
***Facultad***
***de***
***Ciencias***

# APLICACIÓN DE TÉCNICAS DE CIENCIA DE DATOS PARA MEJORAR LA GESTIÓN DE FLOTAS DE TRANSPORTE

## (Application of Data Science techniques to improve the management of transport fleets)

**Trabajo de Fin de Grado**
**para acceder al**

**GRADO EN INGENIERÍA INFORMÁTICA**

**Autor: Jorge Gutiérrez Mantecón**

**Directora: Cristina Tirnauca**

**Co-Director: Diego García Saiz**

**06 - 2022**

# Índice

# Resumen

La electrónica embarcada está aportando a las empresas y usuarios la posibilidad de disponer de gran cantidad de datos sobre el uso de sus vehículos y de la gestión de los mismos por los conductores. En este proyecto se aplicarán técnicas de *Data Science* para investigar patrones descriptivos y predictivos de la actividad logística de una empresa cántabra. Los resultados permitirán tomar decisiones estratégicas con anticipación, orientadas a la optimización de sus procesos actuales o el establecimiento de nuevos procesos, de manera que permita una gestión optimizada de sus flotas.

En concreto, se trabajará con datos que representan el rendimiento de un vehículo en un tramo realizado, tales como el número de frenadas bruscas realizadas, el tiempo de uso de conducción de crucero o de ralentí. Utilizando ese tipo de información, se pretende descubrir posibles relaciones entre atributos e identificar patrones recurrentes en los datos, construir un modelo predictivo para clasificar los tramos de conducción según la valoración que reciben los conductores y obtener perfiles de conducción en función de sus datos.



**Palabras clave**: Ciencia de Datos, minería de datos, reglas de asociación, árbol de decisión, flotas de transporte.



# Abstract

On-board electronics are providing companies and users with the possibility of having a large amount of data about the driver's performance and their use of the vehicles. In this project, Data Science techniques will be applied to investigate descriptive and predictive patterns of the logistics activity of a company headquartered in Cantabria. The results will allow strategic decisions aimed at optimizing or establishing processes to be made in advance, improving management of the fleets.

In particular, data that represent the performance of a vehicle in a section is used, such as the number of sudden braking operations carried out, the time used for cruising or idling. Using this type of information, the aim is to discover possible relationships between attributes and identify recurring patterns in the data, to build a predictive model to classify driving sections according to the assessment received by drivers and to obtain driving profiles based on their data.



**Keywords**: Data Science, data mining, association rules, decision tree, transport fleets.

# 1. Introducción y entendimiento del negocio

Gracias a la electrónica embarcada, diferentes empresas están teniendo la oportunidad de obtener, en tiempo real, información muy variada sobre sus vehículos y la gestión realizada por sus conductores, tales como el consumo de combustible, la localización geográfica o el estado del motor.

En un sector donde la tecnología está ganando cada vez más importancia, este tipo de datos, explotados de forma correcta, permiten a las empresas gestionar de manera óptima su flota de vehículos, obteniendo una ventaja dentro del mercado respecto a su competencia y reduciendo costes de forma significativa[1]. Por esta razón, la empresa Fagor Electrónica S. Coop. ha decidido apostar por la Universidad de Cantabria[2], para que juntos logren extraer aquellas conclusiones que les permitan tomar las mejores decisiones. Es importante destacar que en este trabajo son igual de importantes el dominio en análisis de datos como el conocimiento del negocio.

Fagor Electrónica S. Coop. es una empresa cooperativa española fundada en 1966 dentro de la Corporación Mondragón, el grupo corporativo más grande del mundo[3] formado por 98 cooperativas, 8 fundaciones, 1 mutua, 10 entidades de cobertura y 7 delegaciones internacionales, distribuidas en cuatro áreas: finanzas, industria, distribución y conocimiento.

En el año 2000 y de la mano de la Universidad de Cantabria, Fagor Electrónica fundó en Santander una nueva división para el tratamiento de la información en la cadena de suministro y que hoy tiene el nombre de Fagor Smart Data Services[4] (en adelante SDS). SDS se encarga del diseño, desarrollo y comercialización de servicios digitales basados en geolocalización para los sectores del transporte y la logística. A día de hoy tiene numerosos clientes en dichos sectores, además de aquellas empresas donde la cadena del suministro es muy importante.

Dentro de los datos proporcionados por la empresa para la realización de este proyecto, hay unos en concreto en los que se prestará especial atención: los "tramos". Los tramos representan un trayecto realizado por un vehículo y conductor determinados y con unas fechas de inicio y final establecidas. Dentro de cada uno, se pueden acceder a numerosos atributos, tales como los indicados al inicio del apartado, y con los que se trabajará para lograr los objetivos propuestos durante las reuniones iniciales con los integrantes del equipo del proyecto arriba mencionado, que se explicarán con más detalle a continuación.

[1] Deloitte. Global truck study 2016 - The truck industry in transition, página 5, 2016. https://www2.deloitte.com/us/en/pages/manufacturing/articles/global-truck-study-2016.html

[2] "RUT-IA: Investigación Industrial de tecnologías de analítica de datos para el tratamiento de información de explotación logística georeferenciada", proyecto competitivo regional financiado por la Conserjería de Innovación, Industria, Transporte y Comercio dentro de la convocatoria INNOVA COVID-19 2020.

[3] CEPES. Las empresas más relevantes de la economía social. CEPES, página 82, 2017-2018. https://www.cepes.es/files/publicaciones/112.pdf

[4] Cuya página web es https://www.fagorsmartdata.com/

## 1.1. Objetivos

Los objetivos concretos para este Trabajo de Fin de Grado son los siguientes:

- Descubrir posibles relaciones entre los valores de los atributos relevantes dentro de los tramos de conducción e identificar patrones recurrentes en los datos.
- Construir un modelo predictivo para clasificar los distintos tramos de conducción según la valoración que reciben los conductores en función de los atributos del tramo. Para este caso, se tiene que dividir la valoración entre buena y mala, especificando un límite numérico para ello.
- Obtener perfiles de conducción en función de datos como la velocidad media, el consumo realizado en litros cada 100 kilómetros o el porcentaje del tiempo de conducción en el cual se han estado realizando aceleraciones bruscas.

## 1.2. Herramientas a utilizar

Para la realización de este proyecto, la empresa ha proporcionado una base de datos con toda la información logística que poseen. Por lo tanto, ha sido necesario utilizar un sistema de gestión de bases de datos relacionales, para así poder manejar las tablas y hacer consultas, además de observar y analizar los códigos de los procedimientos almacenados y funciones. Para ello, se ha utilizado *SQL Server Management Studio*[5] de *Microsoft SQL Server*, una de las herramientas más populares para este tipo de casos[6] y que, como su propio nombre indica, utiliza el lenguaje de dominio específico *SQL* (*Structured Query Language* o lenguaje de consulta estructurada).

Para realizar el análisis de datos, se emplea el lenguaje de programación *Python*, uno de los más utilizados para este tipo de proyectos[7], con las siguientes librerías:

- La librería *pandas* es la encargada de cargar los conjuntos de datos para su posterior tratamiento. Con ella se pueden analizar las distribuciones de los datos, formatearlos y eliminar aquellos con los que no se esté conforme, por lo que es una librería fundamental en el apartado de preprocesado.
- *NumPy* es una librería que permite operar de manera muy eficiente con *arrays* de dimensiones elevadas.
- La librería *sklearn* (también llamada *Scikit-learn*) [1] proporciona numerosos y distintos algoritmos útiles para el proceso de minería de datos.
- La librería *matplotlib* ha sido utilizada para visualizar gráficas de una manera sencilla y rápida.

[5] Versión 18.10, publicada el 5 de octubre de 2021.

[6] Las herramientas más populares para la gestión de bases de datos relacionales, actualizado a junio de 2022: https://db-engines.com/en/ranking/relational+dbms, *Microsoft SQL Server* se encuentra la tercera.

[7] Valorado como el primero en el siguiente ranking: https://www.datacamp.com/blog/top-programming-languages-for-data-scientists-in-2022

- La librería *mlxtend* es la encargada de proporcionar todas las herramientas para generar las reglas de asociación, incluyendo la creación de tablas booleanas y algoritmos como *Apriori*.

- El ejecutable *SPMF* (*Sequential Pattern Mining Framework*) permite producir las reglas raras. Para poder utilizarlo en *Python*, se ha usado un *Python wrapper* con el mismo nombre[8], que es el que ejecuta directamente este programa con los parámetros indicados [2].

# 2. Desarrollo del trabajo

El éxito en los proyectos de minería de datos no está nunca garantizado, pero la probabilidad de conseguirlo aumenta, en parte, si se usa una metodología adecuada [3]. En este caso, después de analizar las distintas opciones, se ha optado por CRISP-DM (*Cross-industry standard process for data mining*), al ser considerado el modelo más apropiado para este caso[9].

CRISP-DM es un modelo estándar propuesto en 1996 por tres grandes empresas: Daimler-Benz, ISL y NCR, y financiado por la Comisión Europea. Posteriormente, fue publicado en 1999 [4, 5]. Este modelo sugiere seis fases que se deben seguir a la hora de realizar un proyecto: entendimiento de negocio, entendimiento de datos, preparación de datos, fase de modelado o análisis, evaluación y despliegue. Las fases están relacionados entre sí de diferentes formas, tal y como se puede ver en la Figura 1:

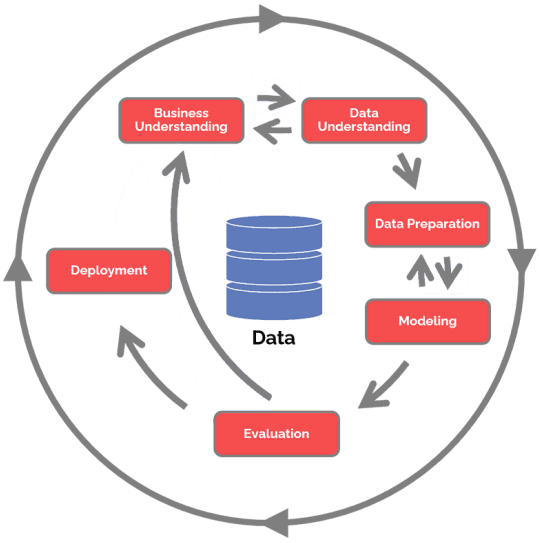


Figura 1: Las diferentes fases de la metodología CRISP-DM y sus conexiones[10].

[8] Cuyo software se encuentra en https://pypi.org/project/spmf/.

[9] Forbes: What IT Needs To Know About The Data Mining Process, https://www.forbes.com/sites/metabrown/2015/07/29/what-it-needs-to-know-about-the-data-mining-process/?sh=4ef91a2c515f.

[10] Imagen obtenida de https://www.datascience-pm.com/crisp-dm-2/.

El funcionamiento de cada una de las fases es el siguiente:

- La fase del entendimiento de negocio engloba todas las actividades relacionadas con la comprensión de los requisitos y la determinación de los objetivos. A partir de esta fase, se formará una hoja de ruta con la que realizar el resto de fases del proyecto.
- A continuación, está la fase del entendimiento de los datos, que consiste en el acceso y exploración de los datos iniciales con el fin de describirlos y verificar su calidad.
- La siguiente fase es la preparación y gestión de los datos, donde se acondicionan los datos según las conclusiones obtenidas en la fase anterior, ya sea limpiando filas de datos o atributos que no sirvan, reestructurándolos y/o formateándolos, para finalmente seleccionar aquellos que sean útiles.
- El modelo o análisis es la siguiente fase, que consiste en aplicar diferentes técnicas de minería de datos para conseguir los objetivos propuestos. Mientras se realiza esta fase, es habitual que tengamos que volver a la fase anterior para refinar la estructura de los datos.
- La penúltima fase es la de resultados y evaluación, que consiste en analizar y verificar si los resultados obtenidos en la fase de análisis satisfacen los objetivos planeados al inicio del proyecto.
- Por último, está la fase final de despliegue, que consiste en implementar el modelo para que los usuarios puedan verlo y usarlo.

Como ya se ha indicado anteriormente, hay varias fases que se han realizado más de una vez en diferentes etapas del proyecto. Estas fases, con excepción de la última, se han llevado a cabo dentro del proyecto propuesto para una beca de colaboración, financiada por el Ministerio de Educación y Formación Profesional y dirigida por Diego García Saiz, miembro del Departamento de Ingeniería Informática y Electrónica y co-director del presente Trabajo de Fin de Grado. El fruto de las fases realizadas se describe con detalle en un apartado de la presente memoria, tal y como se indica a continuación:

- La fase del entendimiento de datos está representada en el tercer apartado de la memoria.
- La fase de la preparación y gestión de los datos se encuentra en el cuarto apartado de la memoria, con una parte realizada en el sub-apartado 3.2.1, anterior al análisis de las distribuciones.
- La fase de análisis se encuentra en el quinto apartado, donde se explican los algoritmos utilizados.
- La fase de resultados y evaluación está en el sexto apartado.

No tendrá su propio apartado la fase de entendimiento de negocio al no considerarse necesario emplear toda una sección para ello.

## 2.1. Cronología

El orden que se ha seguido a la hora de realizar el trabajo es el representado en la Figura 2:

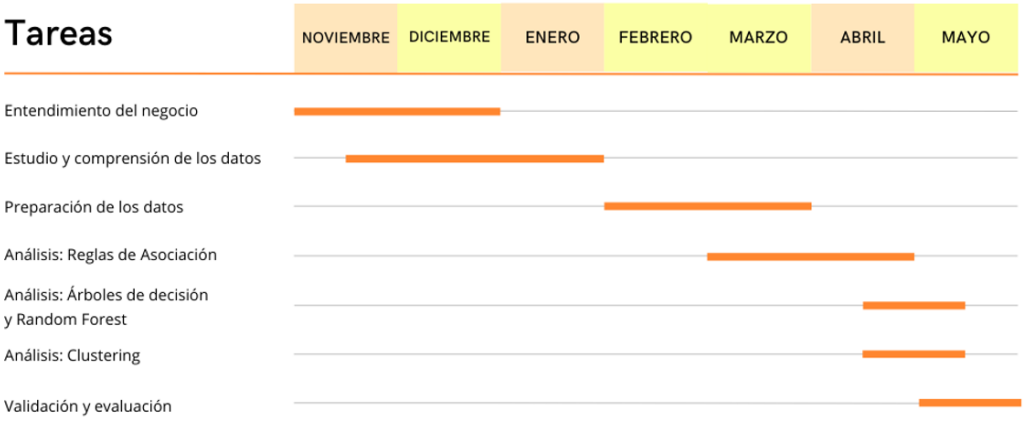


Figura 2: Diagrama de Gantt representando el orden seguido en el trabajo.

Se empezó inicialmente con la fase de entendimiento del negocio, una fase bastante densa compuesta de varias reuniones con la empresa, clarificando los objetivos y adquiriendo una idea inicial del proyecto que había que desarrollar. Se pasó de forma paralela a realizar el estudio y comprensión de datos a partir de mediados de noviembre, otra etapa con mucho trabajo en la que se revisaron y estudiaron todas las tablas relevantes pertenecientes a la base de datos proporcionada, además de los procedimientos almacenados y métodos. Estas dos fases estuvieron relacionadas, muchas veces un descubrimiento en la segunda implicó una nueva reunión de negocio para aclararlo, y como consecuencia, un nuevo avance en el entendimiento de negocio.

Tras haber analizado todos los datos, y teniendo claro qué se debe realizar, se pasó a la fase de preparación de los datos en febrero. Esta fase, que ocupó unos dos meses, consistió en tratar los datos de forma que se puedan eliminar aquellos que sean extremos o no interesantes, utilizando el trabajo previo en anteriores apartados y garantizando el buen funcionamiento de los posteriores.

A partir de marzo, se empezó a aplicar finalmente una de las técnicas de minerías de datos para obtener uno de los principales objetivos, las reglas de asociación. Este análisis fue largo, ya que hubo que repetirlo en numerosas ocasiones utilizándolo de diferentes formas hasta obtener un resultado interesante y útil. Por otra parte, a lo largo de abril se empezaron a aplicar las técnicas de generación de predictores y de generación de perfiles con *clustering*. Por último, se utilizó el mes final para la finalización de estos análisis y para la validación y evaluación de todos los resultados obtenidos.

# 3. Estudio y comprensión de los datos

La información proporcionada por Fagor SDS acerca de su flota y utilizada para cumplir los objetivos procede de dos fuentes principales:

- Sistema de telemetría: Equipo de comunicaciones instalado en cada vehículo, que se encarga de transmitir datos relacionados con el posicionamiento GPS (*Global Positioning System* o Sistema de Posicionamiento Global), tales como la posición, el rumbo o los metros ascendidos, y datos también relacionados con sensores directos, como por ejemplo medidas de temperatura.

