Universidad de Cantabria

Facultad de Ciencias

Cálculo de ruta óptima y ETAs en transporte multimodal

Optimal route calculation and ETAs in multimodal transport

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Alberto Moro Carrera

Director: Diego García Saiz

Co-Director: Héctor Fernández Román

06 - 2023

Resumen

En una compañía logística que gestiona rutas internacionales, un sistema de cálculo de rutas y estimación de tiempo de llegada es crucial para realizar una primera línea base sobre el recorrido que se realizará y cuánto tiempo podría llevar realizar dicho recorrido, de cara a que los organismos pertinentes puedan organizarse para el tratamiento y operación de los diferentes productos transportados.

En estas rutas existe un concepto crucial, la multimodalidad, que consiste en la realización de estos recorridos alternando el medio de transporte, y teniendo en cuenta el tiempo que se tarda en transferir la mercancía de uno de los medios de transporte a otro.

Este TFG trata de resolver este problema para una empresa del parque tecnológico de Cantabria llamada Fieldeas perteneciente al grupo CIC, teniendo en cuenta el medio de transporte marítimo y el ferroviario, y centrándose en rutas internacionales, donde el barco tiende a ser el medio predilecto.

El uso del sistema se realizará a partir de un conjunto de coordenadas de inicio y final, generará una ruta la cual será utilizada como primera aproximación, pues los puertos en los que pueda parar, con los que la empresa tenga acuerdos y demás temas administrativos, son ajenos al sistema, por lo que finalmente ésta variará, por este motivo, se busca priorizar una respuesta rápida a una respuesta con muchos *checkpoints*, ya que se utilizará como directriz general para conocer si se bordea un continente en barco o se atraviesa en tren y por tanto, constituir una estimación del tiempo global, en lugar de una aplicación de guía en tiempo real. El sistema también valorará el tiempo que se tarda en realizar el transbordo de mercancías entre los dos medios, teniendo en cuenta el peso transportado, además del clima y la fecha en tiempo real, para realizar la ruta óptima.

Para implementarlo, se ha decidido utilizar varias tecnologías, por ejemplo, Python como lenguaje de programación, por su mayor soporte de librerías de *Machine Lerning*, Power Bi para la visualización de las rutas o CSV para el almacenamiento de datos por su integración directa con ambas tecnologías.

Palabras clave: Ciencia de datos, Rutas, Multimodalidad, Transporte, Algoritmos Predictivos.

Abstract

In a logistics company that manages international routes, a route calculation and estimated time of arrival system is crucial to establish an initial baseline for the journey and how long it may take to complete it, allowing relevant organizations to organize themselves for the treatment and operation of the different transported products.

In these routes, there is a crucial concept called "multimodality," which involves alternating the means of transportation during the journey and considering the time it takes to transfer the cargo from one mode of transport to another.

This final project aims to solve this problem for a company in the Cantabria Technology Park (Fieldeas), which belongs to the CIC group. The focus is on maritime and railway transport, with a specific emphasis on international routes, where the ship tends to be the preferred mode of transport.

The system will be used based on a set of starting and ending coordinates, generating a route that will serve as an initial approximation. However, the actual ports where stops may occur, along with agreements with the company and other administrative matters, are external to the system. Consequently, the route will ultimately vary. For this reason, prioritizing a quick response over a response with many checkpoints is ideal, as it will serve as a general guideline to determine whether to circumnavigate a continent by ship or cross it by train, as well as to estimate the overall time, rather than providing real-time guidance. The system will also consider the time required for cargo transshipment between the two modes, taking into account the transported weight and real-time weather conditions to determine the optimal route.

To implement this, several technologies have been chosen, such as Python as the programming language due to its extensive machine learning library support, Power BI for route visualization, and CSV for data storage due to its direct integration with both technologies.

Keywords: Data Science, Routes, Multimodality, Transportation, Predictive Algorithms.

Índice

1. Introducción

Con el auge de la Inteligencia Artificial y el *Machine Learning*, múltiples empresas están apostando por estas tecnologías para mejorar sus servicios o implementar soluciones que en el pasado eran muy difíciles o costosas. Éste es el caso de la empresa FIELDEAS S.L.U, la cual desarrolla una aplicación de ámbito logístico, que permite la gestión de recursos en empresas de transporte o almacenaje, además de ayudar a llevar un registro de rutas realizadas por los conductores y trabajadores de estas empresas.

La empresa Fieldeas[1], posee 22000 usuarios y más de 100 empresas clientes como Repsol, Danone, Seat o DHL, llevando a cabo la gestión de más de 60 millones de transacciones mensualmente. Posee una aplicación móvil disponible tanto en Android como en iOS y una plataforma web, desde que se puede realizar toda la gestión logística. La empresa además, se distribuye en varios departamentos, encargados del desarrollo del software, de la adaptación de este a los clientes finales o de la atención en post-venta entre otros. La empresa se compone de más de medio centenar de empleados.

Fieldeas forma parte del grupo empresarial CIC, con el cual la Universidad de Cantabria lleva años realizando acuerdos de prácticas académicas o de desarrollo de TFGs y se trata una empresa cántabra.

Debido al aumento de clientes multinacionales, los cuales realizan una gran cantidad de rutas de internacionales, transatlánticas y de manera multimodal, la empresa ha decidido crear un sistema que les sirva para crear una línea base de las rutas que se van a realizar en función del inicio y final, y que tenga en cuenta factores como el peso transportado o el clima actual. Este sistema se compondrá de 2 medios de transporte, el ferroviario y el marítimo, al ser los principalmente usados en este tipo de rutas, pues más del 70% de transporte de mercancías es marítimo, y siempre que se pueda para largas y medias distancias por tierra, se recurre a transporte ferroviario por su coste más bajo[2]. Este sistema se utilizará como directriz de partida para conocer el camino ideal que seguiríamos para transportar una mercancía de un punto A a un punto B y también nos permitirá conocer los *checkpoints* que se deberían atravesar en la ruta y el tiempo total estimado. Sin embargo, esta ruta finalmente se verá afectada por restricciones legislativas, esperas en puertos o acuerdos comerciales particulares de la empresa de transporte que no pueden ser predichas por el sistema para idear la ruta, ya que muchas surgen después de establecer la ruta base y contactar con los puertos. Para el transporte por carretera, al tener unos factores muy diferentes que condicionan el cálculo del tiempo estimado de llegada, se está desarrollando por otros empleados de la empresa de manera paralela, y se plantea integrar ambos sistemas conjuntamente en el futuro.

La empresa no posee actualmente datos históricos de rutas realizadas, por lo que es necesario comenzar a recopilarlos dentro de unos años, de modo que el modelo deberá ser reentrenado, sin embargo, se busca obtener un *dataset* sintético con datos lo más próximos posibles a los reales, o con los mismos patrones de relación de cara a poder

[1] Fieldeas. https://www.fieldeas.com/

[2] (20 de julio de 2018) Transporte de mercancías: Comparativa de medios de transporte. Grupo Ibertransit https://ibertransit.com/transporte-de-mercancias-comparativa/

crear un script de limpieza y preparación de datos que pueda utilizarse con pocas modificaciones cuando se disponga de datos reales, del mismo modo que el código que permita analizar y visualizar el conjunto de datos disponibles.

En cualquier sistema de cálculo de rutas, es fundamental conocer, en primer lugar, el contexto en el que se va a realizar el transporte, por ejemplo, si se trata de rutas en coche particular, el conjunto de coordenadas no deberá mapear localizaciones en el mar, en montañas sin acceso para vehículos o vías de tren, deberá ceñirse a carreteras públicas. En este caso sucede lo mismo, necesitamos un conjunto de coordenadas que le sirvan a nuestro sistema para mapear los puntos que puede atravesar un barco, ya que no son todos en los que haya agua, y del mismo modo, para las rutas ferroviarias solo se mapearan las ubicaciones en las que haya vías de tren. Obtener estas coordenadas y adaptarlas a nuestra necesidad será el punto de partida para la realización de este proyecto.

1.1 Metodología

Para desarrollar el proyecto se ha seguido la metodología *Cross-Industry Standard Process for Data Mining* (CRISP-DM), que compone una base para el desarrollo de trabajos de minería de datos. Esta metodología ofrece el proceso del ciclo de vida del proyecto y la descripción del desarrollo de cada fase.[3] Las cuales se reflejan en la Figura 1.

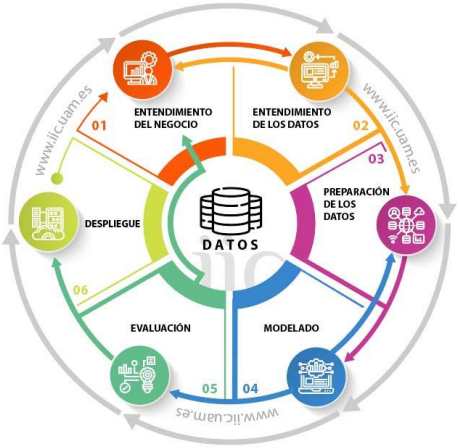

Figura 1: Esquema del ciclo CRISP-DM estándar. [4]

Como observamos en la figura, la metodología divide el proceso de desarrollo de este tipo de proyectos en 6 fases, las cuales se suceden de manera cíclica con cada mejora del sistema, además, se permite el retorno a fases previas del ciclo de vida en varios casos.

