science and Software Engineering*, volume 6, pages 452–455, 2008.

[61] F. Castro, A. Vellido, A. Nebot, and J. Minguillón. Detecting atypical student behaviour on an e-learning system. In *I Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación*, pages 153–160, 2005.

[62] F. Castro, A. Vellido, A. Nebot, and J. Minguillón. Detecting atypical student behaviour on an e-learning system. In *VI Congreso Nacional de Informática Educativa Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación*, pages 153–160, España, 2005.

[63] T. Tang and G. McCalla. Smart recommendation for an evolving e-learning system: Architecture and experiment. *International Journal on E-Learning*, 4(1):105–129, 2005.

[64] G. McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education*, 2004(1):1–23, 2010.

[65] H. Fu and M. OFoghlu. A conceptual subspace clustering algorithm in e-learning. In *10th International Conference on Advanced Communication Technology*, volume 3, pages 1983–1988, 2008.

[66] C. Chan. A framework for assessing usage of web-based e-learning systems. In *2nd International Conference on Innovative Computing, Information and Control*, pages 147–147, Sept 2007.

[67] P.B. Myszkowski, H. Kwasnicka, and U. Markowska-Kaczmar. Data mining techniques in e-learning celgrid system. In *7th International Conference on Computer Information Systems and Industrial Management Applications*, pages 315–319, June 2008.

[68] P.B. Myszkowski, H. Kwasnicka, and U. Markowska-Kaczmar. Data mining techniques in e-learning celgrid system. In *7th Conference on Computer Information Systems and Industrial Management Applications*, pages 315–319, June 2008.

[69] Z. Su, W. Song, M. Lin, and J. Li. Web text clustering for personalized e-learning based on maximal frequent itemsets. In *International Conference on Computer Science and Software Engineering*, volume 6, pages 452–455, Dec 2008.

[70] Duncan Mullier. A tutorial supervisor for automatic assessment in educational systems. *International Journal on E-Learning*, 2(1):37–49, 2003.

[71] E. W. Forgy. Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics*, 21:768–769, 1965.

[72] K. Hammouda and M. Kamel. Data mining in e-learning. In S. Pierre, editor, *E-Learning Networked Environments and Architectures*, Advanced Information and Knowledge Processing, pages 374–404. Springer London, 2007.

[73] G. Hwang. A test-sheet-generating algorithm for multiple assessment requirements. *IEEE Transactions on Education*, 46(3):329–337, 2003.

[74] A. Vellido. Assessment of an unsupervised feature selection method for generative topographic mapping. In *Artificial Neural Networks*, volume 4132 of *Lecture Notes in Computer Science*, pages 361–370. Springer Berlin Heidelberg, 2006.

[75] M. M. Sacks, M. E. S. Mendes, E. Martinez, and L. Sacks. Knowledge-based content navigation in e-learning applications. In *London Communications Symposium*, 2002.

[76] A. Vellido, F. Castro, A. Nebot, and F. Mugica. Characterization of atypical virtual campus usage behavior through robust generative relevance analysis. In *5th International Conference on Web-based Education*, pages 183–188, 2006.

[77] C. Teng, C. Lin, S. Cheng, and J. Heh. Analyzing user behavior distribution on e-learning platform with techniques of clustering. In *Proceedings of Society for Information Technology & Teacher Education International Conference*, pages 3052–3058, 2004.

[78] S. Eschrich, J. Ke, L. O. Hall, and D. B. Goldgof. Fast accurate fuzzy clustering through data. *IEEE Transactions on Fuzzy Systems*, 11:262–270, 2003.

[79] P. Yua, C. Own, , and L. Lin. On learning behavior analysis of web based interactive environment. In *International Conference on Computer and Electrical Engineering*, pages 1–9, 2001.

[80] J. Freyberger, N. T. Heffernan, and C. Ruiz. Using association rules to guide a search for best fitting transfer models of student learning. In *Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, pages 1–10, 2004.

[81] C. Romero, A. Zafra, J. M. Luna, and S. Ventura. Association rule mining using genetic programming to provide feedback to instructors from multiple-choice quiz data. *Expert Systems*, 30(2):162–172, 2013.

[82] W. Hämäläinen and M. Vinni. *Handbook of Educational Data Mining*, chapter Classifiers for Educational Data Mining, pages 57–74. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press, 2010.

[83] J. Superby, J.P. Vandamme, and N. Meskens. Determination of factors influencing the achievement of the first-year university students using data mining methods. In *Workshop on Educational Data Mining*, pages 37–44, 2006.

[84] S. Herzog. Estimating student retention and degree-completion time: Decision trees and neural networks vis-à-vis regression. *New Directions for Institutional Research*, 2006(131):17–33, 2006.

[85] N. T. Nghe, P. Janecek, and P. Haddawy. A comparative analysis of techniques for predicting academic performance. In *37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, pages 7–12, 2007.

[86] Y. Ma, B. Liu, C. K. Wong, P. S. Yu, and S. M. Lee. Targeting the right students using data mining. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 457–464, 2000.

[87] W. Hämäläinen and M. Vinni. Comparison of machine learning methods for intelligent tutoring systems. In *Intelligent Tutoring Systems*, volume 4053 of *Lecture Notes in Computer Science*, pages 525–534. Springer Berlin Heidelberg, 2006.

[88] W. Zang and F. Lin. Investigation of web-based teaching and learning by boosting algorithms. In *International Conference on Information Technology: Research and Education*, pages 445–449, Aug 2003.

[89] S.B. Kotsiantis, C.J. Pierrakeas, and P.E. Pintelas. Preventing student dropout in distance learning using machine learning techniques. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 2774 of *Lecture Notes in Computer Science*, pages 267–274. Springer Berlin Heidelberg, 2003.