- *CanCliQ*: Hardware de conexión y preprocesado de la información generada por las centralitas del vehículo. Se encarga de elegir los datos de interés enviados por el sistema de telemetría anteriormente mencionado, preprocesándolos posteriormente antes de guardarlos en las bases de datos.

## 3.1. Presentación de los datos

A continuación, se presentan aquellas tablas seleccionadas como relevantes para la consecución de los objetivos propuestos, siendo las siguientes:

- *Vehiculos*: Esta tabla contiene información de todos los vehículos registrados en la empresa, siendo un total de 5020.

- *ValoraciónConductores_Parametros*: Contiene todos los parámetros utilizados a la hora de valorar conductores, indicando su valoración, mínimo y máximo.

- *ValoraciónConductores_Perfiles*: En esta tabla se encuentran todos los perfiles posibles a la hora de valorar conductores. Los perfiles son creados para determinar los porcentajes de Seguridad, Eficiencia, Ecología y Calidad que tendrán a la hora de calcular la valoración.

- *EventosHistorico*: Todos los eventos registrados por la empresa se encuentran en esta tabla, a partir de los cuales se calculan los tramos. Un evento es una lectura del dispositivo empotrado del vehículo, generado en muchas ocasiones por un cambio de estado, como un arranque de motor, por ejemplo. Por otra parte, hay otros eventos que podrían ser generados por exceso de velocidad o por entrar a una zona geográfica no permitida, por lo que se podría entender entonces que un evento se genera cuando hay un suceso interesante. Dicho de otra forma, en esta tabla se encuentran todos los datos preprocesados por CanCliQ.

- *Tramos*: Tabla que contiene los distintos tramos creados, los cuales están asociados a un vehículo. Un tramo se genera a partir de los eventos históricos guardados en la tabla anterior y se delimita por aquellos eventos en los que haya un arranque, una parada del motor o incluso el ralentí. De forma que, por ejemplo, si se arranca el motor, se realiza un trayecto y luego se para el motor, todo ese tiempo desde el arranque hasta la parada es un tramo.

El análisis de los atributos de las tablas proporcionadas por la empresa y de los procedimientos almacenados relacionados se encuentra detallado en el documento titulado "RUT-IA: Caso de uso 1", publicado en diciembre de 2021 y enviado a la empresa[11]. Por otra parte, debido a la gran cantidad de valores en algunas tablas, como en EventosHistorico, donde hay 15675511 filas, o en Tramos, donde hay 2085037 filas, se ha optado por seleccionar un periodo concreto que tenga una cantidad de valores interesante, pasando a guardar dicha información en ficheros *Excel* para su posterior uso en *Python*. Este periodo concreto es el mes de marzo de 2021.

La tabla utilizada para el proyecto, que unifica la información de las tablas arriba mencionadas, tiene un total de 171454 filas y 132 columnas, siendo una cantidad bastante elevada de atributos. La lista total de columnas se encuentra en el documento "Anexo I: Lista total de columnas de la tabla Tramos" dentro del anexo, al ser demasiado extensa.

De primeras, se pueden apreciar varios atributos cuyo nombre deja bastante claro cuál es su función, como por ejemplo Id_Vehiculo, que claramente representa el identificador del vehículo que ha realizado el tramo, o VelocidadMedia, que representa la velocidad media que se ha realizado en el tramo. Sin embargo, otros atributos no dejan tan claro su significado si solo se presta atención al nombre.

Por otra parte, también se pueden ver varios atributos con el mismo nombre junto a un 2 o un 3, como por ejemplo TiempoConduccionCrucero2 y TiempoConduccionCrucero3, lo que parece representar una distinta versión del atributo. Para poder ver qué atributos son interesantes y qué representa cada uno, se ha llevado a cabo un estudio de ellos, realizando previamente una fase de preprocesado y preparación de los atributos para reducir su tamaño.

## 3.2. Preprocesado y preparación de los atributos

En este sub-apartado se redujo el tamaño de atributos disponible, hasta obtener un conjunto razonable e interesante que se pudiese analizar para entender a la perfección los datos de los que disponemos. Para ello, primero se prestó atención al alto número de columnas, eliminando aquellas que no contengan información interesante, ya sea por lo que representan o por una baja variedad de valores. De igual manera, ha sido necesario visualizar la correlación entre columnas para evitar aquellas parejas que estén muy relacionadas.

### 3.2.1. Reducción de columnas

La cantidad de atributos es muy grande y hay una gran probabilidad de que varios de ellos no contengan información interesante, así que antes de ver qué representa cada atributo realmente y sus distribuciones, se analizó qué variedad de valores contiene cada uno, para así saber si iban a ser útiles en, por ejemplo, las reglas de asociación.

[11] El documento se puede leer en el siguiente enlace: https://unicancloud-my.sharepoint.com/:b:/g/personal/jgm188_alumnos_unican_es/EQtXqwfVlANLmNx1JNSbhZABtfo_RAnOaLpGPDQ_KByTfw?e=MbRe2a

Inicialmente, se eliminaron todas aquellas columnas que solo contenían un único valor, ya sea un 0 o cualquier otro valor, ya que estos datos no aportan ningún tipo de información a los tramos. La columnas eliminadas de esta forma son las indicadas en el documento "Anexo II: Columnas de valor único" del anexo. De esta forma, se eliminó una cantidad considerable de atributos que no aportaban ningún tipo de información interesante, reduciendo así peso a la hora de generar las reglas.

Por otra parte, se observaron también aquellas columnas que tuviesen muchos valores nulos, ya que tampoco aportaban información interesante. Dado que hay un total de 171454 filas, he considerado que aquellas columnas que tengan un valor superior al 15 % de nulos serán columnas poco interesantes. El valor se ha decidido observando el recuento de valores nulos por atributo, ya que la gran mayoría de los que tenían valores nulos tenían un 15 % o más. De esta forma, se descartaron las columnas indicadas en el documento "Anexo III: Columnas con muchos valores nulos" del anexo, consiguiendo que la lista de atributos tenga una cantidad más manejable.

Hay un atributo con el que se hizo una excepción respecto a lo indicado anteriormente. La columna Error1 indica si hay algún error en esa fila que implique que los datos no son útiles, de forma que si no hay error y la fila es correcta, el valor en este campo es un nulo. Se utilizó este atributo para eliminar aquellas filas erróneas, que serán las que no tengan este campo con un valor nulo. Los posibles valores del campo Error1 y su representación se encuentran en el documento "Anexo IV: Códigos de error" del anexo.

Después de estos pasos, se prestó especial atención a un tema mencionado al principio del apartado: los atributos que tienen un 2 al final del nombre y los que tienen un 3. Según lo indicado en reuniones con la empresa, estos números representan diferentes versiones de un mismo atributo. Es decir, en el caso de las aceleraciones bruscas, hay dos atributos, AceleracionesBruscas2 y AceleracionesBruscas3, ambos representan lo mismo pero con diferente versión. La diferencia de versión viene dada porque la empresa tiene numerosos clientes que introducen los datos de distinta forma. La última versión es la que termina en 3, por lo que, en el caso de este tipo de atributos, se mantuvieron los que terminaban en dicho número. De esta manera, las columnas indicadas en el documento "Anexo V: Columnas eliminadas por distinta versión" del anexo fueron eliminadas, y se utilizaron sus versiones con un 3.

Por último, se finalizó la eliminación de columnas retirando aquellas que sean muy similares, utilizando para ello la correlación. Dos variables están correlacionadas si sus valores disminuyen o aumentan de una forma muy similar en ambos casos, haciendo que uno de los dos atributos no sea útil al representar ambos unos valores muy parecidos. Buscando aquellas parejas de atributos que tengan 1 de correlación, es decir, el valor máximo que represente que son atributos muy relacionados, se obtuvo el resultado mostrado en la Tabla 1.

Tabla 1: Tabla de parejas de atributos con 1 de correlación.

| **Atributo 1** | **Atributo 2** | **Coeficiente de correlación** |
|---|---|---|
| TotalFuel3 | Combustible | 1.0 |
| Odometro3 | Distancia | 1.0 |
| DistProf | Distancia | 1.0 |
| FrenadasBruscas3 | FrenadasBruscasOut | 1.0 |
| AceleracionesBruscas3 | AceleracionesBruscasTOut | 1.0 |
| TiempoConduccionCrucero3 | CruiseActiveTOut | 1.0 |
| CNoPredictiva3 | ConduccionNoPredictivaTOut | 1.0 |
| DistProf | Odometro3 | 1.0 |
| TiempoTotal | TiemProf | 1.0 |

De estas parejas, es necesario eliminar un atributo de cada una para evitar que haya dos con correlación máxima. En el primer caso, TotalFuel3 tiene el mismo valor que Combustible pero multiplicado por 1000 (es decir, está en otra unidad), por lo que se eliminó TotalFuel3, y Combustible se mantuvo al estar en litros, que es una medida más popular. Exactamente lo mismo ocurre con Odometro3 y Distancia, representando la distancia recorrida en metros y en kilómetros respectivamente. Esta es la razón por la que se mantuvo el segundo, debido a que el kilómetro es una unidad preferible que podremos relacionar directamente con la velocidad. En el resto de casos, las parejas tienen exactamente los mismos valores, por lo que directamente se eliminaron aquellas que contienen "Out" o "Prof" en el nombre, siguiendo así con lo indicado por la empresa acerca de las versiones de los atributos.

Finalmente, hubo que seleccionar unos datos, dentro de la lista que pasó los filtros anteriores, que fuesen útiles para el análisis de reglas. Si bien los atributos que quedaban ya tenían una información muy completa, muchos de ellos no eran realmente útiles para el objetivo del Trabajo de Fin de Grado, por lo que hubo que ir filtrándolos. Dentro de la lista filtrada de atributos, hay muchos de tipo *String* o de tipo "fecha y hora" que realmente no son utilizables para, por ejemplo, la generación de reglas. Los atributos seleccionados son los siguientes:

- *Id_Vehiculo*: representa el identificador del vehículo que ha realizado el tramo, por lo que puede ser interesante a la hora de ver si hay ciertos vehículos más propensos a obtener ciertas métricas. Otros atributos como el nombre del vehículo no son necesarios, debido a que este atributo es suficiente para identificar a cada uno.

- *Id_Conductor*: por la misma razón que antes, se mantuvo el identificador del conductor que es lo que mejor distingue a la persona que realizó el tramo. El resto de posibles parámetros del conductor no son interesantes.

- *MinutosTrayecto*: tiempo transcurrido desde el momento en el que el vehículo empieza a moverse dentro del tramo hasta el final del tramo, expresado en minutos. Es decir, dentro del tramo, la fecha del primer evento en el que se detecte movimiento (variación en la distancia recorrida) será la fecha de salida. Esta fecha es la que se resta a la fecha de fin de tramo, obteniendo el tiempo de trayecto. No se debe confundir con el tiempo total, que va desde la fecha del primer

evento del tramo, esté el vehículo en movimiento o no, hasta la fecha del final del tramo.

- *TiempoConduccion3seg*: representa el tiempo de conducción en segundos del tramo, es decir, el tiempo en el que el vehículo ha estado en movimiento.
- *VelocidadMedia*: velocidad media del vehículo en todo el tramo, en kilómetros por hora. Es un atributo interesante para poder ver qué tramos han sido con una velocidad más alta y cuáles no, pudiendo identificar distintos tipos de conducción. Se obtiene dividiendo el campo de Distancia entre el campo de TiempoConduccion3seg, este último campo pasado previamente de segundos a horas.
- *Combustible*: consumo de combustible realizado a lo largo del tramo, en litros.
- *Distancia*: distancia en kilómetros recorrida por el vehículo en el tramo.
- *Consumo100*: este atributo representa el consumo de combustible en litros cada 100 kilómetros. De esta forma, se puede ver el consumo realizado por el vehículo a lo largo del tramo independientemente de la distancia recorrida, siendo una variable importante de minimizar.
- *BuenaConduccionOut*: porcentaje de conducción considerada de buena calidad a lo largo del tramo. Un porcentaje muy alto se entiende como que se ha realizado un tramo con una conducción muy buena. Los atributos RegularConduccionOut y MalaConduccionOut no son tan interesantes, ya que si por ejemplo hay un 60 % de buena conducción, directamente se sabe que el 40 % restante no es de buena calidad. La diferencia entre regular y mala conducción no importa mucho, es más interesante saber si ha tenido buena conducción o no.
- *AceleracionesBruscas3*: representa el tiempo en formato *horas:minutos:segundos* de aceleraciones bruscas realizado en el tramo. Una aceleración brusca es cuando se detecta una aceleración superior a $1,5 m/s^2$. Este atributo permite identificar el tipo de conducción de un conductor, ya que un valor alto puede hablar de una conducción agresiva o con errores. El valor más recomendable es 0, pero hay ocasiones en las que es inevitable realizar una aceleración brusca, como en una incorporación.
- *FrenadasBruscas3*: representa la cantidad de frenadas bruscas realizadas en el tramo, siendo por lo tanto un contador. Una frenada brusca es cuando se detecta una deceleración superior a $1,5 m/s^2$. Al igual que antes, el valor idóneo es 0, aunque hay ocasiones en las que por seguridad es necesario realizar una frenada brusca.
- *TiempoConduccionCrucero3*: este atributo representa el tiempo de conducción de crucero realizado en el tramo, expresado en formato *horas:minutos:segundos*. La conducción de crucero, también conocido como control de velocidad, es un sistema que fija la velocidad del vehículo, de forma que se mantenga constante sin necesidad de que el conductor presione el acelerador.

- *TiempoRal3*: este atributo representa el tiempo de uso del ralentí en el tramo, expresado en formato *horas:minutos:segundos*. Un motor está en ralentí cuando está en funcionamiento con el coche parado, con el embrague presionado o en punto muerto.

- *ConsumoRal3*: este atributo representa el consumo por culpa del ralentí en el tramo, expresado en litros.

- *RalInnec3*: representa el número de veces que se han realizado ralentíes innecesarios a lo largo de tramo, lo cual representa un tramo poco óptimo en caso de ser un valor grande.

- *CNoPredictiva3*: representa el número de veces que se ha realizado una conducción no predictiva en un tramo. Una conducción no predictiva se entiende como una situación en la que se realiza una acción brusca, sin predecir lo que ocurre en la carretera. Por ejemplo, cuando se ve un semáforo en rojo, lo que se debe hacer es levantar el pie del acelerador, y después, ir frenando suavemente con una cierta distancia de antelación. Sin embargo, si frenamos de manera muy brusca, esto contará como una conducción no predictiva. Por lo tanto, lo que se mide en este campo es la no anticipación a una conducción eficiente.

- *Seguridad, Eficiencia, Calidad* y *Ecología*: son atributos que representan del 0 al 100 el porcentaje de seguridad, eficiencia, calidad y ecología de la conducción del piloto a lo largo del tramo.

    - La seguridad evalúa la prevención de riesgos para el conductor y la seguridad vial.
    - La eficiencia valora la utilización de los recursos del vehículo.
    - La calidad evalúa los aspectos de calidad de una conducción, tales como el cumplimiento de normas de tráfico o el buen mantenimiento del vehículo.
    - La ecología puntúa los aspectos de la conducción relacionados con la protección del medio ambiente y contaminación.

- *MetrosAscendidos3* y *MetrosDescendidos3*: estos atributos representan los metros ascendidos y descendidos a lo largo del tramo. Son útiles para ver si en un tramo se ha realizado en el total un ascenso o un descenso.

### 3.2.2. Conversión de algunos atributos para ganar utilidad

Hay varios atributos a los que ha sido necesario aplicar distintas técnicas para que tengan una mayor utilidad. Los cambios que se realizaron son los siguientes:

- Los campos que están en formato *horas:minutos:segundos* (TiempoRal3, TiempoConduccionCrucero3 y AceleracionesBruscas3) deben ser pasados a segundos, para así poder operar con ellos y analizar sus distribuciones. Lo mismo se hizo también con MinutosTrayecto, que fue transformado de minutos a segundos.

- Al realizar comparaciones entre el campo MinutosTrayecto y el campo TiempoConduccion3seg, se han encontrado varias situaciones extrañas en las que el segundo campo era mayor que el primero, lo cual sería imposible según la definición indicada anteriormente de estos atributos. En el informe "Análisis del tiempo de conducción y de trayecto" enviado a la empresa se encuentra más información acerca de este suceso[12]. Por esta razón, finalmente no se usó este campo, y se obtuvo el tiempo total como la resta entre la fecha final (Fecha_EVT_Fin) y la fecha inicial del tramo (Fecha_EVT_Ini) en segundos. Este nuevo campo tiene el nombre de TiempoTotal.