- Entendimiento del negocio: En esta fase es necesario obtener conocimientos sobre el contexto y la temática que abordará nuestro proyecto. Todos los factores que

[3] (17 de agosto de 2021) Conceptos básicos de ayuda de CRISP-DM. IBM. https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview
[4] La metodología CRISP-DM en ciencia de datos. Instituto de Ingeniería del conocimiento. https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/

pueden afectarlo o limitarlo, que influirán sobre los datos a analizar o predecir… Es recomendable consultar a un experto en el ámbito y recopilar información de diversas fuentes.

- Comprensión de los datos: En este paso, se debe obtener un conjunto inicial de datos en caso de que no sea aportado al inicio del proyecto. Es importante analizar la calidad de los datos, pues a pesar de ser tratados posteriormente es interesante que estos datos posean toda la información requerida, además, debemos entender que parte de los datos nos interesará para nuestro proyecto y cual deberíamos descartar.

- Preparación de datos: En esta fase se deben seleccionar las características de los datos que nos interesan, por ejemplo en nuestro caso, en el set de datos de coordenadas marítimas, teníamos información sobre el barco que había obtenido el registro de esa coordenada, la cual para este caso no nos servía, por lo que simplemente eliminamos esa columna de nuestro set de datos. A continuación, debemos escoger los valores del conjunto que nos interesan. Esta es la parte más larga del proyecto y quizás la más importante pues la precisión de las predicciones dependerá directamente de ella, debemos tratar de limpiar los casos extremos, ya que a pesar de existir modelos para predecir estos casos, no es nuestro objetivo, o en general, datos que no representan el comportamiento normal del sistema. También dependiendo de los objetivos del proyecto, en este paso se deben encontrar reglas de asociación de los datos, lo cual nos facilitará la labor de limpieza y mejorará la calidad de los datos.

- Modelado: En esta fase se determina el algoritmo que se utiliza para la predicción, se genera el modelo y se entrena, también se establece tanto el formato de entrada de datos que aceptará el modelo como el de salida, y se realizan todas las modificaciones pertinentes sobre los datos para dar la salida adecuada. En esta fase, también se evalúan varios algoritmos o implementaciones de ellos, de cara a corroborar que se ha escogido el más preciso o acorde para las predicciones de nuestro caso de estudio, en caso de identificar uno que se adapte mejor se podrá cambiar nuestra elección.

- Evaluación: En esta etapa se evalúa la calidad de los resultados obtenidos y se analiza la precisión de las predicciones respecto a la muestra de ejemplo aportada en el caso de un proyecto de Machine Learning. En caso de determinar que las predicciones no son de suficiente calidad, se analiza qué aspectos de negocio adicionales se pueden sumar al conjunto de datos para aumentar la precisión de las predicciones. También se debe evaluar si los resultados coinciden con los objetivos de negocio. [5]

[5] Elizabeth León Guzmán, Ph.D . Universidad Nacional de Colombia. Modulo Minería de Datos.
https://disi.unal.edu.co/~eleonguz/cursos/md/presentaciones/Sesion5_Metodologias.pdf

- Despliegue: Esta es la última fase del ciclo, donde se pone en producción el modelo resultado del desarrollo, se integra con las demás herramientas que se utilicen en la aplicación final y se dispone para su funcionamiento.

El formato de la memoria realizada trata de seguir esta metodología en la medida de lo posible. Para el estilo de citas y referencias bibliográficas se ha seguido el estándar *Chicago*[6], por la similitud que presenta con el proceso seguido a la hora de buscar información. Se han utilizado varias fuentes como apoyo de manera global para el proyecto, las cuales constituyen la bibliografía, y otras fuentes particulares para dudas o detalles concretos que han surgido durante el desarrollo, que se indican como notas al pie de página.

1.2 Cronología

La distribución temporal para la realización del TFG se puede observar en la Figura 2:

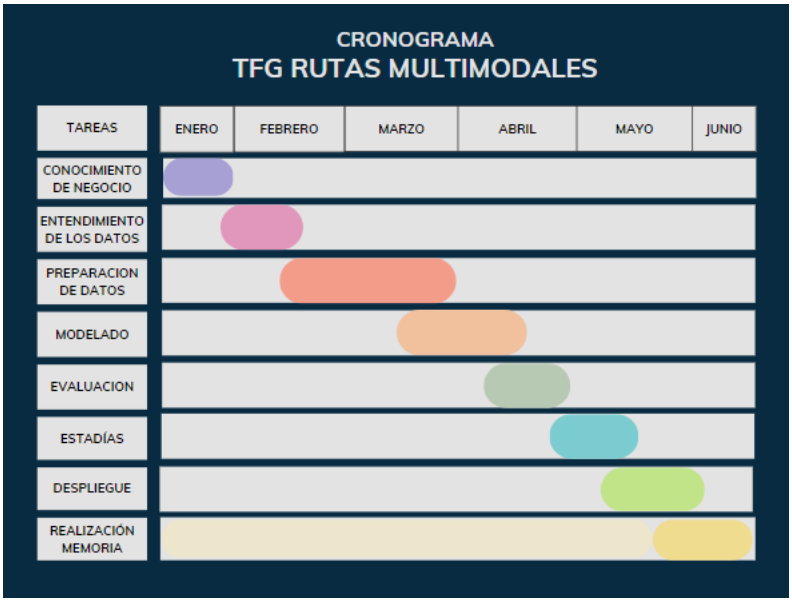

Figura 2: Diagrama de Gantt sobre las fases de desarrollo del TFG. [7]

El TFG se ha realizado a lo largo de varios meses y se han cubierto las 25 horas de trabajo por crédito requeridos para su realización.

La distribución del tiempo para realizar el trabajo se ha centrado especialmente en la preparación de datos, al ser la etapa más larga, esta se ha solapado con la fase de modelado debido a que se ha requerido realizar cambios en los datos procesados al inicio del modelado, al identificar algunos problemas en los datos o aspectos a mejorar la calidad. Lo cual como se ha explicado previamente es algo típico.

---

[6] Scribbr. Chicago Style Citation Guide. https://www.scribbr.com/category/chicago-style/
[7] https://www.canva.com/

El conocimiento del negocio ha sido la primera fase llevada a cabo, a lo largo de medio mes, pues ya había trabajado previamente en la empresa y tenía cierto conocimiento sobre la temática a tratar. Disponiendo del conocimiento de expertos en el ámbito pertenecientes a la empresa, se ha comenzado con el entendimiento de los datos y la búsqueda de estos, se han obtenido en diversas fuentes varios conjuntos de datos potenciales candidatos para ser usados como coordenadas y se ha comprendido cuales son las características buscadas en estos datos.

La fase de preparación de datos comprende tanto la generación del set de datos de coordenadas marítimas como de ferroviarias, la generación de la simulación del histórico de rutas y la creación del código que permite analizar y limpiar estos datos. Durante el desarrollo de esta fase se solapó la etapa de modelado, pues se encontraron formas de adaptar los datos para obtener mejores resultados.

De la misma forma, la etapa de evaluación y la de modelado también se solaparon, al encontrar formas de mejorar el modelo gracias a su evaluación, la cual duró casi un mes.

Tras esto, se realizó el mismo proceso para la realización del modelo e histórico de estadías. Para ello se pudo reutilizar gran parte del código con leves adaptaciones. De este modo, su realización llevó poco tiempo en comparación con todo el proceso previo.

La última fase es el despliegue, en la cual se llevó a cabo toda la puesta en conjunto del sistema, además de generar la visualización y obtener los climas, esta fase llevó cerca de un mes.

La fase final es la de realización de la memoria, la cual abarca medio mes, sin embargo, este es el periodo durante el que se realiza de manera específica, obteniendo la versión final, pues a lo largo de todo el proyecto se fue redactando el trabajo desarrollado a modo de borrador, por lo que para cuando llegó el momento de centrarse en ello, ya existía una versión completa, la cual solo necesitó revisiones.

1.3 Objetivos y restricciones

Respecto a los objetivos, el propósito global del proyecto consiste en la realización de un sistema de cálculo de rutas y tiempo de llegada que en el futuro pueda ser integrado en una aplicación logística. Las salidas del sistema deben ser el ETA (Tiempo estimado de llegada, en inglés *Estimated Time Arrival* de ahí las siglas), y la ruta que deberá tomar, en concreto los requisitos son:

- Cálculo de rutas: El sistema deberá ser capaz de calcular la ruta que el medio de transporte debe realizar dados el inicio y el fin.

- Cálculo del ETA: El sistema deberá ser capaz de obtener una aproximación del tiempo que se tardará en realizar la ruta.

- Multimodalidad: El sistema deberá tener en cuenta que el transporte a nivel internacional se pueda realizar tanto en tren como en barco. Considerando, además, el tiempo que se tarda en realizar el intercambio de mercancías entre

estos dos medios de transporte. El sistema de transporte por carretera se está implementando en la empresa por otro grupo de trabajo.