[90] B. Minaei-Bidgoli, D.A. Kashy, G. Kortemeyer, and W.F. Punch. Predicting student performance: an application of data mining methods with an educational web-based system. In *33rd Annual Conference on Frontiers in Education*, volume 1, pages 2–13, Nov 2003.

[91] C.C. Liu. *Knowledge discovery from web portfolios: tools for learning performance assessment*. PhD thesis, 2000.

[92] M. Cocea and S. Weibelzahl. Can log files analysis estimate learners‘ level of motivation?. In *14th Workshop on Adaptivity and User Modeling in Interactive Systems*, volume 1/2006, pages 32–35, 2006.

[93] M. Lee. Profiling students' adaptation styles in web-based learning. *Computers & Education*, 36(2):121 – 132, 2001.

[94] M. Damez, T. Ha Dang, C. Marsala, and B. Bouchon-Meunier. Fuzzy Decision Tree for User Modeling from Human-Computer Interactions. In *5th International Conference on Human System Learning)*, pages 287–302, 2005.

[95] T. Hurley and S. Weibelzahl. Eliciting adaptation knowledge from on-line tutors to increase motivation. In *User Modeling*, volume 4511 of *Lecture Notes in Computer Science*, pages 370–374. Springer Berlin Heidelberg, 2007.

[96] I. A. Khan and J. T. Choi. An application of educational data mining (edm) technique for scholarship prediction. *International Journal of Software Engineering and its Application*, 8(12):31–42, 2014.

[97] S.A. Abaya and B.D. Gerardo. An education data mining tool for marketing based on c4.5 classification technique. In *2nd International Conference on e-Learning and e-Technologies in Education (ICEEE)*, pages 289–293, Sept 2013.

[98] D. F. Butcher and W. A. Muth. Predicting performance in an introductory computer science course. *Commun. ACM*, 28(3):263–268, March 1985.

[99] L.V. Fausett and W. Elwasif. Predicting performance from test scores using backpropagation and counterpropagation. In *IEEE International Conference on Neural Networks*, volume 5, pages 3398–3402 vol.5, 1994.

[100] S. Kotsiantis, C. Pierrakeas, and P. Pintelas. Predicting stundetns' performance in distance learning using machine learning techniques. *Applied Artificial Intelligence*, 18(5):411–426, 2004.

[101] B. Minaei-Bidgoli and W. F. Punch. Using genetic algorithms for data mining optimization in an educational web-based system. In *International Conference on Genetic and Evolutionary Computation: PartII*, pages 2252–2263, 2003.

[102] P.L. Hsu, R. Lai, and C.C. Chiu. The hybrid of association rule algorithms and genetic algorithms for tree induction: an example of predicting the student course performance. *Expert Systems with Applications*, 25(1):51 – 62, 2003.

[103] M. D. Calvo-Flores, E. G. Galindo, M. C. Pegalajar Jiménez, and O. Pérez Pineiro. Predicting students' marks from moodle logs using neural network models. *Current Developments in Technology-Assisted Education*, 1:586–590, 2006.

[104] S. T. Karamouzis and A. Vrettos. An artificial neural network for predicting student graduation outcomes. In *World Congress on Engineering and Computer Science*, pages 991–994, 2008.

[105] G. W. Dekker, M. Pechenizkiy, and J. M. Vleeshouwers. Predicting students drop out: A case study. *International Working Group on Educational Data Mining*, pages 41–50, 2009.

[106] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, and V. Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education*, 53(3):950 – 965, 2009.

[107] S. Brown and B. Forouraghi. Concept classification using a hybrid data mining model. In *21st International Conference on Tools with Artificial Intelligence*, pages 375–378, 2009.

[108] S. Kotsiantis, K. Patriarcheas, and M. Xenos. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems*, 23(6):529 – 535, 2010.

[109] C. Márquez-Vera, C. Romero, and S. Ventura. Predicting school failure using data mining. In *4th International Conference on Educational Data Mining*, pages 271–276, 2011.

[110] M. Koutina and K. Kermanidis. Predicting postgraduate students' performance using machine learning techniques. In *Artificial Intelligence Applications and Innovations*, volume 364 of *IFIP Advances in Information and Communication Technology*, pages 159–168. Springer Berlin Heidelberg, 2011.

[111] M. E. Alper and Z. Çataltepe. Improving course success prediction using abet course outcomes and grades. In *International Conference on Computer Supported Education*, pages 222–229. SciTePress, 2012.

[112] S.B. Kotsiantis. Use of machine learning techniques for educational proposes: a decision support system for forecasting students' grades. *Artificial Intelligence Review*, 37(4):331–344, 2012.

[113] K. Turhan, B. Kurt, and Y. Z. Engin. Estimation of student success with artificial neural networks. *Education and Science*, 38(170), 2013.

[114] S. Huang and N. Fang. Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models. *Computers & Education*, 61(0):133 – 145, 2013.

[115] V. Ribeiro De Carvalho Martinho, C. Nunes, and C.R. Minussi. An intelligent system for prediction of school dropout risk group in higher education classroom based on artificial neural networks. In *25th International Conference on Tools with Artificial Intelligence)*, pages 159–166, Nov 2013.

[116] V.R.C. Martinho, C. Nunes, and C.R. Minussi. Prediction of school dropout risk group using neural network. In *Federated Conference on Computer Science and Information Systems*, pages 111–114, Sept 2013.

[117] Q. Hung Do and J. Chen. A neuro-fuzzy approach in the classification of students' academic performance. *Computational Intelligence and Neuroscience*, 2013: 7, 2013.

[118] J. Kasih and S. Ayub, M.and Susanto. Predicting students' final passing results using the classification and regression trees (cart) algorithm. *World Transactions on Engineering and Technology Education*, 11(1):46–49, 2013.