- Lo más interesante de los atributos MetrosAscendidos3 y MetrosDescendidos3 no es en sí cuántos metros se ha ascendido o descendido, si no la resta entre ellos y el hecho de saber si en el tramo se ha realizado un ascenso o descenso desde la posición inicial. Para ello, se eliminaron estos dos campos y se sustituyeron por uno llamado Subida que tiene tres valores posibles: "Descenso" si los metros descendidos son superiores a los ascendidos, "Ascenso" si es al revés y "Se mantiene" si ambos son iguales.

- Por último, existen cuatro atributos (Seguridad, Calidad, Eficiencia y Seguridad) que valoran el rendimiento del conductor en el tramo. Sin embargo, realmente no es tan interesante ver cuánta puntuación hay en Seguridad o en Eficiencia, sino la valoración general del tramo, calculada a partir de estas cuatro, y usada como tal en otras tablas de la base de datos. En la tabla ValoracionesConductores_Perfiles se indica la importancia de cada columna dentro de la valoración final. En todos los perfiles hay un 25 de importancia para todas las métricas que indica que cada una vale un 25 %, por lo que la valoración es igual a $Seguridad * 0,25 + Eficiencia * 0,25 + Calidad * 0,25 + Ecología * 0,25$.

### 3.2.3. Análisis de distribuciones

Una vez decididos los atributos que se van a utilizar para cumplir los objetivos, hay que analizarlos para ver cómo se distribuyen sus valores, observando sus mínimos, máximos, y, sobre todo, aquellos que sean extremadamente grandes o pequeños, interfiriendo en el funcionamiento de la minería de datos.

Se ha analizado cada atributo por separado, para así poder decidir después a partir de qué valores se pueden eliminar filas. Cabe destacar que el valor total de filas con el que se ha hecho el análisis de distribuciones es 163694, obtenido tras realizar la eliminación de aquellas filas con algún valor en el campo Error1, de forma que se evite cualquier valor erróneo.

#### Id_Vehiculo

Este atributo es un identificador, por lo que obviamente no se han analizado sus posibles valores máximos o mínimos (es como si fuese un *String*). Lo que sí que es

[12]Cuyo enlace para poder leerlo es el siguiente: https://unicancloud-my.sharepoint.com/:b:/g/personal/jgm188_alumnos_unican_es/EfJKpSb14O5JpLJSN3Sy7EoBfGOoQodrirxRureNyBFXlg?e=pXToOF

interesante es ver los valores únicos sobre los valores totales, para hacerse una idea de cuántas veces se repite un vehículo.

Tras una pequeña consulta, se puede apreciar que hay un total de 3659 valores, por lo que se repiten bastantes vehículos. La repetición de vehículos es bastante equilibrada, siendo el más repetido el vehículo con identificador 1083 en un total de 146 veces. El décimo vehículo más repetido es el de identificador 593 con un total de 114 veces, lo cual no es una gran diferencia.

### Id_Conductor

Este atributo es un identificador al igual que antes. Por lo tanto, solo se analizan sus valores únicos y repeticiones. En este caso, hay un total de 3439 conductores distintos, por lo que tiene una distribución similar a la de los vehículos. Respecto a las repeticiones, hay una gran diferencia entre el conductor más repetido (el identificador 0, con 2411 repeticiones) y el resto de conductores. Por ejemplo, el segundo conductor más repetido es el que tiene un identificador de 1114, con un total de 146 repeticiones. Es probable que el identificador 0 represente algo que no sea un conductor real, como un "conductor de prueba", es decir, un valor de prueba para poder introducir valores. Otra posible razón es que represente que no se conoce cuál fue el conductor que realizó ese tramo.

### TiempoConduccion3seg

El atributo TiempoConduccion3seg es un valor continuo, por lo que se analizan tanto sus mínimos y máximos como sus percentiles, para ver cómo están distribuidos los valores del atributo. De esta forma, se puede deducir posteriormente cómo distribuir sus valores en diferentes rangos.

El mínimo valor es 0, lo que representa un tramo sin conducción, probablemente provocado por algún cambio de estado del motor que ha hecho que se genere un nuevo tramo. Por otra parte, el máximo es 11493538, que representa un viaje muy largo de más de 3192 horas, lo cual es prácticamente imposible, por lo que es un valor incorrecto. Respecto a los percentiles, los valores están representados tal y como se puede ver en la Tabla 2.

Tabla 2: Percentiles del atributo TiempoConduccion3seg

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 1,0 |
|---|---|---|---|---|---|
| **Valor** | 1035 | 2810 | 6053 | 12437 | 11493538 |

Como se puede ver, la diferencia entre los valores van aumentando cada vez más según aumenta el percentil; de 0,2 a 0,4 hay una diferencia de 1800 mientras que de 0,6 a 0,8 aumenta en unos 6000. Es decir, una gran mayoría de los valores se concentra alrededor de 1300 y 15000, como se podrá ver posteriormente en la gráfica, existiendo después varios valores extremos hasta los 11493538. De hecho, solo hay 2958 valores superiores a 40000. Esto se puede ver con más detalle en los últimos percentiles, siendo los indicados en la Tabla 3.

Tabla 3: Percentiles más altos del atributo TiempoConduccion3seg

| **Percentil** | 0,85 | 0,9 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 15609 | 21035,6 | 30086 | 35331,88 | 44984 | 11493538 |

Como se puede ver, el valor de 11493538 es un valor muy extremo y para nada representativo. En la Figura 3 se puede ver cómo están distribuidos los valores inferiores a 32400 (es decir, sin valores "extremos"), corroborando lo dicho anteriormente.

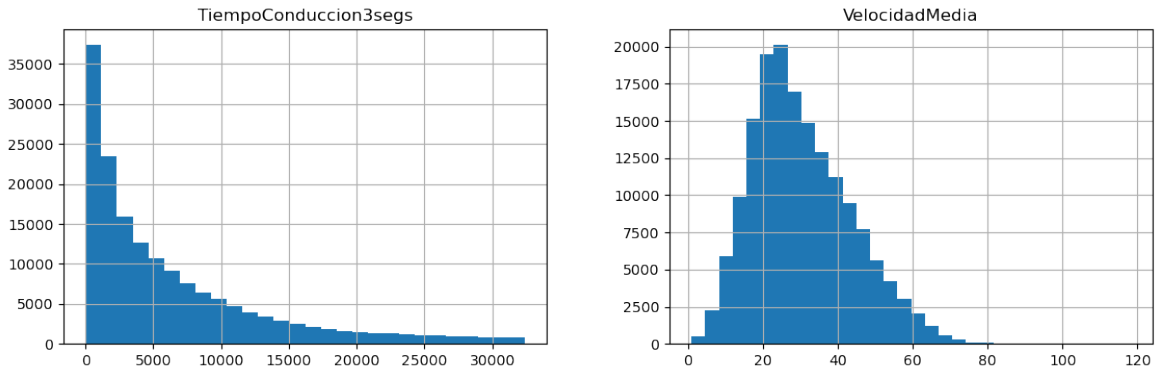


Figura 3: Gráficas de valores de los campos TiempoConduccion3segs y VelocidadMedia.

### VelocidadMedia

Al igual que antes, la velocidad media es también un valor continuo, así que se realiza el mismo análisis. El valor mínimo es 0,10, lo que probablemente viene de un tramo muy corto en el que apenas se haya movido el vehículo, mientras que el valor más alto es 118,2, siendo un valor anormalmente alto. En general, parece que las velocidades suelen ser similares y bastante bajas, excepto una pequeña cantidad de velocidades que son muy altas. Esto se puede ver en la Tabla 4.

Tabla 4: Percentiles del atributo VelocidadMedia

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 19,2 | 25,2 | 32,2 | 42 | 49,3 | 118,2 |

Como se puede ver, por debajo del percentil 0,9 los valores son inferiores a 50, mientras que el máximo es 118,2, por lo que los valores extremos son bastante inferiores en cantidad. En concreto, solo hay cuatro valores superiores a 100. En la Tabla 5 se presta especial atención a los percentiles más altos.

Tabla 5: Percentiles más altos del atributo VelocidadMedia

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 45,1 | 51,3 | 55,2 | 58,5 | 65,1 | 118,2 |

Se puede apreciar que la velocidad de 118,2 es algo circunstancial y que para nada es el valor habitual, ya que el valor del percentil de 0,99 es muchísimo más bajo y razonable (65,1 km/h). En la Figura 3 se puede observar la gráfica con las distribuciones.

### Distancia

Al igual que el anterior caso, es un dato continuo, así que se ha realizado el mismo análisis. El valor mínimo es 0, lo que indica que ha sido un tramo en el que no se ha recorrido ningún kilómetro de distancia, probablemente provocado por algún cambio de estado del motor como ya se ha mencionado anteriormente. Los percentiles se pueden ver en la Tabla 6.

Tabla 6: Percentiles del atributo Distancia

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 7,36 | 22,655 | 53,995 | 125,355 | 234,1385 | 2299,27 |

Al igual que en otras ocasiones, hay unos pocos valores que son claramente superiores al resto; concretamente, solo 203 valores son superiores a 1000. Sin embargo, los valores están equilibrados, por lo que se podría diferenciar entre distintos tipos de tramos de una manera relativamente fácil. Los percentiles más pequeños se pueden visualizar con detalle en la Tabla 7.

Tabla 7: Percentiles más pequeños del atributo Distancia

| **Percentil** | 0,01 | 0,02 | 0,03 | 0,05 | 0,1 | 0,15 |
|---|---|---|---|---|---|---|
| **Valor** | 0,26 | 0,46 | 0,66 | 1,115 | 2,745 | 4,86 |

Se puede apreciar que hay varios tramos con una distancia muy pequeña; de hecho, todo lo menor al percentil 0,1 tiene menos de 2,745 kilómetros. Estos tramos tan pequeños realmente no proporcionan información interesante, por lo que posteriormente se podrían eliminar del análisis aquellos que tengan una distancia inferior a la de este percentil. Por otra parte, los percentiles más altos han sido analizados también, recogidos en la Tabla 8.

Tabla 8: Percentiles más grandes del atributo Distancia

| **Percentil** | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|
| **Valor** | 271,6256 | 350,2935 | 432,8 | 605,247 | 2299,27 |

La diferencia entre el percentil más alto y el 0,99 es muy grande, por lo que claramente el valor máximo mencionado anteriormente es atípico y poco interesante. De igual manera, la diferencia entre el percentil 0,99 y el 0,92 es bastante considerable. La gráfica con aquellos valores positivos e inferiores a 1000 está en la Figura 4.

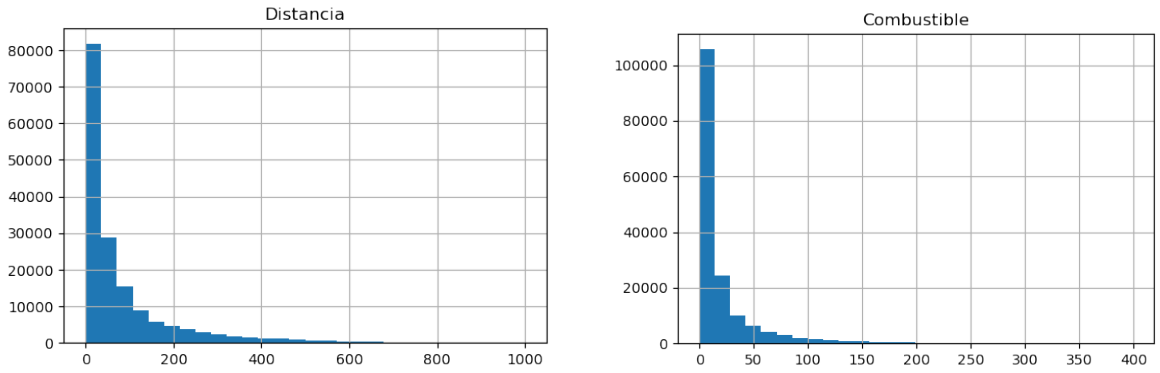


Figura 4: Gráficas de valores de los campos Distancia y Combustible.

### Combustible

El atributo Combustible es también un dato continuo, así que se realiza el mismo análisis. El valor mínimo es 0, por lo que es otra vez un valor que vendrá de un tramo donde no se haya recorrido ninguna distancia. El valor máximo es 716,611, lo que representa un valor bastante grande. Los percentiles de esta columna se pueden ver en la Tabla 9.

Tabla 9: Percentiles del atributo Combustible

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 1,859 | 5,215 | 11,8368 | 28,8272 | 56,9367 | 716,611 |

Al igual que en otras ocasiones, en el último 10 % se ven los valores más grandes. Concretamente, hay un total de 185 valores mayores de 300. Para ver esto con más detalle, se pueden observar los percentiles más altos en la Tabla 10.

Tabla 10: Percentiles más altos del atributo Combustible

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 39,88 | 67,26 | 90,192 | 118,86 | 181,9 | 716,611 |

Se puede confirmar que el valor más alto es un caso excepcional muchísimo más grande que el resto, sorprendiendo sobre todo el hecho de que el valor del percentil 0,99 es 181,90335, mucho menos que el del percentil 1, y aún así, mucho más que el percentil 0,92. La gráfica de combustible con valores mayores de 0 e inferiores a 400 está en la Figura 4.

### Consumo100

Este atributo es también un campo continuo. Su valor mínimo es 0, por lo que es posible que vuelva a representar un caso de tramo sin distancia recorrida. El valor máximo es de 17380, siendo un consumo bastante grande. Los percentiles de este atributo son los mostrados en la Tabla 11.

Tabla 11: Percentiles del atributo Consumo100

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 16,659 | 20,575 | 25,43 | 35,352 | 48,791 | 17380 |

En esta ocasión vuelve a pasar lo mismo de antes, en el último percentil se reúnen los valores extremadamente altos, mientras que en el resto los valores están equilibrados. En concreto, hay un total de 1416 valores más altos que 200. En la Tabla 12 se pueden ver los valores más altos.

Tabla 12: Percentiles más altos del atributo Consumo100

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 40,338 | 54,641 | 69,749 | 93,163 | 180,435 | 17380 |

Al igual que con el anterior atributo, el valor del percentil máximo es muchísimo más grande que el anterior percentil. Eso solo ocurre en muy pocas ocasiones, ya que el resto de valores de los percentiles son más realistas pese a seguir siendo altos. En la Figura 5 se encuentra la gráfica con valores positivos inferiores a 200.

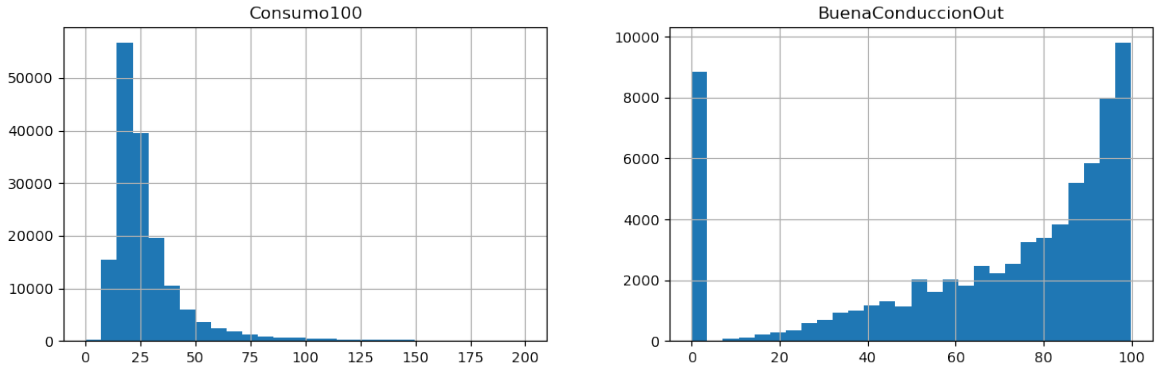


Figura 5: Gráficas de valores de los campos Consumo100 y BuenaConduccionOut.

### BuenaConduccionOut

Este valor representa el porcentaje del trayecto en el que se ha hecho una buena conducción, así que su mínimo valor debería ser 0 y su máximo valor debería ser 100.

Analizando mínimos y máximos, se puede ver que esto se cumple. Además, la mediana es 100, por lo que más de la mitad de los tramos tienen una conducción absolutamente buena. Respecto a los percentiles, me he centrado en los primeros (cuando la conducción no es tan buena), siendo los representados en la Tabla 13.

Tabla 13: Percentiles del atributo BuenaConduccionOut

| **Percentil** | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 48,43 | 75,9 | 90,5 | 97,8 | 100,0 | 100,0 |

En general, pese a no ser de un 100 %, la gran mayoría de los tramos tienen una conducción muy buena. La gráfica con los valores inferiores a 100 se encuentra en la Figura 5.