Respecto a las restricciones, encontramos algunas que nos limitan u obligan a realizar soluciones temporales. Por ejemplo, no se dispone de un histórico de rutas, ni un set de coordenadas, por lo que se plantea comenzar a recogerlas un tiempo después de la finalización del desarrollo del sistema. Por tanto el modelo necesitará ser reentrenado con los nuevos datos, se cuenta con que los resultados que arroje actualmente pueden distar de los reales, por lo que de momento solo se busca que el sistema funcione y arroje una salida adecuada.

Para tratar de aproximar los resultados a un caso real dentro de las posibilidades, se genera una simulación del set de datos histórico, aportando cierta variabilidad, guiado por un miembro de la empresa con gran experiencia en el campo. De este modo también se puede generar un sistema de limpieza y análisis de los datos lo más similar posible al que se deba implementar cuando el set de datos sea en base a un histórico real.

2. Entendimiento de negocio

El primer paso del proyecto es comprender el contexto y el ámbito de estudio que se va a tratar. Para ello, se han mantenido varias reuniones con miembros de la empresa que tienen mayor conocimiento en este campo, permitiendo obtener una visión general del tema, que, en este caso, se trata de sistemas logísticos multimodales. Por este motivo, es necesario comprender tanto las tecnologías como las restricciones que se aplican sobre cada medio de transporte a la hora de realizar las rutas. También se deben identificar los factores que rodean el proceso y que pueden afectar en alguna medida. Por ello, es necesario comprender cierta terminología sobre el contexto.

- **ETA**: Acrónimo de *Estimated Time Arrival,* en castellano "Tiempo de llegada estimado"[8], sin embargo en este contexto también es utilizado como la duración total del trayecto desde la salida del lugar de partida hasta la finalización del trayecto global. En este sistema, el ETA se referirá al tiempo total de trayecto, el cual es el que se busca calcular en este caso.

- **Estadía:** Utilizado principalmente en transportes marítimos, se trata del tiempo que un transporte se encuentra estacionado en un puerto o cualquier lugar en el que se vaya a realizar una descarga de los productos que transporte. [9]

[8] ETA. Logistiche Informationssysteme Iberia, S.L.U. https://www.lis.eu/es/lexikon/eta/#:~:text=La%20abreviatura%20ETA%20significa%20Tiempo,transporte%20en%20las%20condiciones%20actuales.

[9] Jorge Selma. (2 de diciembre de 2003). ¿Qué se entiende por estadías o sobreestadías? Veintepies. https://www.veintepies.com/secciones/blegal_more.php?id=M6855_0_20_0_C

- **OMI:** Acrónimo de "Organización Marítima Internacional". Su función es establecer un conjunto de reglas y normativas para garantizar la eficacia, seguridad, respeto medioambiental e innovación en el transporte marítimo internacional, el cual representa un 80% del transporte de mercancías mundial.[10]

- **AIS:** Es el acrónimo de "Sistema de identificación automática", se trata de un sistema que transmite periódicamente y en tiempo real la ubicación de un barco, con el propósito de evitar colisiones, conocer la última ubicación conocida en caso de accidente o conocer tiempos de llegada. Es exigido por la OMI en buques de más de 300 toneladas. [11]

- **IMO:** Se trata de un número de identificación de barcos que tiene como objetivo la seguridad marítima, limitar la contaminación y evitar el fraude marítimo. Se trata de un número permanente, sin embargo, no es obligatorio para todos los tipos de embarcaciones. [12]

- **MMSI:** *Marine Mobile Service Identity*. Se trata de un número de 9 dígitos con información sobre el país de procedencia del buque, el país donde se encuentra en ese momento y el propio barco. Se transmite de manera digital e identifica al barco a nivel de telecomunicaciones. [13] La diferencia con el IMO radica en su propósito, identificar una estación de radio en el MMSI y evitar fraude en el IMO, además de que el IMO es permanente mientras que el MMSI varía según el país en el que se ubique el barco.[14]

3. Tecnologías

Para la realización de este proyecto se han debido escoger las tecnologías que se quisiera emplear en el desarrollo, no habiendo ninguna restricción al respecto. Por este motivo, se ha realizado la debida investigación previa para tomar la decisión adecuada. Además, se han utilizado algunas otras tecnologías cuya necesidad se ha identificado durante el propio proceso de desarrollo. A continuación se detallan estas tecnologías.

[10] Introducción a la OMI. IMO. https://www.imo.org/es/about/Pages/Default.aspx
[11] ¿Qué es AIS?. Global Fishing Watch. https://globalfishingwatch.org/es/faqs/que-es-ais/
[12] IMO identification number schemes. IMO. https://www.imo.org/en/ourwork/msas/pages/imo-identification-number-scheme.aspx
[13] MMSI: Qué Es Y Para Qué Sirve?. Promonautica. https://www.promonautica.com/blog/35-post/35_mmsi-que-es-y-para-que-sirve?page_type=post%23
[14] (12 de julio de 2022) What is the Difference Between IMO and MMSI?. Sinay. https://sinay.ai/en/what-is-the-difference-between-imo-and-mmsi/

3.1 Programación

Python: Como lenguaje de programación se ha decidido escoger *Python*. El motivo es su gran soporte para tareas de *Machine Learning* y *Data Science* en general, lo cual facilita en gran medida la realización del proyecto y lo dinamiza.

Anaconda: Se trata de la distribución del lenguaje de programación *Python*, la ventaja que aporta respecto a otras es su facilidad en la gestión e implementación de paquetes, además de tener previamente incorporados varios dedicados a *Data Science* de este modo se agiliza la ejecución de código y se facilita la programación y desarrollo. [15]

Jupyter Notebooks : Se trata de una aplicación cliente-servidor cuyo objetivo es enlazar el desarrollo de código por celdas con celdas de texto de manera interactiva[16], de modo que se pueda organizar, esquematizar y explicar el código de forma más dinámica y completa que como se haría en un documento de *Python* normal. [17] Los ficheros de *Jupyter Notebooks* se almacenan con formato ".ipynb", el cual se guarda internamente en formato JSON. [18]

Se ha escogido esta tecnología para desarrollar el código *Python* por la facilidad que supone para separar el código en regiones e identificar la funcionalidad o propósito de cada región, de este modo se pueden analizar de manera separada y esquemática las salidas por consola de las ejecuciones, facilitando el entendimiento de su funcionamiento además de la depuración del código.

Visual Studio Code: Se trata de un editor de texto desarrollado para la programación como principal objetivo. Se han valorado otras alternativas como Atom, o Pycharm, sin embargo se ha escogido *Visual Studio Code* puesto que posee grandes facilidades para la integración de *Plug-ins*, o herramientas externas que ayuden a facilitar la labor de programación, como por ejemplo el propio *Jupyter NoteBooks*, además posee un autocompletado bastante útil para agilizar la producción de código.

---

[15] Izary Rondón. (4 de febrero de 2022). ¿Qué es Anaconda?. EIP. https://eiposgrados.com/blog-python/que-es-anaconda/

[16] The Jupyter Notebook. Jupyter. https://jupyter-notebook.readthedocs.io/en/latest/notebook.html

[17] (28 de febrero de 2019). Jupyter Notebook: documentos web para análisis de datos, código en vivo y mucho más. Ionos. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/jupyter-notebook/

[18] IPYNB. FILEFORMAT. https://docs.fileformat.com/es/word-processing/ipynb/#qu%c3%a9-es-un-archivo-ipynb

3.2 Almacenamiento de Datos

Pandas Dataframe: Utilizado para el almacenamiento de datos temporal, se trata de una estructura de datos basada en tablas, con organización por filas y columnas[19]. Una de las grandes ventajas de este tipo de almacenamiento de datos temporal es la versatilidad que ofrece, al ser aceptado por la gran mayoría de librerías utilizadas para desarrollo en *Machine Learning* sin ningún tipo de transformación. También ofrece de manera nativa una enorme cantidad de métodos para trabajar con la estructura de datos sin tener que iterarlo manualmente o implementar la función de manera externa.

CSV: Utilizado para el almacenamiento de datos permanente. Al utilizar una estructura de tabla única, posee un *casting* directo con un *Dataframe* de Pandas, lo que permite su importación o exportación de un set de datos a una enorme velocidad, lo cual es muy importante a la hora de trabajar con un volumen de datos tan grande.

3.3 APIs/Librerías

Weather Forecast API [20]: Se utiliza para obtener el clima en las diferentes coordenadas geográficas de nuestro set de coordenadas transitables, con el propósito de aumentar la precisión del sistema de rutas al tener en cuenta dicha variable. Esta API permite obtener mucha información respecto al clima, como el viento a diferentes alturas, la nubosidad, humedad relativa, visibilidad, precipitaciones… En nuestro caso nos interesan tres parámetros de la petición a la API.

- *latitude*: La latitud de la coordenada de donde queremos conocer el clima.
- *longitude*: La longitud de la coordenada de donde queremos conocer el clima.
- *current_weather*: Su valor debe ser *true* para que nos devuelva el código del clima actual, ese código se puede encontrar en la documentación de la API como WMO, siglas de *Weather interpretation codes*, y debe ser adaptado por el sistema a las categorías que nuestro sistema utiliza.