[119] P. Strecht, L. Cruz, C. Soares, J. Mendes-Moreira, and R. Abreu. A comparative study of classification and regression algorithms for modelling students' academic performance. In *8th International Conference on Educational Data Mining*, pages 292–295, 2015.

[120] I. Hidayah, A. E. Permanasari, and N. Ratwastuti. Student classification for academic performance prediction using neuro fuzzy in a conventional classroom. In *International Conference on Information Technology and Electrical Engineering*, pages 221–225, 2013.

[121] N.M. Norwawi, S.F. Abdusalam, C.F. Hibadullah, and B.M. Shuaibu. Classification of students' performance in computer programming course according to learning style. In *2nd Conference on Data Mining and Optimization*, pages 37–41, Oct 2009.

[122] R.M. Felder and B.A. Soloman. Index of learning styles, 1991. URL http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/styles.htm.

[123] L. Lykourentzou, L. Giannoukos, G. Mpardis, V. Nikolopoulos, and V. Loumos. Early and dynamic student achievement prediction in e-learning courses using neural networks. *Journal of the American Society for Information Science and Technology*, 60(2):372–380, 2009.

[124] P. G. Romero, C .and Espejo, A. Zafra, J. R. Romero, and S. Ventura. Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146, 2013.

[125] C. Romero, S. Ventura, P. G. Espejo, and C.Hervás. Data mining algorithms to classify students. In *International Conference on Educational Data Mining*, pages 8–17, 2008.

[126] C. Romero, S. Ventura, and E. García. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368 – 384, 2008.

[127] A. Zafra and S. Ventura. Predicting student grades in learning management systems with multiple instance genetic programming. In *International Working Group on Educational Data Mining*, number 1, page 8, 2009.

[128] A. Zafra and S. Ventura. Multi-instance genetic programming for predicting student performance in web based educational environments. *Applied Soft Computing*, 12(8):2693 – 2706, 2012.

[129] J. Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.

[130] P. A. Whigham. *Grammatical Bias for Evolutionary Learning*. PhD thesis, New South Wales, Australia, Australia, 1996.

[131] A. Zafra, C. Romero, and S. Ventura. Multiple instance learning for classifying students in learning management systems. *Expert Systems with Applications*, 38 (12):15020 – 15031, 2011.

[132] J.M. Luna, C. Romero, J.R. Romero, and S. Ventura. An evolutionary algorithm for the discovery of rare class association rules in learning management systems. *Applied Intelligence*, pages 1–13, 2014.

[133] S. E. Sorour, T. Mine, K. Goda, and S. Hirokawa. Predicting students' grades based on free style comments data by artificial neural network. In *Frontiers in Education Conference*, pages 1–9, Oct 2014.

[134] J. Luo, E. Sorour, K. Goda, and T. Mine. Predicting student grade based on free-style comments using Word2Vec and ANN by considering prediction results obtained in consecutive lessons. pages 396–399, 2015.

[135] C. Romero, M. López, J. Luna, and S. Ventura. Predicting students' final performance from participation in on-line discussion forums. *Computers and Education*, 68(0):458 – 472, 2013.

[136] M. I. López, J. M. Luna, C. Romero, and S. Ventura. Classification via clustering for predicting final marks based on student participation in forums. In *International Conference on Educational Data Mining*, page 4, Jun 2012.

[137] M. M. Molina, J. M. Luna, C. Romero, and S. Ventura. Meta-learning approach for automatic parameter tuning: A case study with educational datasets. pages 180–183, 2012.

[138] G. Kortemeyer, W. Bauer, D. Kashy, E. Kashy, and C. Speier. The Learning Online Network with capa initiative. In *Frontiers in Education Conference*, pages 2–23. IEEE, 2001.

[139] S. B. Kotsiantis and P. E. Pintelas. Comparing regression algorithms for predicting students' marks in hellenic open university.

[140] Z. A. Pardos, Q. Y. Wang, and S. Trivedi. The real world significance of performance prediction. In *International Conference on Educational Data Mining*, pages 192–195, 2012.

[141] S. Trivedi, Z. A. Pardos, and N. T. Heffernan. Clustering students to generate an ensemble to improve standard test score predictions. In *Artificial Intelligence in Education*, volume 6738 of *Lecture Notes in Computer Science*, pages 377–384. Springer Berlin Heidelberg, 2011.

[142] W. Xing, R. Guo, E. Petakovic, and S. Goggins. Participation-based student final performance prediction model through interpretable genetic programming: Integrating learning analytics, educational data mining and theory. *Computers in Human Behavior*, 47:168 – 181, 2015.

[143] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

[144] C. Lemke, M. Budka, and B. Gabrys. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, 44(1):117–130, 2013.

[145] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[146] J. Gama and P. Brazdil. Cascade generalization. *Machine Learning*, 41(3):315–343, December 2000.

[147] J. R. Rice. The algorithm selection problem. Technical report, 1975.

[148] D. Michie, D. J. Spiegelhalter, and C.C. Taylor. *Machine Learning, Neural and Statistical Classification*. 1994. URL http://www1.maths.leeds.ac.uk/~charles/statlog/. Último acceso: marzo 2016.

[149] R. D. King, C. Feng, and A. Sutherland. Statlog: Comparison of classification algorithms on large real-world problems, 1995.

[150] G. Lindner and R. Studer. Ast: Support for algorithm selection with a cbr approach. In *Principles of Data Mining and Knowledge Discovery*, volume 1704 of *Lecture Notes in Computer Science*, pages 418–423. Springer Berlin Heidelberg, 1999.

[151] C. Köpf, C. Taylor, and J. Keller. Meta-analysis: from data characterization for meta-learning to meta-regression. In *Workshop on Data Mining, Decision Support, Meta-Learning and ILP*, 2000.

[152] K. A. Smith, F. Woo, V. Ciesielski, and R. Ibrahim. Modelling the relationship between problem characteristics and data mining algorithm performance using neural networks, in: C. dagli, et al. In *Eds.), Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems*, pages 357–362, 2001.