### AceleracionesBruscas3

Para este atributo se ha aplicado el mismo tipo de análisis previamente realizado. Su valor mínimo es 0, representando un tramo perfecto sin ninguna aceleración brusca, mientras que su valor máximo es 12307, lo que sería casi 3 horas y media, siendo una cantidad excesivamente grande. Los percentiles son los indicados en la Tabla 14.

Tabla 14: Percentiles del atributo AceleracionesBruscas3

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 23 | 65 | 148 | 340 | 594 | 12307 |

Por lo que se puede confirmar que el valor de 12307 es muy alto, ya que el resto de percentiles tienen valores más razonables. En concreto, hay un total de 1519 tramos que superan los 2500 segundos de aceleraciones bruscas. Los percentiles más altos se pueden visualizar en la Tabla 15.

Tabla 15: Percentiles más altos del atributo AceleracionesBruscas3

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 436 | 698 | 975 | 1393 | 2418,07 | 12307 |

Otra vez más se observa lo mismo, el percentil más alto siempre contiene un valor exageradamente alto, mientras que el resto contienen valores algo más razonables. La gráfica con todos aquellos valores menores a 2500 está en la Figura 6.

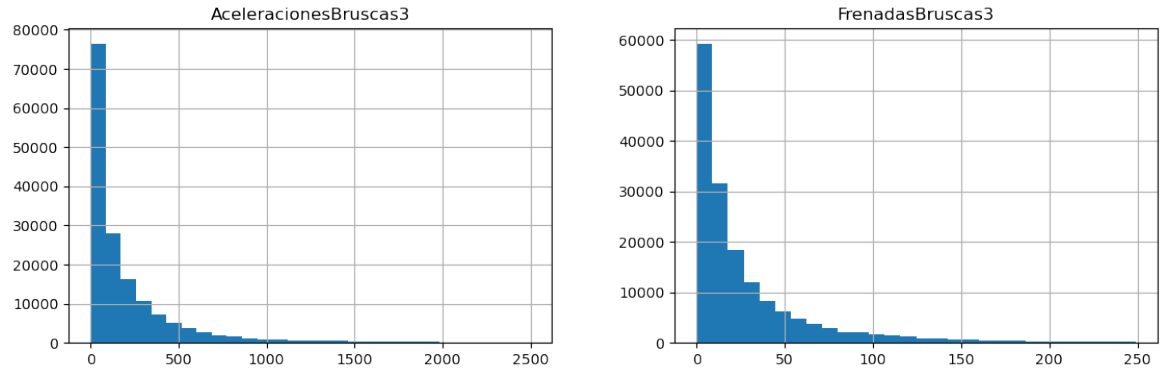


Figura 6: Gráficas de valores de los campos AceleracionesBruscas3 y FrenadasBruscas3.

### FrenadasBruscas3

Es un atributo del mismo estilo que el anterior, solo que es un contador en vez de ser un atributo temporal. Pese a ello, se ha realizado también el mismo análisis. Su valor

mínimo es 0, representando un tramo perfecto sin ninguna frenada brusca, mientras que su valor máximo es 3344, lo que parece una cantidad aún más exageradamente grande. Los percentiles son los mostrados en la Tabla 16.

Tabla 16: Percentiles del atributo FrenadasBruscas3

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 4 | 10 | 21 | 46 | 81 | 3344 |

En este caso, la diferencia entre el resto de percentiles y el último es excesivamente grande, lo que no parece del todo razonable. En concreto, solo hay 1530 valores que superan las 250 frenadas bruscas, pese a que el valor máximo es de 3344. Se han examinado también los percentiles más altos en la Tabla 17 para verlo más en detalle.

Tabla 17: Percentiles más altos del atributo FrenadasBruscas3

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 59 | 93 | 121 | 157 | 245 | 3344 |

Otra vez se puede observar como el percentil más alto tiene un valor muy superior al resto. El resto de valores son más razonables, aunque se podrían directamente eliminar aquellos valores a partir del percentil 0,92. En la Figura 6 está la gráfica con los valores inferiores a 250.

### TiempoConduccionCrucero3

Este atributo es también un dato continuo. Su valor mínimo es 0, y de hecho, la mediana también es 0, por lo que, aproximadamente, la mitad son ceros. Por lo tanto, se puede decir que en la mayoría de los tramos no se usa la conducción de crucero. Su valor máximo es 55635. Para los percentiles, se han mirado aquellos casos que no son cero, siendo los mostrados en la Tabla 18.

Tabla 18: Percentiles del atributo TiempoConduccionCrucero3

| **Percentil** | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|
| **Valor** | 11 | 263 | 907 | 2629 | 55635 |

Al igual que en el resto de atributos, la diferencia en el último percentil es abismal. En concreto, hay 4948 valores superiores a 7500. Se ha analizado con más detalle los valores más grandes en la Tabla 19.

Tabla 19: Percentiles más grandes del atributo TiempoConduccionCrucero3

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 1498 | 3457 | 5393 | 7525 | 12064 | 55635 |

Se puede comprobar que el valor del percentil 1 es excesivamente grande incluso comparado con el del percentil 0,99. Se podrían eliminar aquellos valores superiores a 3600 ya que son muy grandes (una hora de conducción de crucero). La gráfica obtenida con los valores inferiores a 7000 y superiores a cero está en la Figura 7.

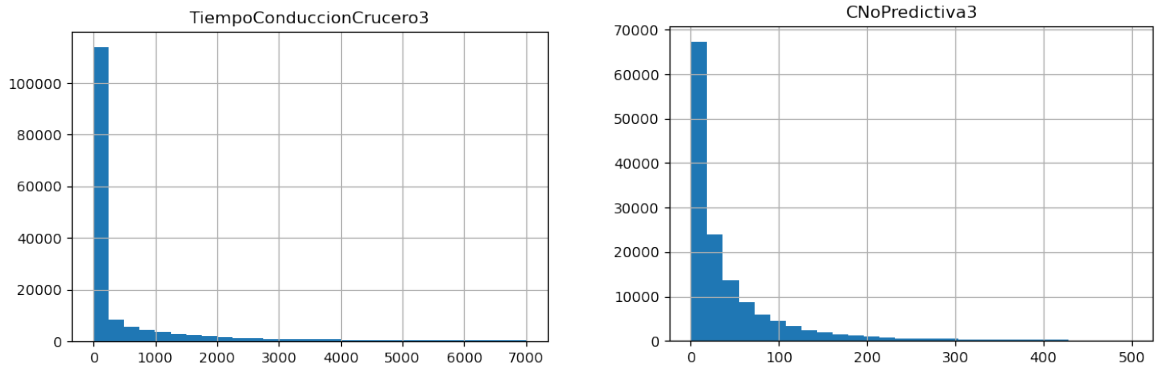


Figura 7: Gráficas de valores de los campos TiempoConduccionCrucero3 y CNoPredictiva3.

### CNoPredictiva3

Este atributo es también un dato continuo. Su valor mínimo es 0, lo cual representa un tramo sin conducciones no predictivas. El valor máximo es 2105, siendo un valor grande para ser un contador. Los percentiles son los mostrados en la Tabla 20.

Tabla 20: Percentiles del atributo CNoPredictiva3

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 2 | 9 | 25 | 64 | 121 | 2105 |

Como se puede apreciar, aquí también tiene el último percentil un valor muy grande respecto al resto. Hay unos 1673 tramos con un valor de conducción no predictiva superior a 500. Esto se puede ver más detenidamente con los percentiles más altos recogidos en la Tabla 21.

Tabla 21: Percentiles más altos del atributo CNoPredictiva3

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 85 | 143 | 199 | 276 | 505 | 2105 |

El valor de 2105 es claramente muy superior al resto, teniendo en cuenta que le saca 1600 al valor del percentil 0,99. En la Figura 7 se adjunta la gráfica con los valores inferiores a 500.

### TiempoRal3

Este atributo es también un dato continuo. Su valor mínimo es 0, mientras que su valor máximo es 181358. Si se analizan los percentiles, se puede apreciar que el último

vuelve a tener un valor claramente superior al resto, como se puede observar en la Tabla 22.

Tabla 22: Percentiles del atributo TiempoRal3

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 977 | 2003 | 3485 | 6446 | 9900 | 181358 |

En concreto, hay un total de 2985 tramos con un valor superior a 20000. Respecto a los percentiles más altos, los valores son los recogidos en la Tabla 23.

Tabla 23: Percentiles más altos del atributo TiempoRal3

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 7815 | 11173,12 | 13900 | 17052,21 | 23637,21 | 181358 |

El valor de 181358 es claramente muy exagerado. En la Figura 8 se adjunta la gráfica con los valores inferiores a 20000.

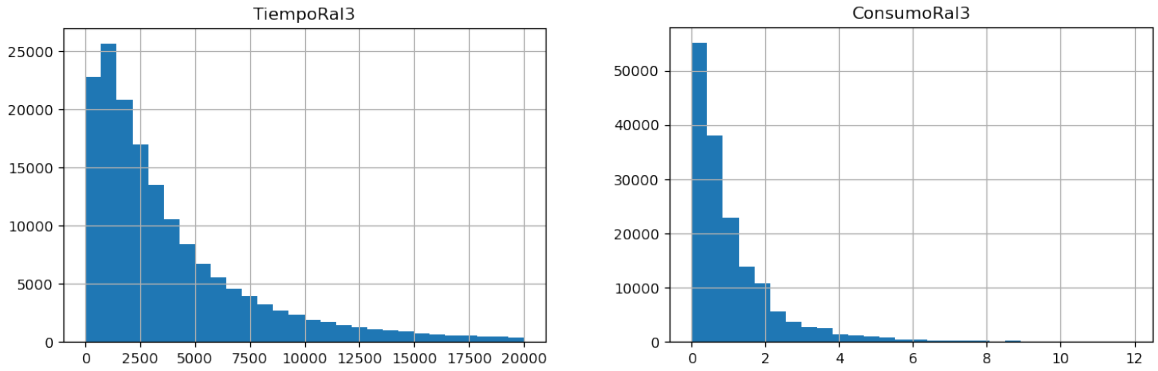


Figura 8: Gráficas de valores de los campos TiempoRal3 y ConsumoRal3.

### ConsumoRal3

Este atributo es, otra vez, un dato continuo. Su valor mínimo es 0 y 0,7 es su mediana, lo que hace ver que, por lo general, tiene valores pequeños. Su valor máximo es 154,5, que destaca claramente respecto al resto de valores. Sus percentiles son los recogidos en la Tabla 24.

Tabla 24: Percentiles del atributo ConsumoRal3

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 0,3 | 0,5 | 0,9 | 1,7 | 2,6 | 154,5 |

El valor de 154,5 destaca por mucho respecto al resto de valores. De hecho, solo hay 770 tramos con un valor superior a 14. La diferencia entre el percentil 0,9 y el 1

parece bastante notoria, así que se han analizado otra vez los percentiles más altos, recogidos en la Tabla 25.

Tabla 25: Percentiles más altos del atributo ConsumoRal3

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 2 | 3 | 3,8 | 4,9 | 8,8 | 154,5 |

El valor máximo vuelve a ser exageradamente grande respecto al resto. La gráfica con valores inferiores a 12 está en la Figura 8.

### RalInnec3

El atributo RalInnec3 representa la cantidad de ralentíes innecesarios realizados en el tramo, por lo que su valor mínimo podría ser 0. Es también un dato continuo. Analizando el atributo, se puede confirmar lo dicho, ya que su valor mínimo es 0, mientras que su valor máximo es 150. Sus percentiles son los recogidos en la Tabla 26.

Tabla 26: Percentiles del atributo RalInnec3

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 3 | 6 | 10 | 18 | 27 | 150 |

Como ha pasado hasta ahora en casi todos los atributos, en el último percentil el valor es mucho más grande que en el resto de percentiles. En concreto, solo 771 tramos superan los 70 ralentíes innecesarios. Observando los percentiles más altos recogidos en la Tabla 27, se puede ver lo exagerado que es este valor.

Tabla 27: Percentiles más altos del atributo RalInnec3

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 21 | 30 | 37 | 44 | 60 | 150 |

En la Figura 9 se encuentra la gráfica con los valores inferiores a 70.

### Valoración

Su valor mínimo es 0, es decir, un tramo nefasto, y su valor máximo es 100, lo que vendría a ser un tramo perfecto. Cabe destacar que la mediana es 93,952525, así que la mayoría de los tramos tienen una buena valoración. Los percentiles son los mostrados en la Tabla 28.

Tabla 28: Percentiles del atributo Valoración

| **Percentil** | 0,1 | 0,2 | 0,3 | 0,4 | 0,6 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 15,88 | 35,53 | 68,84 | 90,15 | 98,48 | 100 |

La gráfica de los valores se encuentra en la Figura 9.

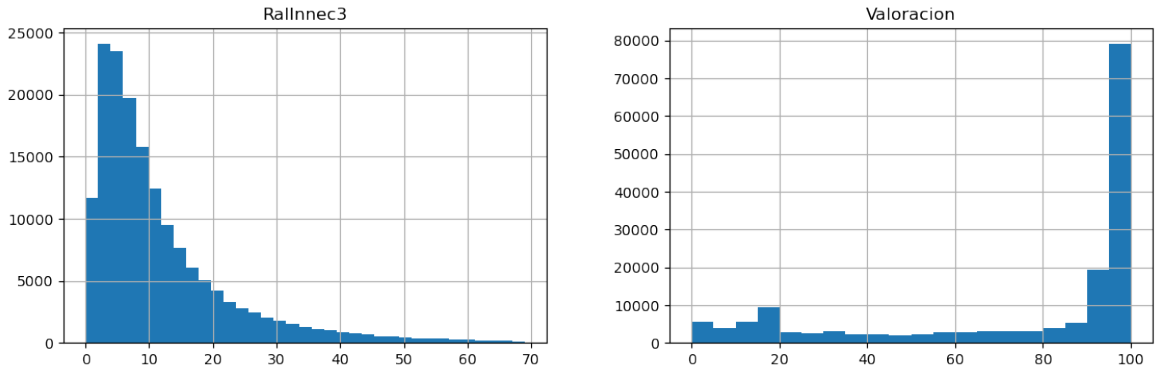


Figura 9: Gráficas de valores de los campos RalInnec3 y Valoracion.

### TiempoTotal

Este atributo es también un dato continuo. Su valor mínimo es 0, provocado probablemente por un tramo entre distintos estados de motor. Por otra parte, su valor máximo es 312959, un valor bastante alto comparado con la mediana, que es 35784. Los valores obtenidos a partir de los percentiles son los indicados en la Tabla 29.

Tabla 29: Percentiles del atributo TiempoTotal

| **Percentil** | 0,2 | 0,4 | 0,6 | 0,8 | 0,9 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 15688 | 27851,2 | 44510,8 | 70035,8 | 89835,7 | 312959 |

En esta ocasión vuelve a ocurrir lo mismo que en atributos anteriores: el último percentil tiene un valor superior al resto. En concreto, hay 4911 tramos con un valor en TiempoTotal superior a 170000. Los últimos percentiles del atributo se encuentran en la Tabla 30.

Tabla 30: Percentiles más altos del atributo TiempoTotal

| **Percentil** | 0,85 | 0,92 | 0,95 | 0,97 | 0,99 | 1,0 |
|---|---|---|---|---|---|---|
| **Valor** | 81045,05 | 97216,56 | 137064,15 | 169065,15 | 237971,4 | 312959 |

La gráfica de la columna con los valores inferiores a 70000 se puede ver en la Figura 10.

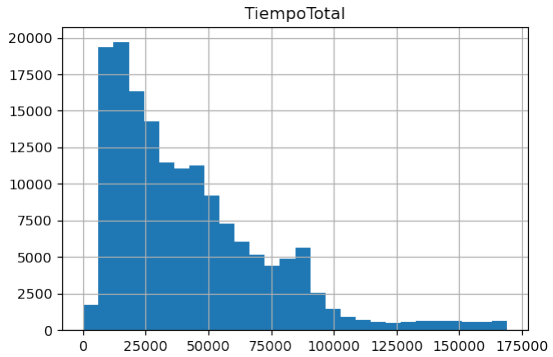


Figura 10: Gráficas de valores del campo TiempoTotal.

#### Subida

Por último, está el atributo Subida, que ha sido calculado a partir de las columnas MetrosAscendidos2 y MetrosDescendidos2 de la tabla de tramos.

En este caso, el único análisis que se puede realizar es ver cuántas repeticiones tiene cada uno de los tres valores. El ascenso y descenso están bastante equilibrados con 82300 y 80913 repeticiones respectivamente, mientras que solo 7840 tramos se mantienen igual. Hay más tramos en descenso que en ascenso, pero por una diferencia bastante leve. Por otra parte, es improbable que en un tramo no se ascienda ni tampoco se descienda, por lo que es entendible que haya pocos tramos con "Se mantiene".