XGBoost: Se trata de una librería centrada en la eficiencia, flexibilidad y portabilidad. Implementa algoritmos basados en *Gradient Boosting*, una técnica de *Machine Learning* usada en regresión y clasificación que funciona con múltiples árboles de decisión entrenados de manera secuencial, de modo que cada uno mejore los errores y la precisión de los árboles previos[21]. Por otro lado, el código se ejecuta en varios entornos de ejecución distribuidos para aumentar la velocidad. [22]

---

[19] DataFrame Documentation. Pandas. https://pandas.pydata.org/docs/reference/frame.html
[20] Weather Forecast API. Open-Meteo. https://open-meteo.com/en/docs
[21] Joaquín Amat Rodrigo (Octubre de 2020). Gradient Boosting con Python. https://www.cienciadedatos.net/documentos/py09_gradient_boosting_python.html
[22] XGBoost Documentation. https://xgboost.readthedocs.io/en/stable/

Sklearn: Se trata de una biblioteca de aprendizaje automático que contienen varios algoritmos de *Machine Learning*. Está perfectamente integrada con la librería numpy[23], de ella se han importado y usado varios módulos:

- *tree*: Módulo que implementa un árbol de decisión, con varios métodos para su entrenamiento, modificación y trabajo con el mismo.
- *model_selection*: Contiene métodos para separar los datos de aprendizaje y de validación de los modelos.
- *metrics*: Este módulo posee métodos que permiten evaluar el correcto funcionamiento del modelo predictivo y adecuación a la realidad. Múltiples métricas de evaluación.

Networkx: Librería utilizada para generar y trabajar con grafos, cabe destacar los métodos *dijkstra_path* que implementa el algoritmo de Dijkstra a partir de un grafo implementado con el objeto *Graph,* también presente en la librería, y *path_weight,* que a partir de la ruta mínima encontrada, devuelve su peso total.

Otras: Además, se han utilizado otras librerías cuyo uso ha sido particular y circunstancial como *math* para el uso de funciones matemáticas ya implementadas, *tqdm* para generar barras de progreso, *matplotlib* y *seaborn* para generar visualizaciones del grafo y gráficas, *scipy* la cual posee el cálculo de distancias en un plano previamente implementado o *request* para automatizar las peticiones a servicios web.

4. Preparación de datos

El primer paso en el desarrollo del sistema es obtener un conjunto de coordenadas que puedan ser transitadas por los medios de transporte. Para esto no vale cualquiera, pues las rutas marítimas están fijadas por ley y un tren solo podrá ir por donde haya vías de tren.

Para generar el set de datos de coordenadas marítimas primero se ha debido buscar un conjunto de coordenadas. Partiendo del archivo "routes.csv" obtenido a partir de datos de los AIS de varios barcos que se encuentra en el artículo[24] de Alexei Novikov, se ha reducido parte de la información para preservar solo la que nos interesa. Esta es la que se encuentra en los campos "lat" y "lon" del CSV, correspondiendo con la latitud y longitud respectivamente. Sin embargo, consta de 1,5 millones de entradas aproximadamente, puesto que su objetivo es diferente al nuestro. Para disminuir el volumen de datos en primer lugar se ha aplicado una reducción aleatoria, pues lo que nos interesa es tener una vista global de las coordenadas marítimas. Posteriormente, a las entradas resultantes, se les ha aplicado otro script de limpieza, en este caso para eliminar los grupos de coordenadas que se encuentren muy cercanos, dejando una única coordenada representativa. De este modo, evitamos tener demasiada precisión en el mapeado de un área y menos en otro, y por tanto que la distribución sea lo más uniforme

[23] Universidad de Alcalá. https://www.master-data-scientist.com/scikit-learn-data-science/
[24] (15 de mayo de 2019) Alexei Novikov. Creating sea routes from the sea of AIS data. https://towardsdatascience.com/creating-sea-routes-from-the-sea-of-ais-data-30bc68d8530e

posible con un conjunto de coordenadas manejable. El conjunto de coordenadas resultante se ha comprobado empíricamente con *Google Earth* de manera visual, como se observa en la Figura 3, adaptándolo levemente al formato que requiere esta aplicación para importarlas.

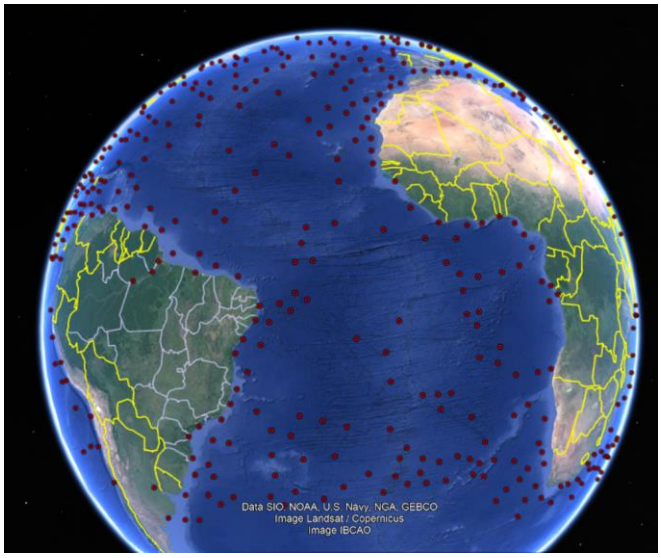

Figura 3: Conjunto de coordenadas transitables marítimas procesado.

El procedimiento con las coordenadas ferroviarias ha sido el mismo, en este caso se ha extraído de una base de datos logística encontrada en una página web[25]. El *dataset* en este caso no es de un histórico de rutas realizadas, si no directamente un mapeo de todas las vías de tren del mundo.

También se ha verificado empíricamente, como se observa en la Figura 4.

Figura 4: Visualización del conjunto de coordenadas ferroviarias.

[25] Global railways (WFP SDI-T - Logistics Database. HDX. https://data.humdata.org/dataset/global-railways.

4.1 Generación de Dataset Sintético

Al no disponer de un set de datos con un histórico de tiempo que haya tomado la realización de diferentes tramos marítimos y ferroviarios, se ha decidido que la mejor opción en este caso era generar un set de datos sintéticos, el cual se trata de un conjunto de datos artificial generado por un algoritmo o script, usado cuando no se dispone de datos reales para entrenar un modelo[26].

Para generar estos datos, se ha partido del conjunto de coordenadas previamente obtenido. Se ha obtenido la distancia de todos los puntos entre sí para generar los tramos.

La distancia se obtiene como la hipotenusa del triángulo rectángulo formado por la distancia de las dos coordenadas en el eje X como un cateto, eje Y como otro cateto, y la conexión de ambos puntos como hipotenusa. Esto se conoce como distancia euclídea y se puede obtener al estar trabajando con un espacio bidimensional de coordenadas cartesianas[27]. La visualización de esta fórmula se observa en la Figura 5, la cual muestra el triángulo descrito por estas coordenadas.

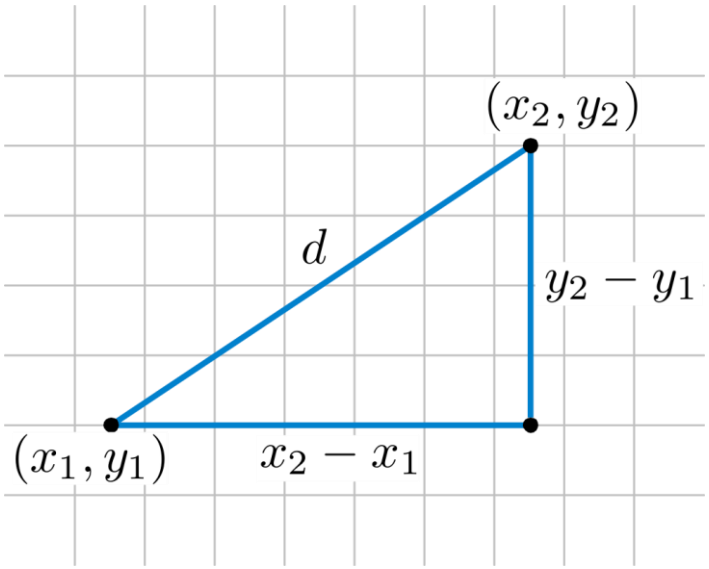

Figura 5: Representación de distancia euclídea.[28]

A partir de estos tramos, se ha dado una velocidad media inicial escogida aleatoriamente para cada tramo, en función de un rango entorno a la velocidad media de un tren de mercancías y de un barco de mercancías. Esta es la única diferencia entre la generación del set de datos ferroviario y marítimo, pues las condiciones siguientes aplican de igual manera.