[153] S. Cacoveanu, C. Vidrighin, and R. Potolea. Evolutional meta-learning framework for automatic classifier selection. In *IEEE 5th International Conference on Intelligent Computer Communication and Processing*, pages 27–30, Aug 2009.

[154] S. Ali and K. A. Smith. On learning algorithm selection for classification. *Applied Soft Computing*, 6(2):119–138, 2006.

[155] C. Soares and P. B. Brazdil. Zoomed ranking: Selection of classification algorithms based on relevant performance information. In *Principles of Data Mining and Knowledge Discovery*, volume 1910 of *Lecture Notes in Computer Science*, pages 126–135. Springer Berlin Heidelberg, 2000.

[156] Y. Peng, P. A Flach, P. Brazdil, and C. Soares. Decision tree-based data characterization for meta-learning. In *2nd Workshop on Integration and Collaboration Aspects of Data Mining*, pages 111–122, 2002.

[157] Y. Peng, P. A. Flach, C. Soares, and P. Brazdil. Improved dataset characterisation for meta-learning. In S. Lange, K. Satoh, and C. H. Smith, editors, *Discovery Science*, volume 2534 of *Lecture Notes in Computer Science*, pages 141–152. Springer Berlin Heidelberg, 2002.

[158] P. B. Brazdil, C. Soares, and J. P. Da Costa. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, 50(3): 251–277, 2003.

[159] M. Reif, F. Shafait, M. Goldstein, T. Breuel, and A. Dengel. Automatic classifier selection for non-experts. *Pattern Analysis and Applications*, 17(1):83–96, 2014.

[160] A. Kalousis and T. Theoharis. Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis*, 3(5):319 – 337, 1999.

[161] A. Kalousis and M. Hilario. Model selection via meta-learning: a comparative study. In *12th IEEE International Conference on Tools with Artificial Intelligence*, pages 406–413, 2000.

[162] A. Kalousis and M. Hilario. Feature selection for meta-learning. In *Advances in Knowledge Discovery and Data Mining*, volume 2035 of *Lecture Notes in Computer Science*, pages 222–233. Springer Berlin Heidelberg, 2001.

[163] S. Zeroski, L. Todorovski, and H. Blockeel. Relational ranking with predictive clustering trees. In *In Proceedings of the 13th European Conference on Machine Learning*, pages 444–456. SpringerVerlag, 2002.

[164] L. Todorovski, H. Blockeel, and S. Dzeroski. Ranking with predictive clustering trees. In *Machine Learning: ECML 2002*, volume 2430 of *Lecture Notes in Computer Science*, pages 444–455. Springer Berlin Heidelberg, 2002.

[165] G. Wang, Q. Song, X. Zhang, and K. Zhang. A generic multilabel learning-based classification algorithm recommendation method. *ACM Transactions on Knowledge Discovery from Data*, 9(1):1–30, October 2014.

[166] M. Zhang and Z. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038 – 2048, 2007.

[167] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *7th International Conference on Machine Learning*, pages 743–750. Morgan Kaufmann, 2000.

[168] J. Fürnkranz and J. Petrak. An evaluation of landmarking variants. In *Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, pages 57–68, 2001.

[169] H. Bensusan and C. Giraud-Carrier. Discovering task neighbourhoods through landmark learning performances. In D. A. Zighed, J. Komorowski, and J. Żytkow, editors, *Principles of Data Mining and Knowledge Discovery*, volume 1910 of *Lecture Notes in Computer Science*, pages 325–330. Springer Berlin Heidelberg, 2000.

[170] A. Orriols-Puig and N. Macià. Data complexity library in c++, 2015. URL http://dcol.sourceforge.net/. Último acceso: marzo 2016.

[171] J. Luengo and F. Herrera. An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. *Knowledge and Information Systems*, 42(1):147–180, 2015.

[172] J. W. Lee and C. Giraud-Carrier. Predicting algorithm accuracy with a small set of effective meta-features. In *7th International Conference on Machine Learning and Applications*, pages 808–812, Dec 2008.

[173] P. Ridd and C. Giraud-Carrier. Using metalearning to predict when parameter optimization is likely to improve classification accuracy. In *21st European Conference on Artificial Intelligence*, pages 18–23, 2014.

[174] M. Reif, F. Shafait, and A. Dengel. Meta-learning for evolutionary parameter optimization of classifiers. *Machine Learning*, 87(3):357–380, 2012.

[175] P.B.C. de Miranda, R.B.C. Prudencio, A.C.P.L.F. Carvalho, and C. Soares. Combining a multi-objective optimization approach with meta-learning for SVM parameter selection. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 2909–2914, 2012.

[176] P.B. C. Miranda, R. B. C. Prudencio, A. P. L. F. Carvalho, and C. Soares. Combining meta-learning with multi-objective particle swarm algorithms for SVM parameter selection: An experimental analysis. In *Brazilian Symposium on Neural Networks*, pages 1–6. IEEE Computer Society, 2012.

[177] P. B.C. Miranda, R. B.C. Prudêncio, A. P.L.F. de Carvalho, and C. Soares. A hybrid meta-learning architecture for multi-objective optimization of SVM parameters. *Neurocomputing*, 143(0):27 – 43, 2014.

[178] T. A.F. Gomes, R. B.C. Prudêncio, C. Soares, A. L.D. Rossi, and A. Carvalho. Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing*, 75(1):3 – 13, 2012.

[179] D. G. Ferrari and L. N. de Castro. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. *Information Sciences*, page 181–194, 2015.

[180] C. Soares. UCI++: Improved support for algorithm selection using datasetoids. In *Advances in Knowledge Discovery and Data Mining*, volume 5476 of *Lecture Notes in Computer Science*, pages 499–506. Springer Berlin Heidelberg, 2009.