# 4. Preparación y gestión de los datos

Una vez vistos, entendidos y analizados los campos seleccionados de la tabla Tramos, se ha pasado a eliminar filas y a procesarlas, de forma que sean útiles a la hora de aplicar las técnicas de minería de datos.

## 4.1. Preparación de los datos

### 4.1.1. Reducción de filas

Como se ha podido ver en el anterior apartado, en la gran mayoría de los atributos existen valores muy extremos que podrían dificultar la aplicación de técnicas de minería de datos. Muchos de ellos parecen inverosímiles a primera vista; por ejemplo, parece difícil de creer que un tramo se haya hecho con una velocidad media de 118,2 km/h o que en un tramo se hayan realizado unas 3344 frenadas bruscas.

Sin embargo, según la definición de tramo indicada en el anterior apartado, estos datos han podido darse en situaciones excepcionales. Por ejemplo, en el caso del tramo con velocidad media muy alta, ha podido ocurrir en un tramo que se haya registrado con una duración muy corta, donde haya habido un cambio de estado en el vehículo que haya provocado este corte. En el segundo caso, puede ser un vehículo que haya realizado un viaje de una distancia muy larga, donde no se ha apagado el motor en los descansos, por lo que realmente serían varios tramos que se han contabilizado como

uno solo. De hecho, localizando los datos de forma concreta, se puede apreciar que esto es así, teniendo el primero una duración de tiempo de conducción corta, y el segundo teniendo una duración muy larga. En la Tabla 31 se pueden ver estos valores.

Tabla 31: Ejemplos de atributos con valores inverosímiles, acompañados del tiempo de conducción que los justifican.

| **TiempoConduccion3seg** | **VelocidadMedia** |
|---|---|
| 2625 | 118.2 |

| **TiempoConduccion3seg** | **FrenadasBruscas3** |
|---|---|
| 83290 | 3344 |

Para evitar datos "extraños" como estos, han sido filtrados con el objetivo de eliminar los más extremos. Como hemos podido ver en el análisis del apartado anterior, estos valores se encuentran en los percentiles más altos de las columnas, así que, para evitarlos de forma segura, se han eliminado aquellos que sean superiores al percentil 0,88, a partir del cual se empiezan a ver valores muy exagerados, con la excepción de atributos como Valoracion o BuenaConduccionOut, donde no se han apreciado valores de este tipo. Por otra parte, en el caso del atributo Distancia, también se han eliminado aquellas filas menores al percentil 0,1, ya que tramos con menos de 3 kilómetros son muy poco interesantes.

## 4.2. Medias por unidad de los atributos

Atributos como el número de frenadas bruscas o el de ralentíes innecesarios son muy interesantes, sin embargo, no es lo mismo hacer 15 frenadas bruscas en un pequeño tramo de 7 kilómetros que hacer la misma cantidad en 300 kilómetros; en el primer caso es una cantidad claramente muy alta, mientras que en el segundo caso es una cantidad que puede ser considerada baja. Por ello, se ha calculado la media por unidad para ciertos atributos, obteniendo así valores que no dependen de la longitud del tramo, siendo más interesantes.

Los nuevos atributos que se han conseguido de esta forma son los siguientes:

- Frenadas bruscas por kilómetro: obtenido directamente como la división entre el campo FrenadasBruscas3 y el campo Distancia.
- Aceleraciones bruscas por segundos de conducción, o dicho de otro modo, porcentaje de tiempo de conducción en el que se hacen aceleraciones bruscas: obtenido directamente como la división entre el campo AceleracionesBruscas3 y el campo TiempoConduccion3seg.
- Conducción de crucero por tiempo de conducción, o dicho de otro modo, porcentaje de tiempo de conducción en el que se usa la conducción de crucero: obtenido directamente como la división entre el campo TiempoConduccionCrucero3 y el campo TiempoConduccion3seg.

- Motor en ralentí por tiempo total, o dicho de otro modo, porcentaje del tiempo total en el que el motor está en ralentí: obtenido directamente como la división entre el campo TiempoRal3 y el campo TiempoTotal. En muchas ocasiones, cuando el motor está en ralentí no cuenta como conducción, así que esa es la razón por la que se ha elegido el tiempo total en este caso.

- Número de acciones o conducciones no predictivas por kilómetro, siendo la división entre el campo CNoPredictiva3 y el campo Distancia.

- Consumo en ralentí por tiempo total: obtenido directamente como la división entre el campo ConsumoRal3 y el campo TiempoTotal.

- Ralentíes innecesarios por tiempo total: obtenido directamente como la división entre el campo RalInnec3 y el campo TiempoTotal.

Estos, unidos a los atributos Consumo100, VelocidadMedia, Valoración y Subida, son los atributos a los que a partir de ahora me referiré como "datos independientes", siendo datos representativos e interesantes.

## 4.3. División final de atributos continuos en rangos

A la hora de poder aplicar la minería de reglas de asociación, ha sido necesario dividir previamente los atributos continuos en rangos. Se han considerado tres opciones para ello:

- Lo primero y más sencillo es dividir el atributo en tres rangos equilibrados en cuanto a cantidad (división por percentiles). Puede parecer la manera más justa de hacerlo, sin embargo, da lugar a divisiones poco interesantes. Como se ha visto en las gráficas del análisis, hay algunos atributos que tenían valores que se repetían muy frecuentemente, dando lugar a grupos donde el número de valores es bajo y descompensado respecto al de los otros. Por ejemplo, en el caso de Valoracion o BuenaConduccionOut, el percentil 50 engloba valores desde cero hasta 100, lo que es un grupo muy poco representativo, ya que incluye todo el rango de valores. Por lo tanto, esta opción quedaba totalmente descartada, al no proporcionar grupos interesantes.

- La segunda opción es dividir los atributos en intervalos iguales. Por ejemplo, en el caso de Valoración, sería lógico dividirlo en dos grupos, uno con valores más pequeños de 50 y otro con valores iguales o superiores. Lo mismo se podría hacer con otros atributos siguiendo las gráficas indicadas anteriormente. Sin embargo, si bien ahora los grupos son más útiles, salen muy desequilibrados, con cantidades de valores muy diferentes entre ellos.

- Está claro que entre grupos representativos pero desbalanceados, o grupos balanceados pero nada interesantes, es más recomendable lo primero ante la gran cantidad de filas de las que se disponen, lo que permite una buena variedad. Por ello, finalmente se ha usado algo que da unos resultados más parecidos a lo segundo, pero de una forma algo más balanceada al no tener que determinar uno

mismo los límites. Se ha utilizado *k-means* para calcular los rangos, un algoritmo que permite aplicar *clustering*, descrito en el sub-apartado 5.3.

# 5. Análisis

El siguiente paso es aplicar las distintas técnicas de minería de datos para el análisis, por lo que en este apartado se explicará la parte teórica de los modelos aplicados. Tal y como se ha indicado al inicio de la memoria, es bastante habitual que sea necesario volver a la fase anterior tras haber aplicado alguna de las técnicas, ya sea para refinar las divisiones de datos o para eliminar alguna fila que pudiese generar valores atípicos.

## 5.1. Reglas de asociación

El minado de reglas de asociación ha permitido llevar a cabo el primero de los tres objetivos. Esta técnica es capaz de encontrar relaciones entre distintos atributos representadas como reglas, donde una regla es una implicación de la forma *si* $X$ *entonces* $Y$, donde $X$ e $Y$ son conjuntos de atributos, cada uno con un valor determinado. Las reglas se representan directamente como $X \rightarrow Y$ donde a $X$ se le llama el antecedente y a $Y$ el consecuente. Esto significa que, cuando ocurre el antecedente, entonces el consecuente es probable que ocurra también.

Dentro de las reglas, hay dos conceptos que definen el interés o importancia de una regla: el "soporte" y la "confianza". El soporte de $X$ (definido como $sup(X)$) se especifica como la frecuencia de aparición del evento $X$ dentro del conjunto de observaciones. Por otra parte, la confianza de una regla $X \rightarrow Y$ se define como lo indicado en la Ecuación (1).

$$conf(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \tag{1}$$

Este atributo indica en cuántas ocasiones es cierta la regla, independientemente de si el *itemset* con el que se ha generado es muy frecuente o no. A la hora de buscar reglas, se indicarán una confianza mínima (representada como $\gamma$) y un soporte mínimo (representado como $\tau$), de forma que las reglas que se obtengan son las especificadas en la Ecuación (2).

$$X \rightarrow Y \text{ será una regla obtenida si: } \begin{array}{l} conf(X \rightarrow Y) \geq \gamma \\ sup(X \cup Y) \geq \tau \end{array} \tag{2}$$

Por otra parte, las reglas también tienen otro indicador de intensidad para conocer lo representativas que son realmente, llamado *lift*. Esta medida de intensidad, respecto a una posible regla $X \rightarrow Y$, indica el grado en el que Y tiene más probabilidades de aparecer cuando aparece X. Si es mayor que 1, entonces significa que $Y$ tiene más probabilidades de aparecer cuando $X$ aparece, si es menor que 1, significa todo lo contrario. La fórmula está definida en la Ecuación (3):

$$lift(X \rightarrow Y) = \frac{conf(X \rightarrow Y)}{sup(Y)} = \frac{sup(X \cup Y)}{sup(X) * sup(Y)} \tag{3}$$

El algoritmo utilizado para minar las reglas será *Apriori*, que se encarga de generar conjuntos frecuentes de ítems, obteniendo las reglas a partir de ellos [6]. El funcionamiento es el siguiente: inicialmente, se forman conjuntos de un ítem, y se mantienen aquellos que superen el soporte mínimo indicado. Posteriormente, se forman parejas con los ítems que hayan quedado, y se calculan los nuevos soportes para cada una. De igual manera que antes, aquellas que no lleguen al soporte mínimo serán eliminadas. Después, se formarán tríos con las parejas que hayan pasado el filtro, uniendo aquellas que compartan el primer ítem. Como todos los subconjuntos de un *itemset* frecuente deben ser también frecuentes, se comprueba que los demás subconjuntos de cardinal 2 sean frecuentes. De ser así, se calculará el soporte del nuevo conjunto de cardinal 3 para verificar si supera el soporte mínimo. En definitiva, se generará un grupo de tamaño $k$ a partir de dos conjuntos de tamaño $k-1$ que sean frecuentes y que compartan los $k-2$ primeros elementos, se comprobará que el resto de subconjuntos de tamaño $k-1$ son frecuentes y, si es verdad, se calculará el soporte de este nuevo conjunto. Esto se realizará sucesivamente hasta que no se puedan formar nuevos conjuntos por falta de elementos o cuando ninguno de los nuevos candidatos superen el soporte mínimo.

### 5.1.1. Reglas raras

Además de las reglas de asociación al uso, también se han minado "reglas raras". La idea detrás de esto es encontrar reglas en conjuntos de ítems que no son muy frecuentes, o dicho de otro modo, buscar aquellas que tengan un soporte bajo pero una confianza alta, y que son también interesantes. Esto no se puede hacer con la implementación de *Apriori* que hay en *Python*, ya que no se puede especificar un soporte máximo, y recoger todas las reglas a partir de un soporte muy bajo es algo prácticamente impensable debido al alto número que se generaría, y al tiempo necesario para que terminase la ejecución. Para ello, ha sido necesario utilizar un programa externo llamado *SPMF*, el cual ha sido llamado a través de un *Python wrapper*, o dicho de otro modo, un puente entre *Python* y el propio *jar*. Dentro de este programa, se ha empleado lo que se conoce como *Perfectly Sporadic Association Rules*, que utiliza el algoritmo *Apriori Inverse* [7].

El algoritmo de *Apriori Inverse* está basado en el Apriori anteriormente explicado, pero se puede especificar un soporte máximo para las reglas. A partir de ese soporte máximo, las reglas que se generen son ignoradas. Este algoritmo permite obtener dos tipos de reglas: las *perfectly sporadic rules*, que son reglas que contienen solo valores de un *itemset* donde todos tienen que tener el soporte inferior al máximo indicado, y las *imperfectly sporadic rules*, donde se permiten que haya reglas que contenga algún elemento que tenga un soporte algo superior al indicado. Se ha trabajado con las primeras reglas ya que son más interesantes, además de que el soporte máximo es lo suficientemente grande como para no querer algo que sea ligeramente superior.

Las *perfectly sporadic rules* se definen en la Ecuación (4), donde $\tau$ es el soporte máximo indicado y $\gamma$ la confianza indicada.

$$X \to Y \text{ es una regla esporádica perfecta si: } \begin{array}{l} conf(X \to Y) \geq \gamma \\ sup(x) < \tau, \forall x \in X \cup Y \end{array} \tag{4}$$

El funcionamiento del algoritmo es el mismo que el de *Apriori*, pero con una modificación. Cuando se forman los conjuntos iniciales de un ítem, se comprueba que cada uno de ellos no supera el máximo soporte indicado, en caso contrario, son eliminados. Al expandir los conjuntos, el soporte nunca va a poder aumentar, por lo que con realizar esa comprobación inicial será suficiente para obtener conjuntos de ítems inferiores al máximo soporte establecido.

## 5.2. Árboles de decisión

El segundo objetivo es crear un modelo predictivo que permita clasificar un tramo como bueno o malo en función de sus atributos. Para ello, se han utilizado los árboles de decisión. La razón detrás de esta elección es el mayor beneficio que tienen los árboles de decisión: permiten la visualización del modelo generado, pudiendo identificar las reglas o decisiones generadas que determinan las clasificaciones. De igual manera, tiene un funcionamiento sencillo, lo que hace que sea muy intuitivo.

Un árbol de decisión es un modelo predictivo que plasma de manera visual una serie de condiciones y los posibles resultados que se obtienen en base a si esas condiciones se cumplen o no. Tiene tres elementos importantes:

- Los nodos de decisión, que muestran una condición que puede o no cumplirse. Por ejemplo, que el valor del atributo Combustible sea menor o igual que 14,4.
- Los nodos finales, que se diferencian de los anteriores en que no tienen una condición escrita, sino que directamente muestran el resultado final.
- Las flechas, que unen un nodo de decisión con otro nodo de decisión o con un nodo final. Cada nodo de decisión tendrá dos o más flechas apuntando cada una a otro nodo, una por cada posible respuesta a la condición del nodo de donde se generan.

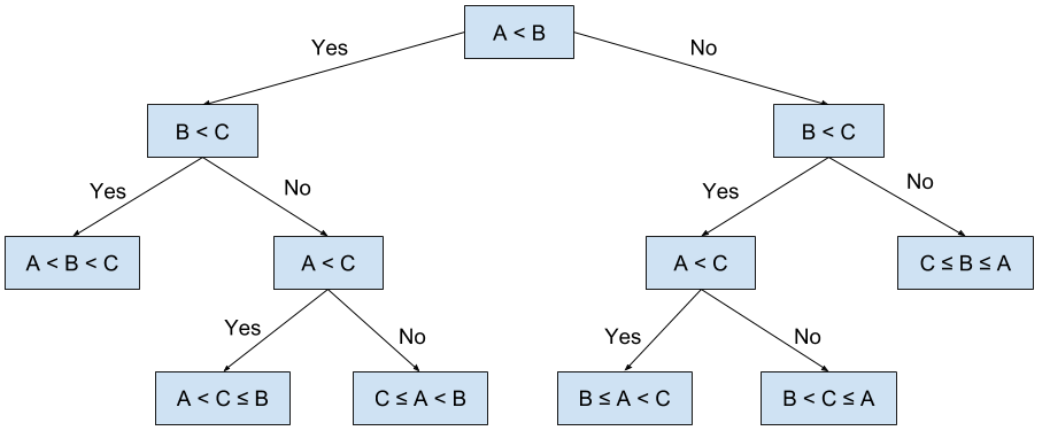


Figura 11: Ejemplo de un árbol de decisión.

En la Figura 11 se puede apreciar un ejemplo de cómo se vería un árbol de decisión. Los nodos de decisión muestran una condición y tienen dos flechas apuntando cada una a otro, la de izquierda si se cumple y la de la derecha si no se cumple. Los nodos finales muestran el resultado final, en este caso, cómo quedarían los tamaños de $A$, $B$ y $C$. A la hora de dar una predicción para un tramo, se seguirán las flechas en función de las condiciones de los nodos, hasta llegar al nodo final donde se obtenga la clase predicha. En este caso, estamos ante un árbol binario que solo admite dos hijos por cada nodo, pero en otro tipo de árboles los nodos podrían tener más de dos hijos.