---

[26] (22 de marzo de 2022) Alex Watson. How to Generate Synthetic Data: Tools and Techniques to Create Interchangeable Datasets. https://gretel.ai/blog/how-to-generate-synthetic-data-tools-and-techniques-to-create-interchangeable-datasets

[27] (4 de diciembre de 2019) Alberto Cajal. Distancia euclidiana: concepto, fórmula, cálculo, ejemplo. https://www.lifeder.com/distancia-euclidiana/

[28] Jim belk. An illustration of the distance formula on the plane. https://es.wikipedia.org/wiki/Distancia_euclidiana#/media/Archivo:Distance_Formula.svg

Posteriormente, se han aplicado correcciones sobre este ETA inicial calculado, siempre trabajando con rangos de datos cuyo valor concreto se calcula aleatoriamente para cada fila del set de datos, de este modo, logramos aumentar la variabilidad de los datos. Los rangos de corrección han sido asesorados por miembros de la empresa con conocimiento del campo logístico.

Los campos que se han obtenido aleatoriamente y sobre los que posteriormente se ha aplicado una corrección en función del valor son los siguientes:

- Clima: Puede ser soleado, lluvia, nuboso, niebla, tormenta y huracán.
- Fecha: Se ha obtenido una fecha aleatoria aunque posteriormente es la estación del año la que afectará al ETA, la cual se corresponderá a un conjunto de meses.

En este punto, la estructura del set de datos es la que se visualiza en la Figura 6.

	punto_x	lat_x	long_x	modo_x	punto_y	lat_y	long_y	modo_y	distance	speedKn	speedKmH	Climate	eta	Date
0	1	45.55623	13.713285	barco	2	39.541505	-28.197625	barco	4234.030345	8.362513	15.487373	Cloudy	234.608175	1995-08-24
1	1	45.55623	13.713285	barco	3	14.560962	-73.199102	barco	9227.388374	14.645222	27.122951	Rain	470.893723	1998-11-13
2	1	45.55623	13.713285	barco	4	-32.528985	-71.680493	barco	11571.256710	10.578405	19.591206	Cloudy	442.090497	1997-04-13
3	1	45.55623	13.713285	barco	5	9.373852	52.530890	barco	5306.572300	13.387801	24.794207	Foggy	236.427997	1995-08-02
4	1	45.55623	13.713285	barco	6	2.052465	57.105505	barco	6144.479086	8.888863	16.462175	Sunny	234.413137	2006-09-14

Figura 6: Muestra del set de datos de entrenamiento final para el caso marítimo.

Por último, se han introducido datos extremos aplicando una modificación sobre el ETA con rango muy bajo en algunas filas y en otras un rango muy alto.

4.2 Análisis y tratamiento de datos

Se ha generado un script (DataAnalyser.ipynb) que analiza y visualiza el set de datos de entrenamiento obtenido. Probablemente deba ser modificado en el futuro al utilizar un set de datos reales, ya que, podrían encontrarse nuevas reglas de asociación o que condicionen a la distribución de los datos. Sin embargo, es útil para los objetivos de este proyecto, y a efectos de punto de partida en una posterior implementación real.

En primer lugar, para comparar los datos, tenemos que encontrar una unidad de medida común, pues el ETA depende de la distancia, por lo que para posteriormente limpiar los datos es más conveniente una unidad de medida que los estandarice. Esta es el "eta/km" que se refiera al tiempo que se tardó en recorrer cada kilómetro de media en ese registro de ese tramo. Los valores quedan entorno a las centésimas, por lo que obtenemos el "eta/km*100" para que queden entorno a las unidades y sean más fáciles de interpretar por un ser humano, esto se debe a que, un valor decimal es más complejo de comprender por la mayor longitud del número[29].

Realizamos una visualización de los datos en función del mes y el clima usando la librería *matplotlib*.

[29] Roell, M., Viarouge, A., Hilscher, E. et al. Evidence for a visuospatial bias in decimal number comparison in adolescents and in adults. Sci Rep 9, 14770 (2019). https://doi.org/10.1038/s41598-019-51392-6

Como podemos ver en la Figura 7, tenemos varios valores lejanos al conjunto principal, estos son los casos extremos que nos interesa eliminar, pues no representan el comportamiento típico.

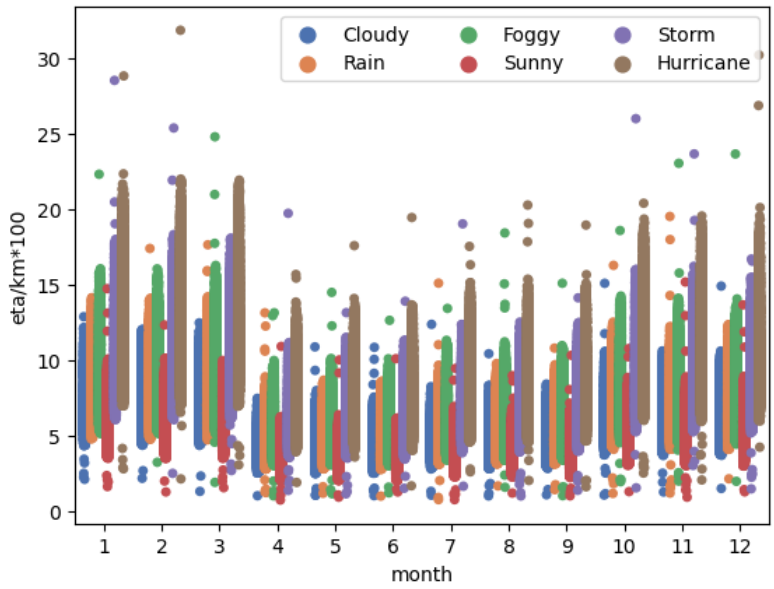

Figura 7: Visualización de set de datos marítimo antes de limpiarlo, en función del mes y el clima.

Observamos que para cada conjunto de 3 meses los datos son similares, lo cual es lógico, pues al generar los datos se ha trabajado por estaciones, ya que se trata de una regla que existe en la realidad.

Para asegurarnos obtenemos las medias del ETA por cada mes y clima, en la Figura 8 observamos la situación para dos climas.

Climate	month		Foggy	1	9.038359
Cloudy	1	7.161901		2	9.070389
	2	7.128837		3	9.062111
	3	7.102734		4	5.459169
	4	4.257140		5	5.435496
	5	4.281527		6	5.431002
	6	4.280202		7	6.052543
	7	4.744022		8	6.021477
	8	4.757987		9	6.068395
	9	4.769496		10	7.840989
	10	6.191676		11	7.874109
	11	6.184903		12	7.890089
	12	6.168026			

Figura 8: Media de ETA/KM*100 por cada mes y clima. (Para resultados completos ver "DataAnalyser.ipynb")

Se observa que la media es similar cada tres meses, sin embargo debemos contrastar esto con matemáticas.

Primero comprobamos la desviación total, la cual se muestra en la Figura 9:

```
Desviacion total = 2.81361249630841
```

Figura 9: Desviación total del set de datos.

Vemos que es alta. Por lo que si trabajásemos con todo el volumen de datos en conjunto, eliminaríamos datos representativos. Además de que visualizando la gráfica observamos que los datos según mes y clima se distribuyen en rangos muy diferentes, si eliminásemos los que superen cierto percentil únicamente estaríamos quitando los de un clima y mes concreto, dejando casos extremos para otros climas y meses sin limpiar, por lo que debemos buscar la manera de separar los datos en grupos con característica similares, para que puedan ser limpiados en conjunto.

A continuación, analizamos la desviación en función del clima, la cual se muestra en la Figura 10.

Climate	
Cloudy	1.754196
Foggy	2.282028
Hurricane	3.125975
Rain	2.016192
Storm	2.582802
Sunny	1.446249

Figura 10: Desviación en función del clima.

La desviación según el clima, en algunos casos es más baja pero seguiríamos teniendo el problema previo, en especial para el clima huracanado, por lo que la siguiente aproximación sería analizarla por clima y mes, como se muestra en la Figura 11.

Climate	month		Foggy	1	2.183850
Cloudy	1	1.657200		2	2.205652
	2	1.672824		3	2.193437
	3	1.664673		4	1.359758
	4	1.007669		5	1.358726
	5	1.025925		6	1.344481
	6	1.019586		7	1.495728
	7	1.109524		8	1.499445
	8	1.130308		9	1.494679
	9	1.127474		10	1.905052
	10	1.455956		11	1.909602
	11	1.443045		12	1.928145
	12	1.445278			

Figura 11: Desviación por clima y mes.

Al obtener la desviación en función de clima y mes vemos que generalmente es más baja, por lo que nos permitiría trabajar con mayor precisión sobre más datos que si trabajásemos con solo el clima.

Sin embargo y como es lógico vemos que la desviación de cada 3 meses es casi idéntica, lo que nos permite, sumándole el hecho de que la media es similar, ver que son conjuntos de datos muy similares y que podrían trabajarse en conjunto. Para asegurarnos que siempre se cumple, obtenemos la desviación de las desviaciones cada 3 meses en mismo clima, la cual debería ser cercana a 0, estos resultados se muestran en la Figura 12.