[181] R. B. C. Prudêncio, C. Soares, and T. B. Ludermir. Combining meta-learning and active selection of datasetoids for algorithm selection. In *Hybrid Artificial Intelligent Systems*, volume 6678 of *Lecture Notes in Computer Science*, pages 164–171. Springer Berlin Heidelberg, 2011.

[182] R. B. C. Prudencio and T. B. Ludermir. Active selection of training examples for meta-learning. In *7th International Conference on Hybrid Intelligent Systems*, pages 126–131. IEEE, 2007.

[183] R. Prudêncio and T. Ludermir. Active learning to support the generation of meta-examples. In *Artificial Neural Networks*, pages 817–826. Springer, 2007.

[184] R. B. C. Prudêncio and T. B. Ludermir. Selective generation of training examples in active meta-learning. *International Journal of Hybrid Intelligent Systems*, 5(2): 59, 2008.

[185] S. D. Abdelmessih, F. Shafait, M. Reif, and M. Goldstein. Landmarking for meta-learning using rapidminer. Technical report, 2010.

[186] R. B.C. Prudêncio and T. B. Ludermira. Combining uncertainty sampling methods for supporting the generation of meta-examples. *Information Sciences*, 196:1 – 14, 2012.

[187] F. Akthar and C. Hahne. Rapidminer 5 operator reference, 2012. URL http://rapidminer.com/. Último acceso: marzo 2016.

[188] H. Bensusan and A. Kalousis. Estimating the predictive accuracy of a classifier. In *Machine Learning*, volume 2167 of *Lecture Notes in Computer Science*, pages 25–36. Springer Berlin Heidelberg, 2001.

[189] K. A. Smith, F. Woo, V. Ciesielski, and R. Ibrahim. Matching data mining algorithm suitability to data characteristics using a self-organizing map. In *Hybrid Information Systems*, volume 14 of *Advances in Soft Computing*, pages 169–179. Physica-Verlag HD, 2002.

[190] D. H. Wolpert. The supervised learning no-free-lunch theorems. In *Soft Computing and Industry*, pages 25–42. Springer, 2002.

[191] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002.

[192] T. K. Ho, M. Basu, and M. H. C. Law. Measures of geometrical complexity in classification problems. In *Data complexity in pattern recognition*, pages 1–23. Springer, 2006.

[193] Commons Math: The Apache Commons Mathematics Library. URL https://commons.apache.org/. Último acceso: marzo 2016.

[194] Y. Yang and G.I. Webb. Proportional k-interval discretization for naive-bayes classifiers. In *European Conference on Machine Learning*, pages 564–575. Springer, 2001.

[195] M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

[196] R. Kohavi and G. H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[197] N.V. Chawla, K. W. Bowyer, L.O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, pages 321–357, 2002.

[198] D.M. Hawkins. *Identification of Outliers*. Monographs on Applied Probability and Statistics. Chapman and Hall, 1980.

[199] J. Han, M. Kamber, and J. Pei. Outlier detection. In ElServier, editor, *Data Mining: Concepts and Techniques*, pages 543–584. Morgan Kaufmann, 2012.

[200] A.J. Tallon-Ballesteros and J.C. Riquelme. Deleting or keeping outliers for classifier training? In *6th World Congress on Nature and Biologically Inspired Computing*, pages 281–286, July 2014.

[201] A. Foss and O. R. Zaïane. Class separation through variance: a new application of outlier detection. *Knowledge Information Systems*, 29(3):565–596, 2011.

[202] M.M. Breunig, H. Kriegel, R.T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[203] M. K. Saad and N. M. Hewahi. A comparative study of outlier mining and class outlier mining. *Computer Science Letters*, 1(1), June 2009.

[204] N. Alaydie, F. Fotouhi, C.K. Reddy, and H. Soltanian-Zadeh. Noise and outlier filtering in heterogeneous medical data sources. In *Workshop on Database and Expert Systems Applications*, pages 115–119, 9 2010.

[205] T. Takacs and L. Vajta. Novel outlier filtering method for aoi image databases. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 2 (4):700 –709, april 2012.

[206] T.M. Padmaja, N. Dhulipalla, R.S. Bapi, and P.R. Krishna. Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection. In *International Conference on Advanced Computing and Communications*, pages 511 –516, dec. 2007.

[207] J. Zhang and M. Zulkernine. Anomaly based network intrusion detection with unsupervised outlier detection. In *IEEE International Conference on Communications*, volume 5, pages 2388 –2393, june 2006.

[208] J. Zhang, M. Zulkernine, and A. Haque. Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(5):649 –659, sept. 2008.

[209] W. Zhang, Q. Yang, and Y. Geng. A survey of anomaly detection methods in networks. In *International Symposium on Computer Network and Multimedia Technology*, pages 1 –3, 2009.

[210] M. Tavallaee, N. Stakhanova, and A.A. Ghorbani. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(5):516 –524, sept. 2010.

[211] Z. He, X. Xu, J. Z. Huang, and S. Deng. Mining class outliers: concepts, algorithms and applications in crm. *Expert Systems and Applications*, 27(4):681–697, 2004.

[212] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[213] Y. Freund. An adaptive version of the boost by majority algorithm. *Machine learning*, 43(3):293–318, 2001.

[214] G. Rätsch, T. Onoda, and K. Müller. Soft margins for adaboost. *Machine learning*, 42(3):287–320, 2001.

[215] P.S. Sastry, G.D. Nagendra, and N. Manwani. A team of continuous-action learning automata for noise-tolerant learning of half-spaces. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(1):19–28, 2010.

[216] J. Abellán and A. R. Masegosa. Bagging decision trees on data sets with classification noise. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 248–265. Springer, 2010.