Concretamente, la implementación que se ha usado del clasificador de árboles de decisión de *sklearn* utiliza el algoritmo CART (*Classification and Regression Trees*) [8]. Este algoritmo genera árboles binarios como el mostrado en la Figura 11, dividiendo cada nodo en dos nuevos nodos en caso de ser posible. El funcionamiento es el siguiente:

- El árbol empieza con un único nodo que representa todas las filas del conjunto de entrenamiento.
- A la hora de dividir el nodo en otros dos nodos "hijos", el algoritmo se basa en el criterio de división (*splitting criterion*), que le indica qué atributo debe utilizarse para hacer la división y qué subconjuntos se generarán. Con el criterio de división lo que se intenta es que cada partición sea lo más pura posible, entendiéndose que una partición es pura si todas las filas que contiene pertenecen a la misma clase. Hay dos criterios de división utilizados frecuentemente en los árboles.

  Uno de ellos es el criterio de *gini*, que utiliza la fórmula de la Ecuación (5).

$$Gini = 1 - \sum_{i=1}^{n} (P_i)^2 \ , \tag{5}$$

  siendo $P_i$ la probabilidad de que un elemento cualquiera sea clasificado como miembro de una determinada clase $i$, por lo que directamente se reemplaza por la frecuencia de esa clase en el conjunto de datos. De esta forma, este criterio indica la posibilidad de que se clasifique erróneamente cualquier elemento de los datos si se hace de forma aleatoria.

  El otro criterio es la entropía, cuya fórmula se muestra en la Ecuación (6).

$$Entropy = -\sum_{i=1}^{n} P_i * \log_2 P_i \ , \tag{6}$$

  donde $P_i$ vuelve a ser la misma probabilidad de antes. En este caso, este criterio mide el desorden o "impuridad" de un grupo de características. En el caso de CART, se utiliza el índice de *gini*, mientras que el criterio de *entropy* se usará en un modelo que se explicará más tarde.
- Este proceso se realiza recursivamente hasta que se llegue al número de profundidad y hojas determinado o cuando todas las filas de un subconjunto tengan la misma clase. En este caso, especificaremos la profundidad y el número de hojas que queremos como parámetros del árbol.

Para crear el árbol, es necesario dividir los datos en datos de entrenamiento y datos de test. Los datos de entrenamiento son usados para entrenar el modelo y elegir sus mejores parámetros (número de hojas y profundidad), y los datos de test indicarán la precisión del modelo. En este caso se han realizado varias ejecuciones a través de la clase *GridSearchCV* de *sklearn*, para así encontrar esos mejores parámetros. Esta clase permite realizar esa búsqueda, iterando con distintos conjuntos de valores de los parámetros, y utilizando la validación cruzada para ir calificando cada uno de ellos.

La validación cruzada (*cross-validation*) [9] consiste en ir dividiendo en $K$ formas distintas los datos de entrenamiento iniciales en unos nuevos datos de entrenamiento y en datos de validación o test, de forma que entrena cada árbol con los primeros datos, y después comprueba su fiabilidad con los datos de validación. Por ejemplo, si se quiere hacer *cross-validation* con cinco divisiones distintas, lo que se hará es realizar una división de los datos de entrenamiento originales en cinco subconjuntos. Cada uno de esos subconjuntos será utilizado una vez como conjunto de test para la búsqueda de parámetros, mientras que el resto de subconjuntos se unirán para formar el conjunto de entrenamiento. Por cada subconjunto usado como test se calculará su puntuación de precisión. Una vez se hayan hecho las cinco pruebas, y se tenga por lo tanto cinco puntuaciones de precisión, se calcula la media, obteniendo la puntuación general, que será la que determine qué parámetros son los mejores.

Finalmente, se usarán los datos de test que se habían reservado al principio para poder visualizar la precisión del árbol con sus mejores parámetros. El ejemplo anteriormente descrito se puede ver en la Figura 12.

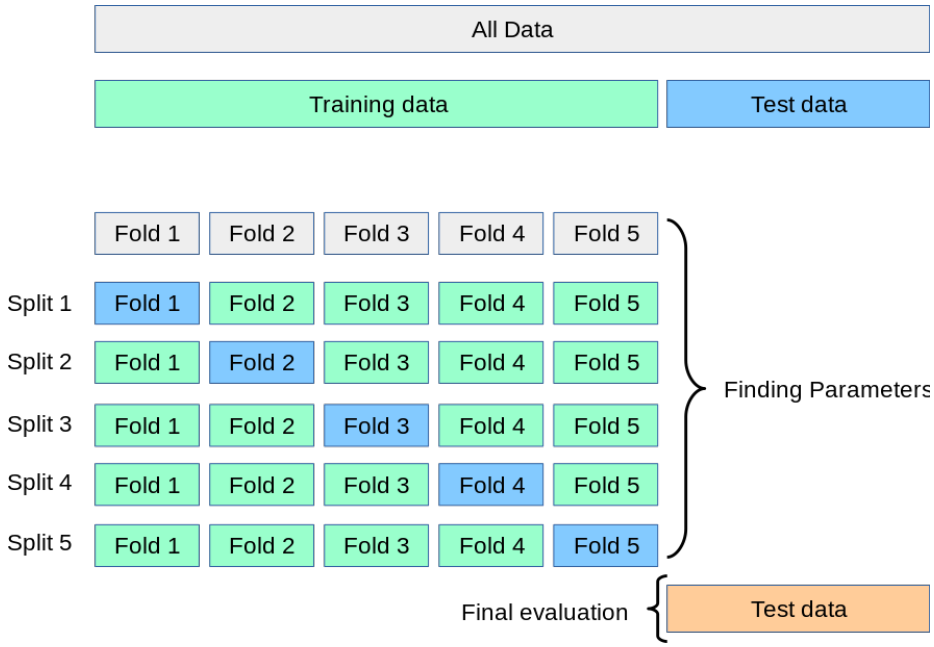


Figura 12: Ejemplo de una validación cruzada con K=5. En la imagen se puede ver el proceso descrito anteriormente[13].

[13]Figura obtenida del siguiente enlace: https://scikit-learn.org/stable/modules/cross_validation.html

En algunas ocasiones, un predictor puede tener un ratio de aciertos malo, lo que haría que no fuese recomendable predecir con él. Sin embargo, combinando varios se puede llegar a conseguir unos resultados bastante satisfactorios, lo que se conoce como metapredictores. Hay dos ideas interesantes relacionadas con esto último:

- *Bagging*: el nombre viene de *Bootstrap aggregating* y tiene un funcionamiento muy sencillo [10]. Se entrenan $k$ predictores en muestras de un mismo tamaño, y la clase retornada será la que prediga la mayoría de ellos.
- *Boosting*: se encarga de transformar predictores que tengan un error grande en uno robusto que obtenga buenas predicciones [11]. Para ello, entre otras cosas, se asignan pesos a los predictores y a los datos en función de las predicciones realizadas, dándole más peso a los datos con los que se obtengan peores resultados de forma que los clasificadores débiles se centren en ellos.

Lo que se usará en este caso es un tipo de *bagging* conocido como *Random Forest*, siendo un algoritmo que relaciona los árboles de decisión con un método tan intuitivo y que retorna tan buenos resultados como el explicado anteriormente. Un clasificador de *Random Forest* es una combinación de árboles de decisión independientes que trabajan en conjunto [12], de forma que, a la hora de clasificar, cada árbol indica la clase que predice, y aquella que se obtenga el mayor número de veces será la predicción del modelo final. Este proceso se puede ver ilustrado en la Figura 13.

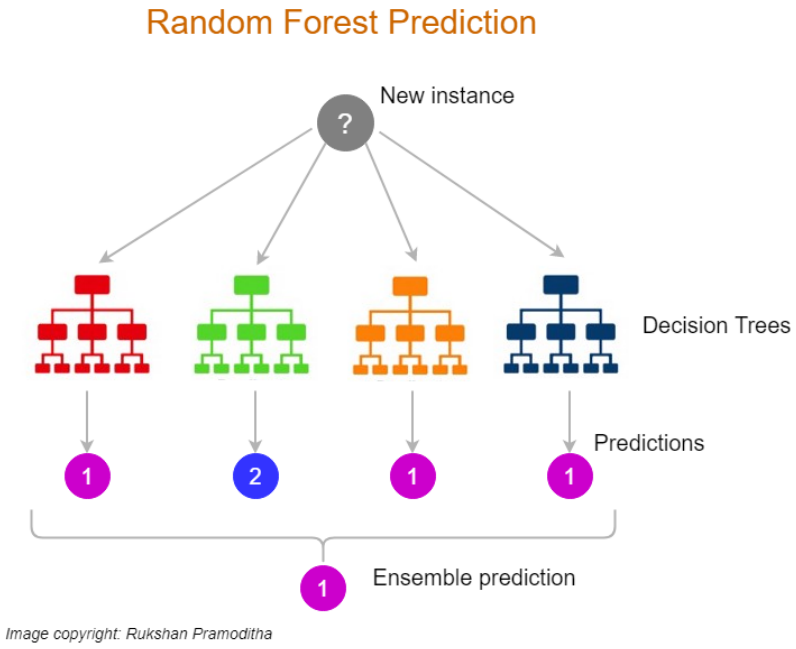


Figura 13: Imagen que ilustra el funcionamiento de un clasificador de *Random Forest*.

De esta forma se mejora el rendimiento y se impide el sobreajuste que puede afectar a veces a cada árbol por separado, siendo los *Random Forest* uno de los modelos más

efectivos que hay, dentro de la clase de meta-predictores. El problema de esto, respecto a los árboles de decisión, es que se pierde la posibilidad de poder visualizar el modelo predictivo. A la hora de aplicar el *Random Forest*, hay dos criterios que se pueden elegir para sus árboles, siendo los dos mencionados anteriormente: *gini* y *entropy*.

## 5.3. Clustering

Por último, tanto para generar los rangos de valores utilizados en las reglas de asociación como para obtener los perfiles de conducción respecto a los atributos, se han utilizado técnicas de *clustering*, analizando los resultados obtenidos para ver qué grupos se han generado y obteniendo conclusiones a partir de ello. El algoritmo empleado ha sido *K-means*.

*K-means* es un algoritmo iterativo de *clustering* que permite dividir una serie de datos en $K$ grupos o *clusters*, siendo $K$ un número entero previamente indicado, y teniendo en cuenta que un elemento no puede estar en más de un *cluster* a la vez. La división se realiza en base a los centroides de un *cluster*, definidos como el punto equidistante o media entre los datos asignados a dicho grupo, y utilizados como representación. La diferencia entre el centroide de un cluster $C_i$ representado como $c_i$ y un objeto $p$ tal que $p \in C_i$ se define como $dist(p, c_i)$, donde $dist(x, y)$ es la distancia Euclídea entre $x$ e $y$. La calidad de cada *cluster* viene definida por la fórmula del *within-cluster variation*, calculada como la suma de las distancias al cuadrado entre todos los elementos pertenecientes a $C_i$ y el centroide $c_i$. Esta fórmula intentará alejar los distintos *clusters* lo máximo posible. Está definida en la Ecuación (7) (véase [13]).

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2 \tag{7}$$

Como podemos ver, para cada elemento perteneciente al *cluster* se calcula su distancia Euclídea respecto al centroide, se eleva al cuadrado y se añade al sumatorio de distancias.

El algoritmo usado en el trabajo, procedente de la librería *sklearn*, utiliza la clásica implementación de Lloyd [14]. El funcionamiento del algoritmo es el siguiente: inicialmente se eligen $k$ elementos aleatorios de los datos totales como los centroides iniciales de los *clusters*. Después, se asignan los puntos a los grupos, basándose en la distancia Euclidea anteriormente mencionada. Un elemento pertenece a un *cluster* si la distancia entre ese valor y el centro de ese *cluster* es menor a la distancia con el resto de centroides. A continuación, se recalculan los centroides como la media de los valores que hay en cada grupo, y se realiza una nueva asignación de los valores a los distintos grupos con el mismo criterio, mejorando el *within-cluster variation* anteriormente mencionado. Estos dos últimos pasos se repetirán las veces necesarias hasta que el algoritmo converja, es decir, cuando las asignaciones no varíen de una iteración a otra, o hasta repetir un número fijado de veces. La posición inicial de los centroides es bastante decisiva en este algoritmo, por lo que se debe repetir varias veces con distintas inicializaciones.

En el caso del *clustering* aplicado a la obtención de rangos para las reglas de asociación, hace falta saber en cuántos *clusters* se deben dividir los datos. Para ello, se

ha utilizado "el método del codo" (*the elbow method*). Este método es una heurística que permite observar cuál es el número óptimo de divisiones, ya que se muestra en una gráfica el *score* por cada división, siendo ese *score* una métrica que indica cómo de coherentes son los *clusters* con los elementos que contienen. Un ejemplo de cómo se ha realizado se puede ver en la Figura 14.

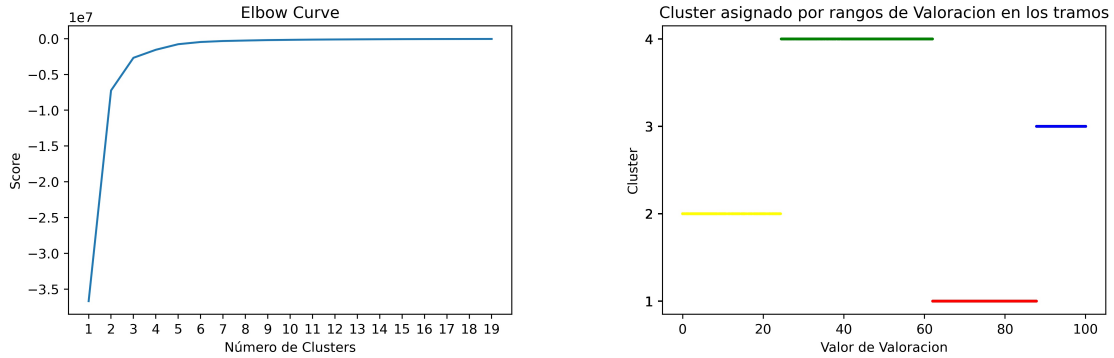


Figura 14: Ejemplo del proceso de la división de un atributo en *clusters*.

En el ejemplo anterior se muestra el caso del atributo Valoración. A través del método del codo, se puede apreciar en la gráfica de la izquierda que una división en 4 *clusters* puede ser bastante interesante debido al *score* que tiene ese valor y a la poca variación que hay a partir de dicho valor. A su vez, tras realizar la división, se puede observar en la gráfica de la derecha cómo se reparten los valores de los grupos, siendo en este caso adecuados para los datos y su distribución.

# 6. Resultados, validación y evaluación

Una vez aplicados los algoritmos, se han obtenido unos resultados que habrá que analizar para ver si son interesantes y útiles. Estos resultados deben permitir poder cumplir los objetivos.

## 6.1. Resultados obtenidos

### 6.1.1. Reglas de asociación

Se han aplicado las reglas tanto a los atributos originales como a los obtenidos calculando las medias por unidad, y a su vez a cada uno de ellos se le han minado reglas de asociación normales, con 0,4 de mínimo soporte y 0,8 de mínima confianza, y reglas esporádicas perfectas, con 0,18 de máximo soporte y 0,7 de mínima confianza, obteniendo por lo tanto cuatro tablas de reglas[14]. En el caso de las reglas con los atributos originales, se obtenían una muy alta cantidad de ellas, por lo que se han

[14]Las reglas han sido descargas en ficheros Excel y se pueden ver aquí: https://unicancloud-my.sharepoint.com/:f:/g/personal/jgm188_alumnos_unican_es/Ek7vVZJD1CNFiPDhywMbObMBWU-TUxhzVfyDNUtp-10u4Q?e=8YL3pA

eliminado aquellas en las que no salían el tiempo de uso de conducción de crucero, las aceleraciones y frenadas bruscas, la conducción no predictiva y el tiempo y consumo de ralentí, al ser considerados como los atributos que podían dar a lugar a conclusiones novedosas.

Las reglas se identificarán a través del nombre del atributo y su rango de valor, que puede ser "Alto", "Medio Alto", "Medio Bajo" y "Bajo" si está dividido en cuatro rangos o "Alto", "Medio" y "Bajo" si está dividido en tres. Para mejorar la visualización de los resultados, se usarán distintos símbolos, tal como se indica en la Tabla 32. Por otra parte, el soporte se indicará como "sup" y la confianza como "conf".

Tabla 32: Tabla con la relación entre los símbolos y el rango del valor del atributo.

| **Símbolo** | $\uparrow$ | $\nwarrow$ | $\leftarrow$ | $\swarrow$ | $\downarrow$ |
|---|---|---|---|---|---|
| **Significado** | Alto | Medio Alto | Medio | Medio Bajo | Bajo |

A través de las reglas, se puede confirmar conocimiento imaginable o esperable. Por ejemplo, en las reglas normales con los atributos por defecto se pueden apreciar reglas como las mostradas en la Tabla 33.