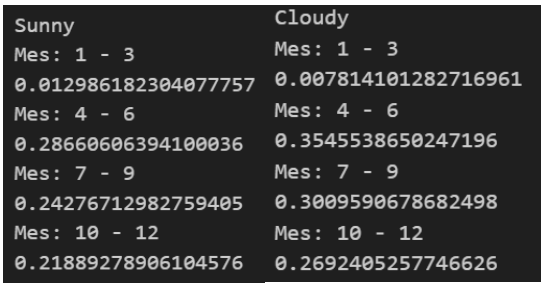

Figura 12: Desviación de desviaciones por clima y conjuntos de 3 meses.

Para realizar la limpieza utilizaremos los percentiles de estos conjuntos de datos, de este modo observamos el momento en que los valores de los datos se disparan y eliminamos los que se encuentren por debajo o por encima de dicho percentil, según estemos analizando los casos límite y no representativos superiores o los inferiores, obtenemos varios percentiles, además de otros datos como la media y la mediana, los cuales observamos en la Figura 13, que nos ayuden a comprender como de distribuyen los datos, en base al mismo clima y 3 meses, ya que como se ha verificado previamente, poseen un comportamiento similar, de cara a poder identificar en que percentil se establece el punto de corte para la limpieza de los datos.

Con clima soleado:

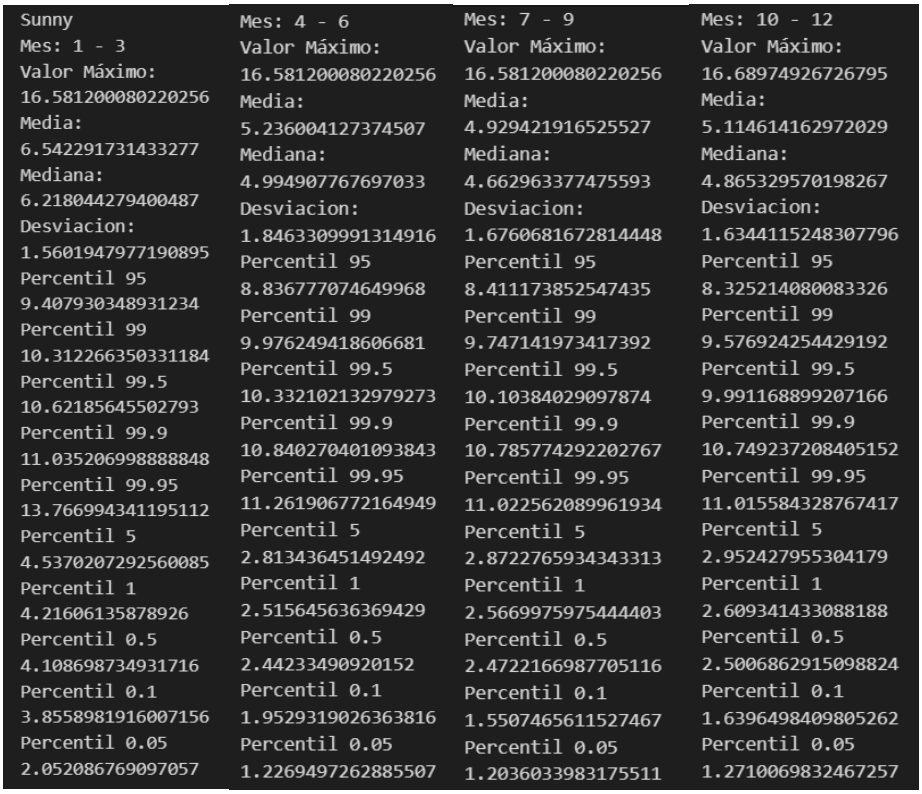

Figura 13: Conjunto de datos representativos para cada clima y conjunto de 3 meses, en este caso para clima soleado.

Este comportamiento se repite con los demás climas.

Observamos que por debajo del percentil 0.1, hay un salto bastante grande en los valores que no se corresponde con la progresión en los percentiles previos.

Lo mismo ocurre en el percentil 99.9, por esto deducimos que el comportamiento típico se encuentra entre el percentil 0.5 y 99.5, y ya que este sistema no pretende predecir casos extremos, se decide limpiarlos.

Realizamos la limpieza de estos datos y observamos el resultado del conjunto de datos. En la Figura 14, visualizamos como queda el set de datos tras la limpieza, con lo que se comprueba que se ha disminuido en enorme medida los casos extremos y, por tanto, hemos obtenido un set de datos mucho más uniforme.

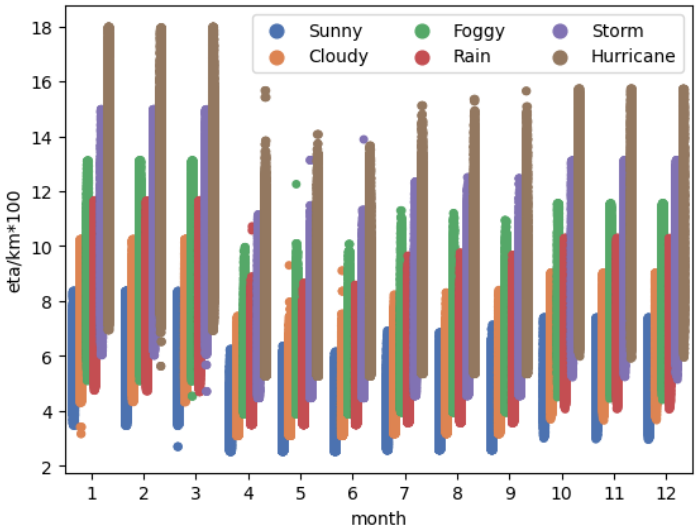

Figura 14: Visualización de datos tras su limpieza.

Como observamos en la Figura 15, los conjuntos de datos han quedado más uniformes que previamente.

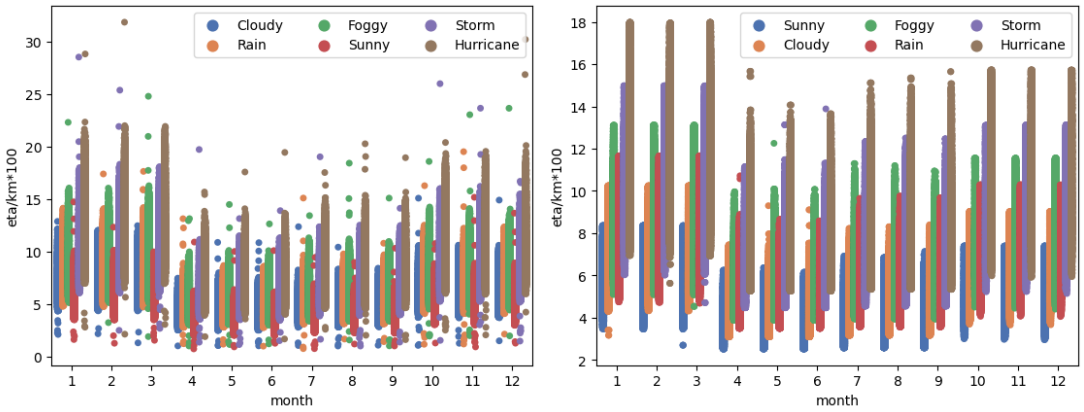

Figura 15: Comparativa de datos antes (izquierda) y después (derecha) de la limpieza.

5. Modelado

Para entrenar el modelo con el histórico generado sintéticamente, se han probado varios algoritmos, comprobando cual funcionaba mejor.

En primer lugar, la aproximación ha sido obtener directamente el valor del ETA como salida del algoritmo. Sin embargo, posteriormente se ha observado, que predecir el valor del "eta/km" y luego multiplicarlo por los kilómetros totales, es una aproximación que arroja resultados mucho más precisos.

Para resolver el problema de la predicción del ETA, existe una enorme cantidad de algoritmos de muchas clases diferentes. Para resolver este problema se ha seleccionado un conjunto manejable de ellos, por haberse utilizado en otros proyectos de la empresa y por ser cada uno de una clase diferente que podría adaptarse a nuestro problema, en este caso regresión, clasificación y red neuronal. Al principio se ha probado con la implementación más básica de esa clase de algoritmo, y al encontrar cuál de las clases es la que arroja mejores resultados, árboles de decisión como se explicará posteriormente, se ha probado con una implementación más moderna y optimizada para comprobar si nos mejora la solución, este es XGBoost, basado en árboles de decisión.

5.1 Multiple Linear Regression

Se trata de la primera aproximación que se ha utilizado para resolver el problema. Este algoritmo obtiene una línea recta en la evolución de un valor respecto a dos o más valores independientes[30], añadiendo dimensiones. En la Figura 16 se puede visualizar cómo funciona el modelo para un caso con 3 variables, de las cuales la porosidad y el porcentaje de vitrinita son independientes, y la producción de gas depende de ellas[31].