[217] L. Joseph, T.W. Gyorkos, and L. Coupal. Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard. *American Journal of Epidemiology*, 141(3):263–272, 1995.

[218] A. Tenenbein. A double sampling scheme for estimating from binomial data with misclassifications. *Journal of the American Statistical Association*, 65(331):1350–1361, 1970.

[219] M. Ruiz, F.J. Girón, C.J. Pérez, J. Martín, and C. Rojano. A bayesian model for multinomial sampling with misclassified data. *Journal of Applied Statistics*, 35(4): 369–382, 2008.

[220] C. Bouveyron and S. Girard. Robust supervised classification with mixture models: Learning from data with uncertain labels. *Pattern Recognition*, 42(11):2649–2658, 2009.

[221] U. Rebbapragada and C. E. Brodley. Class noise mitigation through instance weighting. In *Machine Learning*, pages 708–715. Springer, 2007.

[222] E. Eskin. Detecting errors within a corpus using anomaly detection. In *1st North American chapter of the Association for Computational Linguistics conference*, pages 148–153. Association for Computational Linguistics, 2000.

[223] T. Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25(5):804–813, 1995.

[224] J. Sun, F. Zhao, C. Wang, and S. Chen. Identifying and correcting mislabeled training instances. In *International Conference on Future generation communication and networking*, volume 1, pages 244–250. IEEE, 2007.

[225] D. Gamberger, N. Lavrač, and S. Džeroski. Noise elimination in inductive concept learning: A case study in medical diagnosis. In *International Conference Algorithmic Learning Theory*, pages 199–212. Springer, 1996.

[226] J. Thongkam, G. Xu, Y. Zhang, and F. Huang. Support vector machine for outlier detection in breast cancer survivability prediction. In *International Workshops on Advanced Web and Network Technologies, and Applications*, pages 99–109. Springer, 2008.

[227] Carla E. Brodley and Mark A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, pages 131–167, 1999.

[228] X. Zhu and Q. Wu, X.and Chen. Eliminating class noise in large datasets. In *International Conference on Machine Learning*, volume 3, pages 920–927, 2003.

[229] T. R. Wilson, D. R.and Martinez. Reduction techniques for instance-based learning algorithms. *Machine learning*, 38(3):257–286, 2000.

[230] S. Verbaeten and A. Van Assche. Ensemble methods for noise elimination in classification problems. In *International Workshop on Multiple classifier systems*, pages 317–325. Springer, 2003.

[231] V. Wheway. Using boosting to detect noisy data. In *6th Pacific Rim International Conference on Artificial Intelligence*, pages 123–130. Springer, 2001.

[232] A. Malossini, E. Blanzieri, and R. T. Ng. Detecting potential labeling errors in microarrays by data perturbation. *Bioinformatics*, 22(17):2114–2121, 2006.

[233] J. S. Sánchez, F. Pla, and F. J. Ferri. Prototype selection for the nearest neighbour rule through proximity graphs. *Pattern Recognition Letters*, 18(6):507–513, 1997.

[234] R. Khardon and G. Wachman. Noise tolerant variants of the perceptron algorithm. *The journal of machine learning research*, 8:227–248, 2007.

[235] A. Kowalczyk, A. J. Smola, and R. C Williamson. Kernel machines and boolean functions. In *Annual Conference on Advances in Neural Information Processing Systems*, volume 1, page 439. The MIT Press, 2002.

[236] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[237] P. Clark and T. Niblett. The cn2 induction algorithm. *Machine learning*, 3(4): 261–283, 1989.

[238] C. Domingo and O. Watanabe. Madaboost: A modification of adaboost. In *Proceedings of the 30th Annual Conference on Computational Learning Theory*, pages 180–189, 2000.

[239] N. C. Oza. Boosting with averaged weight vectors. In *International Workshop on Multiple Classifier Systems*, pages 15–24. Springer, 2003.

[240] N. C. Oza. Aveboost2: Boosting for noisy data. In *International Workshop on Multiple Classifier Systems*, pages 31–40. Springer, 2004.

[241] A. Krieger, C. Long, and A. Wyner. Boosting noisy data. In *International Conference on Machine Learning*, pages 274–281, 2001.

[242] G. I. Webb. Multiboosting: A technique for combining boosting and wagging. *Machine learning*, 40(2):159–196, 2000.

[243] D. Guan, W. Yuan, Y. Lee, and S. Lee. Identifying mislabeled training data with the aid of unlabeled data. *Applied Intelligence*, 35(3):345–358, 2011.

[244] D. W. Aha and D. F. Kibler. Noise-tolerant instance-based learning algorithms. In *11th international joint conference on Artificial intelligence*, pages 794–799, 1989.

[245] F. Breve, L. Zhao, and M. G. Quiles. Semi-supervised learning from imperfect data through particle cooperation and competition. In *International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2010.

[246] A. Ganapathiraju and J. Picone. Support vector machines for automatic data cleanup. In *INTERSPEECH*, pages 210–213, 2000.

[247] M. R. Smith and T. Martinez. Improving classification accuracy by identifying and removing instances that should be misclassified. In *International Joint Conference on Neural Networks*, pages 2690–2697, 2011.

[248] S. Verbaeten and A. V. Assche. Ensemble methods for noise elimination in classification problems. *International Workshop on Multiple classifier systems*, pages 317–325, 2003.

[249] C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(6):601–618, 2010.

[250] A. Peña-Ayala. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, 41(4):1432–1462, 2014.

[251] J. Hernándeza, A. Ochoa, J. Muñoz, and G. Burlak. Detecting cheats in online student assessments using data mining. In *International Conference on Data Mining*, pages 183–188, 2006.

[252] M. Ng, C. Vee, B. Meyer, and K. L. Mannock. Understanding novice errors and error paths in object-oriented programming through log analysis. In *Workshop on Educational Data Mining*, pages 13–20, 2006.