Tabla 33: Reglas con los atributos por defecto que muestran un conocimiento que se puede considerar como obvio.

$$\begin{array}{lcl} \text{AceleracionesBruscas}(\downarrow) & \Longrightarrow & \text{Valoración}(\uparrow) \\ \text{FrenadasBruscas3}(\downarrow),\ \text{CNoPredictiva3}(\downarrow) & \Longrightarrow & \text{Valoración}(\uparrow) \end{array}$$

A través de estas dos reglas, se pueden observar relaciones que uno se podría esperar. Por ejemplo, si las aceleraciones bruscas son bajas, es obvio que eso repercutirá en una valoración alta, ya que se estarán siguiendo los estándares de conducción recomendados. De igual manera, si las frenadas bruscas y el número de conducciones no predictivas son bajas, esto también ayudará a una buena valoración. Con los datos independientes, también se pueden obtener reglas de este estilo, mostradas en la Tabla 34.

Tabla 34: Reglas con los atributos independientes que muestran un conocimiento que se puede considerar como obvio.

$$\begin{array}{lcl} \text{RalInnec/s}(\downarrow) & \Longrightarrow & \text{Valoración}(\uparrow) \\ \text{VelocidadMedia}(\swarrow) & \Longrightarrow & \text{Valoración}(\uparrow) \end{array}$$

De igual manera, aquí también se aprecian reglas obvias, como que los ralentíes innecesarios bajos o una velocidad moderada, sin ser muy baja, implican una buena valoración. Sin embargo, hay varias reglas que aportan información nueva muy interesante, pasando a ser analizadas a partir de ahora. El primer grupo de reglas interesantes es el mostrado en la Tabla 35. Cada regla está acompañada de sus parámetros de soporte y confianza.

Tabla 35: Conjunto de reglas interesantes sobre los datos originales.

| Reglas | | | | Sup | Conf |
|---|---|---|---|---|---|
| Aceleraciones Bruscas(↓), | CNoPredictiva(↓) | $\Longrightarrow$ | Frenadas Bruscas(↓) | 0,4327 | 0,9047 |
| Frenadas Bruscas(↓), | CNoPredictiva(↓) | $\Longrightarrow$ | Aceleraciones Bruscas(↓) | 0,4327 | 0,9376 |
| Frenadas Bruscas(↑), | CNoPredictiva(↖) | $\Longrightarrow$ | Aceleraciones Bruscas(↑) | 0,02724 | 0,8166 |

A través de las primeras dos reglas, se puede observar una relación entre aceleraciones bruscas bajas y frenadas bruscas bajas cuando están combinadas con una baja conducción no predictiva. Son reglas con una alta confianza, por lo que son muy confiables, y también son bastante interesantes, ya que si bien es verdad que la conducción no predictiva está estrechamente relacionada con las aceleraciones y frenadas bruscas por definición, estas dos últimas también parecen estar relacionadas entre sí. Finalmente, la última regla del conjunto es una regla rara que demuestra que las aceleraciones bruscas y las frenadas bruscas también van de la mano cuando son altas. Una de las reglas de los datos independientes está relacionada con estas, mostrada en la Tabla 36.

Tabla 36: Regla interesante sobre los datos independientes, que relaciona directamente las aceleraciones bruscas con las frenadas bruscas.

| Reglas | | | Sup | Conf | Lift |
|---|---|---|---|---|---|
| Aceleraciones Bruscas/s(↓) | $\Longrightarrow$ | Frenadas Bruscas/km(↓) | 0,3609 | 0,8673 | 1,3896 |

Por lo que a través de esta regla (cuya confianza y *lift* es también bastante alto) se puede confirmar que las aceleraciones bruscas bajas y las frenadas bruscas bajas van de la mano. Con todas estas reglas se puede concluir que por lo general ambos atributos tienden a estar en un grupo similar de distribución de valores, si las aceleraciones son altas, las frenadas serán también altas y viceversa. Igualmente, si una de ellas es baja, la otra también lo será.

Por otra parte, hay varias reglas acerca de un atributo que son sin duda las más interesantes de todas. Respecto a los atributos normales, la regla con ese atributo interesante (que será el tiempo de uso de conducción de crucero) está en la Tabla 37.

Tabla 37: Regla interesante con el atributo del tiempo de conducción de crucero.

| Reglas | | | | Sup | Conf | Lift |
|---|---|---|---|---|---|---|
| Tiempo Ralentí(↓), | Tiempo Conducción Crucero(↓) | $\Longrightarrow$ | Valoración(↑) | 0,4006 | 0,9162 | 1,1059 |

En esta regla se relacionan dos valores bajos de atributos que dan lugar a una valoración alta: el tiempo ralentí, lo cual es esperable ya que un alto tiempo de ralentí

puede no ser recomendable para el motor, y el tiempo de uso de conducción de crucero, el cual es muy sorprendente. Cabe destacar la alta confianza que tiene esta regla. En principio, la conducción de crucero es algo que se considera eficiente con el consumo de combustible, por lo que se recomienda usarlo. Sin embargo, los tramos con un bajo tiempo de uso de esta característica obtienen una mejor valoración. En el caso de los datos independientes, existen dos reglas relacionadas con lo mencionado, estando representadas en la Tabla 38.

Tabla 38: Reglas interesantes de los datos independientes relacionadas con el tiempo de conducción de crucero y de ralentí.

| **Reglas** | **Sup** | **Conf** | **Lift** |
|---|---|---|---|
| Tiempo Ralentí/s(↓) $\Longrightarrow$ Valoración(↑) | 0,4775 | 0,8522 | 1,0287 |
| Tiempo Conducción(↓) Crucero/s $\Longrightarrow$ Valoración(↑) | 0,6237 | 0,8730 | 1,0538 |

En estas reglas se relacionan directamente el tiempo de ralentí bajo y el tiempo de uso de conducción de crucero bajo con una valoración alta, confirmando lo dicho anteriormente. Cabe destacar el alto soporte que tiene la segunda regla, que unido a la alta confianza y a un buen *lift* hacen ver que está claro que, dentro de los datos de los que se disponen, un bajo uso de la conducción de crucero está muy relacionado con una buena valoración en el tramo, obteniendo una sorprendente conclusión.

### 6.1.2. Árboles de decisión

Para este objetivo se han generado el árbol de decisión y el *Random Forest* más precisos para los datos proporcionados a través de búsqueda de parámetros con la validación cruzada, consiguiendo un buen modelo predictivo. Empezando con los árboles de decisión, se han generado varios árboles para ver si se puede predecir con exactitud si un tramo es bueno o no. A la hora de indicar si un tramo es bueno o no, se debe especificar un valor límite a partir del cual se pueda marcar esa diferencia. Hay una gran cantidad de tramos con muy buena puntuación, por lo que realmente no sería interesante poder obtener un predictor para saber si un tramo tendrá más de 50 de valoración o no, cuando se sabe que la gran mayoría lo va a tener. Se ha empezado por lo tanto con un valor de 90, de forma que más de 90 es considerado un tramo bueno o interesante, y menos de 90 un tramo menos interesante, al ser por debajo de la mediana.

Inicialmente, se ha intentado obtener el árbol de decisión con los atributos por defecto. El campo Valoracion ha pasado a ser un campo discreto con solo dos posibles valores y se ha calculado inicialmente un árbol de profundidad 3 para ver su aspecto, mostrado en la Figura 15. Por otra parte, se ha usado en todos los modelos el 25 % de los datos como validación y el 75 % restante como datos de entrenamiento, al ser el valor por defecto de *sklearn*. A su vez, a la hora de buscar los mejores parámetros, se usará *cross-validation* con 10 divisiones.

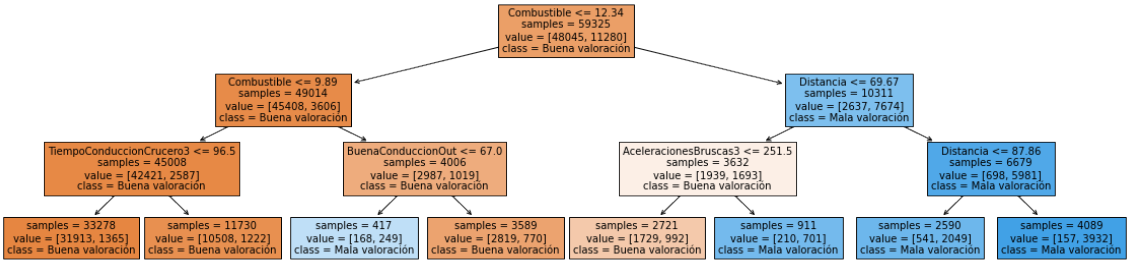


Figura 15: Árbol de profundidad 3 obtenido con los atributos por defecto.

En el árbol podemos apreciar los nodos de decisión y los nodos finales mencionados en la anterior sección, mostrándose en cada uno cuál sería la clase predicha por el modelo si se siguiese ese recorrido de decisiones. Los colores indican la clase a la que pertenece el nodo, siendo el naranja la clase de buena valoración y el azul la de mala valoración. La tonalidad del color indica la predominancia de la clase predicha frente a la otra clase, de forma que un naranja muy intenso indicará una muy alta probabilidad de que la clase final sea la de buena valoración.

Como se puede ver en el árbol, hay dos atributos que parecen tener una importancia muy alta: uno de ellos es Combustible, que con valores pequeños determina de manera casi clara que el tramo será bueno, y un valor muy alto hará que el tramo sea muy propenso a una mala valoración. Por otra parte, el campo Distancia parece también decisivo, dado que un valor muy alto condena al tramo a una mala valoración definitiva. La importancia de los atributos se puede ver en la Tabla 39, donde los que tienen un valor nulo no se han incluido.

Tabla 39: Importancia de los atributos en el árbol de la Figura 15.

| **Atributo** | **Importancia** |
|---:|---:|
| Combustible | 0,853728 |
| Distancia | 0,103060 |
| AceleracionesBruscas3 | 0,024036 |
| BuenaConduccionOut | 0,011745 |
| TiempoConduccionCrucero3 | 0,007431 |

Aquí se puede apreciar la crucial importancia del atributo Combustible, con un porcentaje altísimo, de más del 85 %. El atributo Distancia es también determinante, con poco más de un 10 %. Sorprende el hecho de que atributos como las frenadas bruscas no estén, ya que hemos visto en las reglas de asociación que estaba relacionado con la valoración. Por otra parte, a partir del *GridSearchCV* definido anteriormente, se ha pasado a realizar una búsqueda del mejor árbol para este caso, obteniendo uno con profundidad 11 y 106 nodos terminales distintos. La matriz de confusión obtenida con este árbol clasificando al conjunto de test es la mostrada en la Tabla 40.

Tabla 40: Matriz de confusión obtenida tras clasificar el conjunto de test.

| | | | |
|---|---|---|---|
| **Clase verdadera** | Buena valoración | 15592 | 369 |
| | Mala valoración | 1291 | 2523 |
| | | Buena valoración | Mala valoración |
| | | **Predicción de clase** | |

Como se puede apreciar, predice muy bien los buenos tramos, y algo peor los malos tramos. Los valores de precisión con el conjunto de validación son los mostrados en la Tabla 41.

Tabla 41: Valores de precisión obtenidos tras clasificar el conjunto de test.

| **Precisión** | **TPRate** | **TNRate** |
|---|---|---|
| 91,6056 % | 0,97688 | 0,66151 |

Siendo *TPRate* el porcentaje de buenos tramos clasificados correctamente y *TNRate* el porcentaje de malos tramos clasificados correctamente. Se puede ver que la precisión en la clase de buena valoración es muy alta, mientras que la de la de mala valoración, pese a tener un valor superior al 50 %, es peor. Por otra parte, el hecho de que uno de los atributos más determinantes sea la distancia, siendo un atributo que no depende del desempeño del conductor, hace que este árbol no sea interesante. Para evitar esto último, se ha pasado a usar los atributos independientes, los cuales se sabe que son todos representativos del desempeño que ha tenido el conductor a la hora de manejar el vehículo.

Por otra parte, el hecho de que la clase de mala valoración haya tenido una precisión bastante más baja a la de buena valoración ha podido darse porque la cantidad de filas de la primera clase sea bastante inferior a la otra. Por lo tanto, además de usar los datos independientes, se han añadido filas repetidas de la clase de mala valoración dentro del conjunto de *training* para equiparar la cantidad. Se ha repetido también el mismo proceso con estos datos. Al igual que antes, se ha creado primero un árbol con altura 3, para ver qué aspecto tiene, mostrado en la Figura 16.

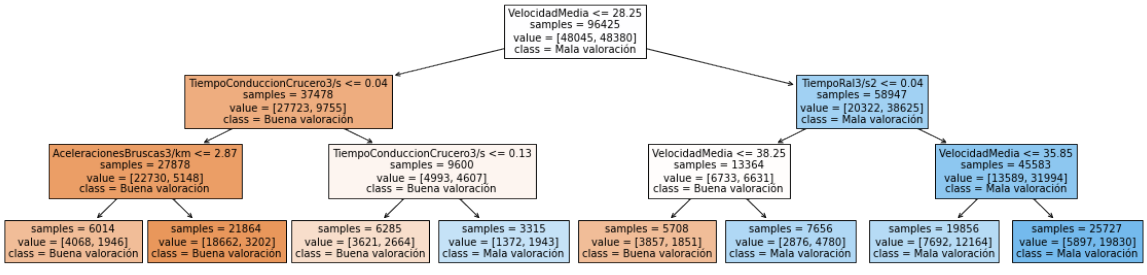


Figura 16: Árbol de profundidad 3 obtenido con los atributos independientes.

En este caso, se puede ver que el nuevo atributo más importante pasa a ser el de la velocidad media, pudiendo observar que una velocidad muy alta va a implicar que

un tramo tenga una valoración por debajo de la mediana. Por otra parte, el tiempo de uso de conducción de crucero también parece bastante importante. De hecho, ya en lo obtenido en las reglas de asociación se hablaba de que un alto tiempo de uso de conducción de crucero implicaba una mala valoración, y aquí se puede ver que eso es así, dado que los altos valores de este atributo llevan a obtener tramos de este tipo. Utilizando el *GridSearchCV*, se ha obtenido un árbol muy grande (47 de profundidad y 8273 hojas) con una precisión en el conjunto de entrenamiento altísima, lo que indica que es muy probable que el árbol esté sobreajustado. Esto se puede ver con la matriz de confusión de la Tabla 42, obtenida tras hacer que clasifique un conjunto de datos que no ha visto.

Tabla 42: Matriz de confusión obtenida tras clasificar el conjunto de test.

| | | | |
|---|---|---|---|
| **Clase verdadera** | Buena valoración | 13677 | 2284 |
| | Mala valoración | 2317 | 1497 |
| | | Buena valoración | Mala valoración |
| | | **Predicción de clase** | |

Se puede apreciar que clasifica bastante bien los tramos buenos, pero se equivoca mucho con los tramos malos. Esto se puede confirmar con las medidas de precisión representadas en la Tabla 43.

Tabla 43: Valores de precisión obtenidos tras clasificar el conjunto de test.

| **Precisión** | **TPRate** | **TNRate** |
|---|---|---|
| 76,7332 % | 0,8569 | 0,3925 |

Una precisión de 0,3925 para la clase negativa es malísima, por lo que el árbol muy probablemente se ha sobreajustado al conjunto de entrenamiento, y claramente no tiene utilidad. Una posible solución a esto último es usar *Random Forest*, tal y como se explicó en el apartado anterior, ya que es muy difícil que se sobreajusten. Sin embargo, en este caso se pierde la posibilidad de visualizar el árbol.

Dentro de *Random Forest*, hay varias características a tener en cuenta. El primero de ellos es el número de estimadores o árboles que tenga el modelo. Hay un artículo que indica que la cantidad recomendable es entre 64 y 128 árboles [15], pero se ha probado también con otros valores como 32, 256 y 512. Por otra parte, se han realizado ejecuciones con dos criterios, *gini* y *entropy*, se ha valorado si incluir una profundidad máxima de los árboles o no, y por último, se ha indicado el máximo número de características. Toda esta búsqueda ha sido realizada con la clase *GridSearchCV*, obteniendo como mejores parámetros el criterio de *gini*, sin límite de profundidad, un máximo de características de 5 y 128 estimadores como los mejores atributos. Con esto, se obtiene la matriz de confusión en el conjunto de validación indicada en la Tabla 44.

Tabla 44: Matriz de confusión obtenida tras clasificar el conjunto de test.

| | | | |
|---|---|---|---|
| **Clase verdadera** | Buena valoración | 14673 | 1288 |
| | Mala valoración | 2067 | 1747 |
| | | Buena valoración | Mala valoración |
| | | **Predicción de clase** | |

Esta matriz mejora a la anterior, pero sigue siendo bastante mala para la clase de malos tramos. Los valores de precisión en el conjunto de test son los obtenidos en la Tabla 45.