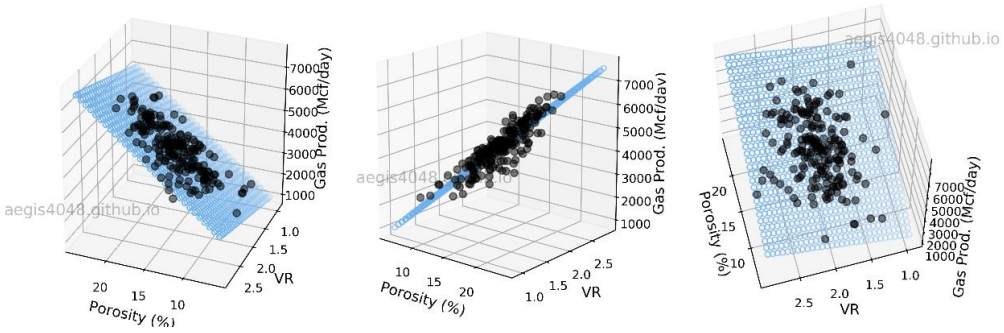

Figura 16: Visualización de un modelo de regresión lineal múltiple.[32]

[30] Montero Granados. R (2016): Modelos de regresión lineal múltiple. Documentos de Trabajo en Economía Aplicada. Universidad de Granada. España.
[31] Rebecca Bevans. 20 de febrero de 2020. Multiple Linear Regression | A Quick Guide. https://www.scribbr.com/statistics/multiple-linear-regression/#:~:text=What%20is%20multiple%20linear%20regression,variables%20using%20a%20straight%20line
[32] Eric Kim. 18 de noviembre de 2019. Multiple Linear Regression and Visualization in Python https://aegis4048.github.io/mutiple_linear_regression_and_visualization_in_python

De este modo es capaz de detectar la tendencia de los valores de manera lineal y obtener una estimación de aquellos valores de los que no exista una correspondencia exacta en el histórico.

Al comprobar el funcionamiento de este algoritmo, se observa que para valores altos funciona muy bien, sin embargo para valores pequeños, con tramos de distancia muy pequeña, arroja valores negativos, como el 16, 17 y 19 de la Figura 17.

	distance	Date	month	Cloudy	Foggy	Hurricane	Rain	Storm	Sunny	Predicted eta
0	7000.246935	1998-05-29	5	0	0	0	0	1	0	620.942194
1	14000.246935	1998-05-29	5	0	0	0	0	1	0	1103.621306
2	7000.246935	1998-05-29	5	1	0	0	0	0	0	345.025122
3	14000.246935	1998-05-29	5	1	0	0	0	0	0	827.704234
4	7000.246935	1998-05-29	5	0	1	0	0	0	0	508.583727
5	14000.246935	1998-05-29	5	0	1	0	0	0	0	991.262838
6	7000.246935	1998-05-29	5	0	0	1	0	0	0	784.146247
7	14000.246935	1998-05-29	5	0	0	1	0	0	0	1266.825358
8	7000.246935	1998-05-29	5	0	0	0	1	0	0	426.599862
9	14000.246935	1998-05-29	5	0	0	0	1	0	0	909.278974
10	7000.246935	1998-05-29	5	0	0	0	0	0	1	233.579973
11	14000.246935	2005-05-29	5	0	0	0	0	0	1	716.259085
12	14000.246935	2005-09-29	9	0	0	0	0	0	1	632.547067
13	14000.246935	2005-12-29	12	0	0	0	0	0	1	569.763054
14	14000.246935	2005-01-29	1	0	0	0	0	0	1	799.971102
15	14000.246935	2005-04-29	4	0	0	0	0	0	1	737.187089
16	90.311833	2023-04-26	4	0	0	0	0	0	1	-221.960784
17	90.311833	2005-04-26	4	0	0	0	0	0	1	-221.960784
18	90.311833	2005-04-26	4	0	0	0	1	0	0	-28.940895
19	20.311833	2005-04-26	4	0	0	0	0	0	1	-226.787575

Figura 17: Valores predichos con Regresión Lineal Múltiple.

5.2 Arboles de decisión

El siguiente algoritmo que se ha probado son los árboles de decisión. Se ha utilizado la implementación *Decission Tree* Regressor, de *sklearn*, por ser una versión básica del mismo, sin utilizar otros algoritmos o técnicas de apoyo. De este modo, se puede utilizar como punto de partida para conocer cómo se comporta esta familia de algoritmos ante nuestro problema. Este algoritmo debería solucionar el problema de los valores negativos, sin embargo, tendrá la desventaja de arrojar la misma predicción para diferentes conjuntos de datos a predecir, pues su funcionamiento es comprobar a qué entrada del histórico se parece más la entrada a predecir. De este modo, como observamos en la Figura 18, dos o más predicciones sobre valores similares podrán dar el mismo valor.

	distance	Date	month	Cloudy	Foggy	Hurricane	Rain	Storm	Sunny	Predicted eta
0	7000.246935	1998-05-29	5	0	0	0	0	1	0	441.919342
1	14000.246935	1998-05-29	5	0	0	0	0	1	0	904.274475
2	7000.246935	1998-05-29	5	1	0	0	0	0	0	311.630554
3	14000.246935	1998-05-29	5	1	0	0	0	0	0	640.888733
4	7000.246935	1998-05-29	5	0	1	0	0	0	0	404.192841
5	14000.246935	1998-05-29	5	0	1	0	0	0	0	784.635315
6	7000.246935	1998-05-29	5	0	0	1	0	0	0	535.914978
7	14000.246935	1998-05-29	5	0	0	1	0	0	0	1080.744629
8	7000.246935	1998-05-29	5	0	0	0	1	0	0	358.103729
9	14000.246935	1998-05-29	5	0	0	0	1	0	0	716.289368
10	7000.246935	1998-05-29	5	0	0	0	0	0	1	252.488205
11	14000.246935	2005-05-29	5	0	0	0	0	0	1	516.639893
12	14000.246935	2005-09-29	9	0	0	0	0	0	1	553.732300
13	14000.246935	2005-12-29	12	0	0	0	0	0	1	739.718140
14	14000.246935	2005-01-29	1	0	0	0	0	0	1	788.169067
15	14000.246935	2005-04-29	4	0	0	0	0	0	1	517.053528
16	90.311833	2023-04-26	4	0	0	0	0	0	1	3.302933
17	90.311833	2005-04-26	4	0	0	0	0	0	1	3.302933
18	90.311833	2005-04-26	4	0	0	0	1	0	0	10.692027
19	20.311833	2005-04-26	4	0	0	0	0	0	1	3.302933

Figura 18: Valores predichos con Árboles de decisión.

Este problema se observa claramente en los datos del final de la tabla, donde 3 datos con valores diferentes arrojan el mismo valor. El problema está en que dos de las entradas son idénticas y solo se diferencian en la distancia, donde una de ellas es aproximadamente 4,5 veces mayor que la otra y sin embargo el resultado es el mismo.

5.3 XGBoost

Se trata de un algoritmo de aprendizaje automático basado en clasificación y regresión. Está basado en árboles de decisión, potenciados con refuerzo de gradiente, lo cual consiste en la creación de múltiples modelos débiles, para a partir del error de cada uno, generar un modelo mucho más preciso como la combinación de todos[33]. Su funcionamiento es mejor cuanto mayor sea el *dataset* de entrenamiento.[34] La ventaja que supone respecto a los otros algoritmos predictivos que se han probado en este proyecto, es la gran cantidad de personalización y parámetros de optimización que posee, esto se conoce como *fine tuning*. Sin embargo, el problema persiste, como observamos en la Figura 19.

	distance	Date	month	Cloudy	Foggy	Hurricane	Rain	Storm	Sunny	Predicted eta
0	7000.246935	1998-05-29	5	0	0	0	0	1	0	441.919342
1	14000.246935	1998-05-29	5	0	0	0	0	1	0	904.274475
2	7000.246935	1998-05-29	5	1	0	0	0	0	0	311.630554
3	14000.246935	1998-05-29	5	1	0	0	0	0	0	640.888733
4	7000.246935	1998-05-29	5	0	1	0	0	0	0	404.192841
5	14000.246935	1998-05-29	5	0	1	0	0	0	0	784.635315
6	7000.246935	1998-05-29	5	0	0	1	0	0	0	535.914978
7	14000.246935	1998-05-29	5	0	0	1	0	0	0	1080.744629
8	7000.246935	1998-05-29	5	0	0	0	1	0	0	358.103729
9	14000.246935	1998-05-29	5	0	0	0	1	0	0	716.289368
10	7000.246935	1998-05-29	5	0	0	0	0	0	1	252.488205
11	14000.246935	2005-05-29	5	0	0	0	0	0	1	516.639893
12	14000.246935	2005-09-29	9	0	0	0	0	0	1	553.732300
13	14000.246935	2005-12-29	12	0	0	0	0	0	1	739.718140
14	14000.246935	2005-01-29	1	0	0	0	0	0	1	788.169067
15	14000.246935	2005-04-29	4	0	0	0	0	0	1	517.053528
16	90.311833	2023-04-26	4	0	0	0	0	0	1	3.302933
17	90.311833	2005-04-26	4	0	0	0	0	0	1	3.302933
18	90.311833	2005-04-26	4	0	0	0	1	0	0	10.692027
19	20.311833	2005-04-26	4	0	0	0	0	0	1	3.302933

Figura 19: Valores predichos con XGBoost.

Con la optimización realizada arroja resultados similares a los árboles de decisión, mantiene el problema de predecir ETAs erróneos o repetidos en valores muy bajos.