[253] M. Ueno and K. Nagaoka. Learning log database and data mining system for e-learning - on line statistical outlier detection of irregular learning processes. In *International Conference on Advanced Learning Technologies*, pages 436–438, 2002.

[254] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009.

[255] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar. A comparative study of anomaly detection schemes in network intrusion detection. In *3rd SIAM International Conference on Data Mining*, pages 25–36, 2003.

[256] M. R. Smith and T. Martinez. Reducing the effects of detrimental instances. In *13th International Conference on Machine Learning and Applications*, pages 183–188. IEEE, 2014.

[257] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):224–227, 1979.

[258] G. Siemens. Connectivism: A learning theory for the digital age, 2014. URL http://www.elearnspace.org/Articles/connectivism.htm. Último acceso: marzo 2016.

[259] J. L. Moreno. *Who shall survive?* Nervous and mental disease publishing co., 1934.

[260] V. E. Krebs. Mapping networks of terrorist cells. *Connections*, 24(3):43–52, 2002.

[261] C. Palazuelos and M. Zorrilla. Fringe: a new approach to the detection of overlapping communities in graphs. In *Internacional Conference on Computational Science and Its Applications*, pages 638–653. Springer, 2011.

[262] C. Palazuelos and M. Zorrilla. Analysis of social metrics in dynamic networks: measuring the influence with fringe. In *Joint EDBT/ICDT Workshops*, pages 9–12. ACM, 2012.

[263] A. Calvó-Armengol, E. Patacchini, and Y. Zenou. Peer effects and social networks in education. *The Review of Economic Studies*, 76(4):1239–1267, 2009.

[264] K. L. Cela, M. Á. Sicilia, and S. Sánchez. Social network analysis in e-learning environments: A preliminary systematic review. *Educational Psychology Review*, 27(1):219–246, 2015.

[265] A. Corallo, M. De Maggio, F. Grippa, and G. Passiante. A methodological framework to monitor the performance of virtual learning communities. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 20(2):135–148, 2010.

[266] A. Duensing, U. Stickler, C. Batstone, and B. Heins. Face-to-face and online interactions-is a task a task? *Journal of learning design*, 1(2):35–45, 2006.

[267] M. De Laat. Network and content analysis in an online community discourse. In *Conference on Computer Support for Collaborative Learning*, pages 625–626. International Society of the Learning Sciences, 2002.

[268] M. De Laat, V. Lally, L. Lipponen, and R. Simons. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for social network analysis. *International Journal of Computer-Supported Collaborative Learning*, 2(1):87–103, 2007.

[269] A. Martınez, Y. Dimitriadis, B. Rubia, E.o Gómez, and P. De La Fuente. Combining qualitative evaluation and social network analysis for the study of classroom social interactions. *Computers & Education*, 41(4):353–368, 2003.

[270] S. Dawson. 'seeing' the learning community: An exploration of the development of a resource for monitoring online student networking. *British Journal of Educational Technology*, 41(5):736–752, 2010.

[271] W. C. Paredes and K. S. K. Chung. Modelling learning performance: A social networks perspective. In *2nd International Conference on Learning Analytics and Knowledge*, LAK '12, pages 34–42, New York, NY, USA, 2012. ACM.

[272] K. Stepanyan, K. Borau, and C. Ullrich. A social network analysis perspective on student interaction within the twitter microblogging environment. In *IEEE 10th International Conference on Advanced Learning Technologies*, pages 70–72, July 2010.

[273] B.Y. Erlin, N. Yusof, and A.A. Rahman. Integrating content analysis and social network analysis for analyzing asynchronous discussion forum. In *International Symposium on Information Technology*, volume 3, pages 1–8, Aug 2008.

[274] L. Lipponen, M. Rahikainen, J. Lallimo, and K. Hakkarainen. Patterns of participation and discourse in elementary students' computer-supported collaborative learning. *Learning and Instruction*, 13(5):487 – 509, 2003.

[275] M. De Laat, V. Lally, L. Lipponen, and R. Simons. Analysing student engagement with learning and tutoring activities in networked learning communities. *International Journal of Web Based Communities*, 2(4):394–412, December 2006.

[276] A.B.F. Mansur, N. Yusof, and M.S. Othman. Analysis of social learning network for wiki in moodle e-learning. In *Interaction Sciences (ICIS), 2011 4th International Conference on*, pages 1–4, Aug 2011.

[277] Peng H. Evaluating students online discussion performance by using social network analysis. In *9th International Conference on Information Technology: New Generations*, pages 854–855, April 2012.

[278] A. Gewerc-Barujel and M. Montero-Mesa, L.and Lama-Penín. Collaboration and social networking in higher education. colaboración y redes sociales en la enseñanza universitaria. *Comunicar*, 21(42):55–63, 2014.

[279] S. Panchoo. Using the social network analysis as a pedagogical tool to enhance online interactions. In *International Conference on Advanced Computer Science Applications and Technologies*, pages 290–294, 2012.

[280] L. M. Romero-Moreno. An approach to collaborative interaction analysis in virtuales learning systems using social network analysis. In *8th Iberian Conference on Information Systems and Technologies*, pages 1–5. IEEE, 2013.

[281] C.S. Fishpaw and M. Ketel. Online social networking practices and the implications for e-learning solutions. In *SouthEastCon 2014, IEEE*, pages 1–7, March 2014.

[282] C. Carceller, S. Dawson, and L. Lockyer. Social capital from online discussion forums: Differences between online and blended modes of delivery. *Australasian Journal of Educational Technology*, pages 150–163, 2015.

[283] G. Putnik, E. Costa, C. Alves, H. Castro, L. Varela, and V. Shah. Analysing the correlation between social network analysis measures and performance of students in social network-based engineering education. *International Journal of Technology and Design Education*, pages 1–25, 2015.