Tabla 45: Valores de precisión obtenidos tras clasificar el conjunto de test.

| **Precisión** | **TPRate** | **TNRate** |
|---|---|---|
| 83,0341 % | 0,9193 | 0,458 |

La precisión de la clase de mala valoración mejora, pero no lo suficiente. Por lo tanto, se deduce que es muy difícil diferenciar entre aquellos tramos con menos de 90 de valoración y aquellos con más de 90 utilizando unos atributos verdaderamente representativos.

A partir de aquí, surge la idea de replantear el propósito de este apartado. Teniendo en cuenta que la mayoría de los tramos tienen una valoración superior a 90 y que cerca de un 30 % tiene una valoración perfecta, es realmente más interesante saber si un tramo es perfecto o no, y ver qué atributos son más importantes para llegar a obtener un tramo de esa magnitud. Por lo tanto, se ha repetido el mismo proceso de antes, pero diferenciando entre dos clases dentro del atributo de valoración: los perfectos, siendo aquellos que tengan un valor superior a 99, y los no perfectos en caso contrario, considerando que, por ejemplo, un tramo de 99,4 es un tramo perfecto. El árbol con profundidad inicial de 3 tiene un aspecto similar al anterior, por lo que no es necesario mostrarlo.

Se ha pasado directamente a calcular el árbol más preciso, el cual tiene una profundidad de 15 y 167 nodos finales, que dificultan el poder visualizarlo en este informe[15]. Los atributos más importantes en este caso son los mostrados en la Tabla 46.

[15]El árbol completo se puede ver pulsando aquí. Dado el gran tamaño del árbol, es necesario hacer zoom y/o descargar la imagen para poder observar los nodos correctamente. También se puede descargar a través de este enlace.

Tabla 46: Atributos más importantes para el clasificador de tramos perfectos con los datos independientes.

| **Atributo** | **Importancia** |
|---:|---:|
| VelocidadMedia | 0,622308 |
| TiempoRal3/s2 | 0,084975 |
| Consumo100 | 0,076186 |
| TiempoConduccionCrucero3/s | 0,062516 |
| FrenadasBruscas3/km | 0,053304 |
| AceleracionesBruscas3/km | 0,046060 |
| CNoPredictiva3/km | 0,037154 |
| ConsumoRal3/s2 | 0,011344 |
| RalInnec3/s2 | 0,006153 |
| Subida | 0,000000 |

Se confirma que, al igual que antes, la velocidad vuelve a tener un papel crucial a la hora de decidir los tramos. Por otra parte, el resto de atributos están más o menos equilibrados, destacando el hecho de que el atributo Subida no afecta en nada, y que el tiempo de uso de conducción de crucero vuelve a ser importante. La matriz de confusión obtenida con el conjunto de validación es la mostrada en la Tabla 47.

Tabla 47: Matriz de confusión obtenida tras clasificar el conjunto de test.

| | | | |
|---|---|---|---|
| **Clase verdadera** | Buena valoración | 7998 | 2380 |
| | Mala valoración | 3372 | 6025 |
| | | Buena valoración | Mala valoración |
| | | **Predicción de clase** | |

En este caso, se consiguen unas predicciones decentes en la clase de la mala valoración, pese a que la otra clase ha perdido algo de precisión. Esto se puede ver más en detalle en la Tabla 48.

Tabla 48: Valores de precisión obtenidos tras clasificar el conjunto de test.

| **Precisión** | **TPRate** | **TNRate** |
|---|---|---|
| 70,9128 % | 0,77067 | 0,6412 |

A través de estas métricas, se puede apreciar que se ha conseguido obtener un árbol de decisión interesante, con atributos representativos y que además predice bastante bien ambas clases. Se ha intentado, como última opción, usar *Random Forest* para ver si se obtiene un modelo que mejore la precisión anterior. Se puede ver que la mejoría obtenida con el mejor *Random Forest*, siendo uno con criterio de *entropy*, una profundidad máxima de 20, 128 estimadores y un número máximo de 5 características, es la indicada en las Tablas 49 y 50.

Tabla 49: Matriz de confusión obtenida tras clasificar el conjunto de test.

| | | | |
|---|---|---|---|
| **Clase verdadera** | Buena valoración | 8154 | 2224 |
| | Mala valoración | 3099 | 6298 |
| | | Buena valoración | Mala valoración |
| | | **Predicción de clase** | |

Tabla 50: Valores de precisión obtenidos tras clasificar el conjunto de test.

| **Precisión** | **TPRate** | **TNRate** |
|---|---|---|
| 73,0822 % | 0,7857 | 0,6702 |

Como se puede apreciar, la mejoría no es muy grande, pero ahora predice con más exactitud ambas clases. Por lo tanto, gracias a este *Random Forest*, se ha obtenido un modelo predictivo capaz de clasificar los tramos en perfectos o imperfectos con bastante éxito, cumpliendo, junto con el árbol anterior, el objetivo marcado. Además, se ha podido apreciar la importancia de algunos atributos como los tiempos de uso de conducción de crucero, el tiempo de ralentí y, sobre todo, la velocidad media, a la hora de clasificar tramos.

### 6.1.3. Clustering

Por último, está el objetivo de agrupación de tramos por perfiles de conducción, donde se ha utilizado *clustering*. Se ha trabajado directamente sobre los datos independientes, ya que no sería tan interesante obtener distintas divisiones donde los atributos determinantes pudiesen ser la distancia recorrida o el tiempo total. Además, se han normalizado los datos antes de realizar las divisiones, para asegurarse de que están en la misma escala. Esto es muy importante para las distancias Euclídeas de *k-means*, ya que de no ser así, aquellos atributos con un rango más amplio de valores dominarían la división en *clusters*, al obtener valores de distancia más altos que con otros atributos [16]. Posteriormente, se han usado los valores iniciales para visualizar los *clusters*.

Inicialmente, se ha realizado una división en dos grupos, analizando este primer resultado. Esta división no es muy satisfactoria, dado que los grupos están muy descompensados, teniendo un grupo 75223 filas y el otro 3296. La división además es muy similar a una división donde se considerasen tramos con una valoración suspensa, es decir, por debajo de 50, y tramos aprobados, con atributos muy buenos. Dado que la división no es interesante y no aporta información nueva, se ha pasado directamente a utilizar 3 *clusters*.

En este caso, sí que se obtiene una división más sorprendente. El grupo con tramos suspensos se mantiene intacto, pero el primer grupo pasa a dividirse en dos bastante interesantes. El primer grupo de los aprobados tiene una media más baja de valoración que el otro grupo aprobado (83,18 frente a 97,92), pero tiene una media muy alta de velocidad media, más que incluso el grupo de valoración mala, y un consumo similar al de otros grupos. Sin embargo, posee atributos bastante bajos en campos como

aceleraciones y frenadas bruscas o el tiempo de conducción de crucero. Por otra parte, el otro grupo de los aprobados tiene estadísticas a la inversa.

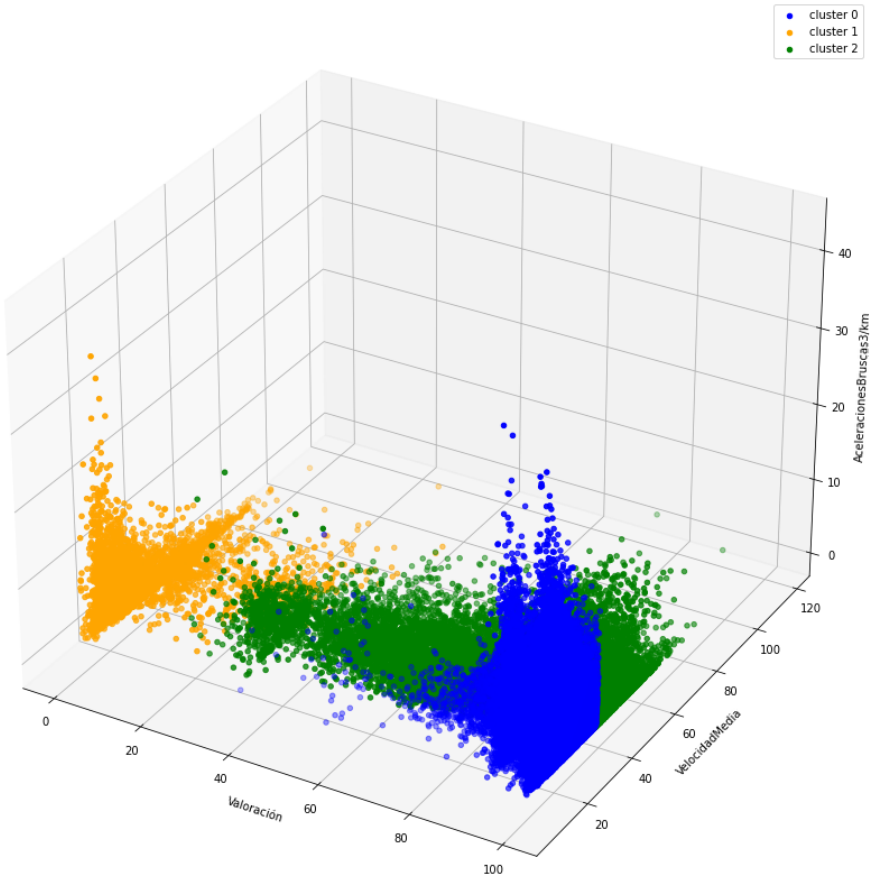


Figura 17: *Clusters* representados en base a su valoración, velocidad media y aceleraciones bruscas.

En la Figura 17 se puede apreciar lo indicado, siendo el *cluster 1* el grupo de tramos suspensos, el *cluster 0* el grupo de tramos aprobados con menor velocidad pero más movimientos bruscos y el *cluster 2* el otro grupo de aprobados con mayor velocidad pero un menor valor de atributos relacionados con la conducción no predictiva. Por lo tanto, hay un grupo de perfiles de conductores que tienen una conducción más "experta", van a velocidad más alta, pero cometen menos errores, ya que los valores de aceleraciones bruscas, frenadas bruscas o conducción no predictiva son bastante bajos en comparación con el resto. Lo he llamado de esa forma ya que, por lo general, cuando uno utiliza durante bastante tiempo un vehículo y coge confianza, tiende a realizar menos errores a la vez que se aumentan las velocidades. Por otra parte, el otro grupo de los aprobados parece un grupo con una conducción más "novata", donde conducen con menor velocidad y además son más propensos a tener movimientos bruscos. Cabe destacar que la valoración es más alta en el *cluster 2* que en el 1, lo que parece indicar que la velocidad media es más determinante que los movimientos bruscos a la hora de indicar las métricas de conducción, lo cual está relacionado con lo deducido en el análisis dedicado al segundo objetivo.

Si se pasa a realizar una división en un número mayor de grupos, se puede ver que esto se acentúa. Con cuatro *clusters*, los grupos pasan a ser:

- Un grupo con valoraciones muy malas, al igual que antes.
- Un grupo con valoraciones muy buenas y velocidad alta, pero con menos movimientos bruscos.
- Un grupo con valoraciones muy buenas y velocidad baja, pero con bastantes movimientos bruscos.
- Un nuevo grupo que parece representar unos tramos con una valoración de media 64, que sería aceptable, y con una velocidad muy alta, pero con el resto de parámetros muy similar al *cluster* 2, por lo que realmente estos dos están relacionados.

Por lo tanto, se puede ver que si se sigue dividiendo en más *clusters* no se conseguirá realmente ninguna información extra de la obtenida hasta ahora, ya que en el caso anterior el último grupo está bastante relacionado con el de la conducción experta. Gracias a la aplicación de *clustering*, se pueden obtener tres perfiles principales de tramos: aquellos mal realizados, que tienen una valoración baja, aquellos que han sido conducidos de forma experta, ya que se aprecian velocidades altas pero pocos errores, recordando a lo que podría ser la conducción de alguien que lleva trabajando bastante tiempo y está muy acostumbrado a la máquina, y un último perfil de conducción más novata, siendo cauto con las velocidades pero cometiendo bastantes errores.

## 6.2. Validación y conclusiones

Una vez obtenidos todos los resultados, se ha mantenido una reunión con el equipo de Fagor SDS para mostrárselos y de esta forma verificar si son interesantes. La retroalimentación enviada por la empresa tras enseñarles el funcionamiento del trabajo y los resultados obtenidos es la siguiente: "El trabajo realizado por Jorge nos ha permitido realizar el acercamiento y primeros pasos a las últimas tecnologías de minería de datos con las que mejorar los resultados actuales obtenidos con el procesamiento de datos tradicional. El análisis realizado sobre el procedimiento de cálculo y extracción de los tramos nos ha permitido corregir errores, como por ejemplo el efecto de la conducción crucero, y aumentar la precisión de los mismos, obteniendo así mejoras en la evaluación de los conductores. Estamos enormemente satisfechos con los resultados y gracias a este proyecto se ha incluido en el plan de gestión la incorporación de un ingeniero de datos encargado de liderar la incorporación de las tecnologías de ciencias de datos en nuestros procesos así como de continuar la investigación."

A partir de este trabajo se pueden obtener varias conclusiones. Respecto a los objetivos, lo más interesante está en las reglas de asociación, donde se mostraba la relación entre la cantidad de aceleraciones bruscas y la de frenadas, y sobre todo, el sorprendente hecho de que un alto uso de la conducción de crucero afecta negativamente a la valoración de tramo, de forma que parece que esto realmente no es eficiente. Los modelos predictivos han mostrado la importancia de otros atributos, como la velocidad

media o el propio tiempo de uso de conducción de crucero, bastante determinantes a la hora de decidir la clase de un tramo, además de la posibilidad de predecir con bastante precisión si un tramo será perfecto o no a través de un modelo. Por último, la aplicación de *clustering* ha permitido descubrir dos perfiles interesantes de conducción dentro de la base de datos, uno que representa a una conducción más veterana y otro que representa a un estilo más novel, ambos con buenas valoraciones.

Respecto a la metodología del proyecto, se ha podido apreciar el potencial de las técnicas de Ciencia de Datos, haciendo posible obtener conclusiones muy interesantes, y en algunas ocasiones inesperadas, consiguiendo información que de otra forma no se podría obtener.

# 7. Bibliografía


[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825-2830, 2011. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[2] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Zhihong Deng, and Hoang Thanh Lam. The SPMF Open-Source Data Mining Library Version 2. In *Machine Learning and Knowledge Discovery in Databases*, pages 36-40, Cham, 2016. Springer International Publishing. https://doi.org/10.1007/978-3-319-46131-1_8.

[3] Jeroen Baijens, Remko Helms, and Rob Kusters. Data analytics project methodologies: Which one to choose? In *Proceedings of the 2020 International Conference on Big Data in Management*, ICBDM 2020, page 41–47, New York, NY, USA, 2020. Association for Computing Machinery. https://dl.acm.org/doi/10.1145/3437075.3437087.

[4] Peter Chapman, Janet Clinton, Randy Kerber, Tom Khabaza, Thomas P. Reinartz, Colin Shearer, and Richard Wirth. Crisp-dm 1.0: Step-by-step data mining guide. 2000.

[5] Gavin Harper and Stephen D. Pickett. Methods for mining hts data. *Drug Discovery Today*, 11(15):694–699, 2006. https://doi.org/10.1016/j.drudis.2006.06.006.

[6] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487-499. Citeseer, 1994. https://rakesh.agrawal-family.com/papers/vldb94apriori.pdf.

[7] Yun Sing Koh and Nathan Rountree. Finding sporadic rules using apriori-inverse. In Tu Bao Ho, David Cheung, and Huan Liu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 97–106, Berlin, Heidelberg, 2005. Springer

Berlin Heidelberg.
https://doi.org/10.1007/11430919_13.

[8] L. Breiman. *Classification and Regression Trees*. (The Wadsworth statistics / probability series). Wadsworth International Group, 1984.

[9] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. Cross-validation. In *The elements of statistical learning: data mining, inference, and prediction*, volume 2, pages 241-249. Springer, 2009.
https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12_toc.pdf.

[10] L. Breiman. Bagging predictors. *Mach Learn*, (24):123140, 1996.
https://doi.org/10.1007/BF00058655.

[11] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.
http://www.yorku.ca/gisweb/eats4400/boost.pdf.

[12] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.
https://doi.org/10.1023/A:1010933404324.

[13] Micheline K. Han, J. and Jian P. *Data Mining. Concepts and Techniques, 3rd Edition*. Elsevier, 2012.

[14] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
https://doi.org/10.1109/TIT.1982.1056489.

[15] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest? In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 154-168, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
https://doi.org/10.1007/978-3-642-31537-4_13.

[16] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551-556, 2004.