[284] K. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1–37, 1989.

[285] L. Szathmary, A. Napoli, and P. Valtchev. Towards rare itemset mining. In *19th IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 305–312. IEEE, 2007.

[286] R. Belohlavek, E. Sigmund, and J. Zacpal. Evaluation of ipaq questionnaires supported by formal concept analysis. *Information Sciences*, 181(10):1774–1786, 2011.

[287] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *20th International Conference on Very Large Data Bases*, pages 478–499. Morgan Kaufmann, Los Altos, CA, 1994.

[288] E. García, C. Romero, S. Ventura, and T. Calders. Drawbacks and solutions of applying association rule mining in learning management systems. In *International Workshop on Applying Data Mining in e-Learning*, pages 13–22, 2007.

[289] T. Scheffer. Finding association rules that trade support optimally against confidence. In *5th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 424–435. Springer, 2001.

[290] A. Merceron and K. Yacef. Interestingness measures for association rules in educational data. In *International Conference on Educational Data Mining*, pages 57–66, 2008.

[291] C. Borgelt and R. Kruse. Induction of association rules: Apriori implementation. In *International Conference on Computational Statistics*, pages 395–400. Springer, 2002.

[292] M. J. Zaki and C. Hsiao. Charm: An efficient algorithm for closed association rule mining. Technical report, 1999. URL http://www.cs.rpi.edu/tr/99-10.pdf.

[293] J.L. Balcázar. Formal and computational properties of the confidence boost of association rules. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(4):19, 2013.

[294] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11): 27–34, 1996.

[295] P. Panov, L. N. Soldatova, and S. Džeroski. Towards an ontology of data mining investigations. In *International Conference on Discovery Science*, pages 257–271. Springer, 2009.

[296] L.N. Soldatova and R.D. King. An ontology of scientific experiments. *Journal of the Royal Society Interface*, 3(11):795–803, 2006.

[297] C. Diamantini, D. Potena, and E. Storti. Ontology-driven kdd process composition. In *International Conference on Advances in Intelligent Data Analysis*, pages 285–296. Springer, 2009.

[298] M. Hilario, A. Kalousis, P. Nguyen, and A. Woznica. A data mining ontology for algorithm selection and meta-mining. In *ECML/PKDD09 Workshop on 3rd generation Data Mining*, pages 76–87, 2009.

[299] M. Hilario, P. Nguyen, H. Do, A. Woznica, and A. Kalousis. Ontology-based meta-mining of knowledge discovery workflows. In *Meta-Learning in Computational Intelligence*, pages 273–315. Springer, 2011.

[300] J. Vanschoren and L. Soldatova. Exposé: An ontology for data mining experiments. In *International workshop on third generation data mining: Towards service-oriented knowledge discovery (SoKD-2010)*, pages 31–46, 2010.

[301] M. M. Campos, P. J. Stengard, and B. L. Milenova. Data-centric automated data mining. In *4th International Conference on Machine Learning and Applications*, pages 8–pp. IEEE, 2005.

[302] M. B. Ayed, H. Ltifi, C. Kolski, and A. M. Alimi. A user-centered approach for the design and implementation of kdd-based dss: a case study in the healthcare domain. *Decision Support Systems*, 50(1):64–78, 2010.

[303] E. Blanco and M. Corominas. Cbs: an open platform that integrates predictive methods and epigenetics information to characterize conserved regulatory features in multiple drosophila genomes. *BMC genomics*, 13(1):688, 2012.

[304] M. R. Kamdar, D. Zeginis, A. Hasnain, S. Decker, and H. F. Deus. Reveald: A user-driven domain-specific interactive search platform for biomedical research. *Journal of biomedical informatics*, 47:112–130, 2014.

[305] T. Luu, A. Rusu, V. Walter, B. Linard, L. Poidevin, R. Ripp, L. Moulinier, J. Muller, W. Raffelsberger, and N. Wicker. Kd4v: comprehensible knowledge discovery system for missense variant. *Nucleic acids research*, 40(1):71–75, 2012.

[306] R.S. Santos, S.M.F. Malheiros, S. Cavalheiro, and J.M. P. De Oliveira. A data mining system for providing analytical information on brain tumors to public health decision makers. *Computer methods and programs in biomedicine*, 109(3): 269–282, 2013.

[307] Y. Psaromiligkos, M. Orfanidou, C. Kytagias, and E. Zafiri. Mining log data for the analysis of learners' behaviour in web-based learning management systems. *Operational Research*, 11(2):187–200, 2011.

[308] A. Bellaachia, E. Vommina, and B. Berrada. Minel: A framework for mining e-learning logs. In *5th International Conference on Web-based education*, pages 259–263. ACTA Press, 2006.

[309] B. Jong, T. Chan, and Y. Wu. Learning log explorer in e-learning diagnosis. *Education, IEEE Transactions on*, 50(3):216–228, 2007.

[310] I. Jugo, B. Kovačić, and E. Tijan. Cluster analysis of student activity in a web-based intelligent tutoring system. *Pomorstvo: Scientific Journal of Maritime Research*, 29(1):75–83, 2015.

[311] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (wsdl) 1.1, 2001. URL http://wsdl2code.googlecode.com/svn/trunk/03-Literature/WSDL/wsdl.1.1..pdf. Último acceso: marzo 2016.

[312] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-oriented domain analysis FODA feasibility study. Technical report, DTIC Document, 1990.

[313] L.M. Rose, R.F. Paige, D.S. Kolovos, and F.A.C. Polack. The epsilon generation language. In *European Conference on Model Driven Architecture–Foundations and Applications*, pages 1–16. Springer, 2008.

[314] E. Cambria and B. White. Jumping nlp curves: a review of natural language processing research [review article]. *Computational Intelligence Magazine, IEEE*, 9(2):48–57, 2014.