[33] (20 de enero de 2022) Tmonori Masui. All You Need to Know about Gradient Boosting Algorithm − Part 1. Regression. https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502

[34] ArcGIS Pro. Cómo funciona el algoritmo XGBoost. https://pro.arcgis.com/es/pro-app/latest/tool-reference/geoai/how-xgboost-works.htm#:~:text=XGBoost%20es%20un%20m%C3%A9todo%20de,(refuerzo%20de%20gradientes%20extremo).

5.4 Red Neuronal

Se trata de un modelo que trata de simular el funcionamiento del cerebro humano, interconectando, su principal funcionalidad es la clasificación y agrupación[35], al encontrar patrones de asociación entre los grupos de datos. Sin embargo, en este caso se utiliza para tratar de predecir un valor numérico en función de estas asociaciones. Se ha utilizado la implementación de sklearn *MLPRegressor*, nuevamente por ser la versión más básica de dicho modelo orientado a regresión, de modo que sirva para comprender como se comporta esta familia de algoritmos ante nuestro problema. En la Figura 20 observamos que arroja valores extremadamente grandes en los casos de poca distancia.

	distance	Date	month	Cloudy	Foggy	Hurricane	Rain	Storm	Sunny	Predicted eta
0	7000.246935	1998-05-29	5	0	0	0	0	1	0	597.734805
1	14000.246935	1998-05-29	5	0	0	0	0	1	0	1112.831629
2	7000.246935	1998-05-29	5	1	0	0	0	0	0	271.577402
3	14000.246935	1998-05-29	5	1	0	0	0	0	0	786.674226
4	7000.246935	1998-05-29	5	0	1	0	0	0	0	479.051107
5	14000.246935	1998-05-29	5	0	1	0	0	0	0	994.147931
6	7000.246935	1998-05-29	5	0	0	1	0	0	0	761.238321
7	14000.246935	1998-05-29	5	0	0	1	0	0	0	1276.335145
8	7000.246935	1998-05-29	5	0	0	0	1	0	0	382.507333
9	14000.246935	1998-05-29	5	0	0	0	1	0	0	897.604157
10	7000.246935	1998-05-29	5	0	0	0	0	0	1	185.267196
11	14000.246935	2005-05-29	5	0	0	0	0	0	1	598.654293
12	14000.246935	2005-09-29	9	0	0	0	0	0	1	487.789049
13	14000.246935	2005-12-29	12	0	0	0	0	0	1	404.640117
14	14000.246935	2005-01-29	1	0	0	0	0	0	1	709.519536
15	14000.246935	2005-04-29	4	0	0	0	0	0	1	626.370604
16	90.311833	2023-04-26	4	0	0	0	0	0	1	185.267196
17	90.311833	2005-04-26	4	0	0	0	0	0	1	185.267196
18	90.311833	2005-04-26	4	0	0	0	1	0	0	185.267196
19	20.311833	2005-04-26	4	0	0	0	0	0	1	185.267196

Figura 20: Valores predichos con red neuronal.

6. Evaluación

La metodología de división utilizada para el entrenamiento ha sido *division train-test* la cual se basa en dividir el conjunto de datos disponibles en dos partes, cuya suma componga el 100% del *dataset*, una de las partes para entrenamiento y la otra para validar los resultados. Se ha escogido esta frente a *cross-validation* por su menor requerimiento computacional, ya que la validación cruzada conlleva más tiempo y uso de recursos de cómputo, pero al poseer un *dataset* de entrenamiento tan grande no aporta una mejora suficientemente sustancial para aceptar esa pérdida de rendimiento. La división realizada ha sido 80/20, un 80% de los datos se usa para entrenamiento y un 20% para validación. Esta división se basa en el principio de Pareto, según el cual, de manera general un 80% de las consecuencias provienen de un 20% de las causas.[36]

[35] (17 de agosto de 2021) IBM. El modelo de redes neuronales. https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model

[36] (31 de enero de 2020) towardsdatascience. The 80/20 Split Intuition and an Alternative Split Method. https://towardsdatascience.com/finally-why-we-use-an-80-20-split-for-training-and-test-data-plus-an-alternative-method-oh-yes-edc77e96295d

A continuación se pasa a explicar las métricas utilizadas para la comparativa de los diferentes algoritmos predictivos.

R^2: Llamado coeficiente de determinación evalúa el ajuste de un modelo de regresión a los datos reales, se mide en una escala de 0 a 1, 1 sería un modelo cuyas predicciones se ajustan totalmente a la realidad, mientras que un valor de 0 indica que el modelo no tiene ningún valor predictivo.[37] Se calcula como la covarianza de los dos conjuntos de datos entre la multiplicación de la varianza de cada uno. En la Figura 21 visualizamos esta fórmula.

$$R^2 = \frac{\sigma_{XY}^2}{\sigma_X^2 \sigma_Y^2} = \rho^2$$

Figura 21: Fórmula del coeficiente de determinación. [38]

Conviene que este valor sea lo más alto posible.

RMSE: Conocido como raíz cuadrada del error cuadrático medio mide el error que hay entre dos conjuntos de datos. En la Figura 22, tenemos la fórmula.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (P_i - O_i)^2}{n}}$$

Figura 22: Fórmula RMSE.[39]

En términos simples se trata de la media de las diferencias entre los valores reales y los valores predichos. Se obtiene calculando la raíz del MSE que es el error cuadrático medio, (diferencia de los valores al cuadrado), por tanto es más intuitivo al darse en la escala de los propios valores predichos. Respecto a simplemente obtener la diferencia de los valores predichos y los reales, el RMSE es mejor, puesto que tiene en cuenta la dirección de los errores, iguala el peso de los errores positivos y negativos.

Es conveniente que el valor sea lo más bajo posible.

Tiempo Ejecución: Se trata del tiempo que tarda el algoritmo en obtener las predicciones para todo el conjunto de datos dado. Obviamente se busca que sea lo más rápido posible.

En primer lugar, comprobamos el funcionamiento de los algoritmos para calcular el ETA directamente.

---

[37] (3 de enero de 2023) IBM. R^2. https://www.ibm.com/docs/es/cognos-analytics/11.1.0?topic=terms-r2

[38] https://es.wikipedia.org/wiki/Coeficiente_de_determinaci%C3%B3n

[39] (22 de mayo de 2018) Gabri. ArcGeek. https://acolita.com/que-es-el-error-cuadratico-medio-rmse/

Algoritmo	R^2	RMSE	Tiempo Ejecución
Reg. Lineal Multiple	0.78	268.25	0.021
Red Neuronal	0.776	270.66	0.052
XGBoost	0.89	189.83	0.134
Árbol de Decisión	0.949	128.73	0.163

Nos interesa que el R2 sea más alto, pues eso implica una mayor precisión, del modelo, también que el RMSE sea bajo, teniendo en cuenta que se trata de la desviación respecto a los valores reales del ETA, por tanto, debe ser bajo en relación con la escala de unidades con la que estamos trabajando.

El tiempo de ejecución al ser tan bajo no supone un gran problema en ningún caso, sin embargo es interesante ver la proporción de tiempo que tarda cada uno respecto al resto.

En esta comparación, podemos ver que la Regresión Lineal Múltiple y la Red Neuronal tienen unos valores muy similares en los dos indicadores de precisión, sin embargo, la Red Neuronal es 2.5 veces más lenta para este caso. Por ello este segundo modelo quedaría descartado como opción final. Por otro lado, XGBoost y el modelo de Árboles de Decisión son mucho más precisos, pero tardan entre 10 y 15 veces más que la Regresión. Por este motivo parece que el modelo de Árboles de Decisión parece el mejor actualmente, pues quita el problema de los valores negativos y es más preciso. Es importante apuntar que XGBoost podría mejorarse con un mejor *tuning* sin embargo, esta es la mejor optimización que se ha podido lograr en este proyecto.

Lo idóneo sería por tanto conseguir mejorar la precisión de la Regresión lineal a valores cercanos a los Árboles de Decisión, para tener un algoritmo rápido y preciso, por ello, se valora la alternativa de calcular el ETA/km, el cual quita ambos problemas (ETAs negativos y ETAs repetidos).

Se utiliza el ETA/Km como métrica en lugar del ETA de cada 10 km u otra distancia, porque es una métrica fácilmente interpretable por el ser humano y que sale directamente como operación de otros dos campos del set de datos, además, posteriormente se debe multiplicar este valor nuevamente por los Km una vez predicho el valor para cada tramo, por lo que se ahorra realizar un mayor número de modificaciones que posteriormente deberían ser corregidas y son innecesarias.

Algoritmo	R^2	RMSE	Tiempo Ejecución
Reg. Lineal Multiple	0.468	0.019	0.013
Red Neuronal	0.275	0.022	0.045
XGBoost	0.665	0.015	0.141
Árbol de Decisión	0.848	0.01	0.212

En este caso vemos que el R^2 es peor en todos los casos que al calcular el ETA directamente. Por su parte el RMSE no se puede comparar porque estamos trabajando en un rango mucho menor de valores.

El uso de una Red Neuronal queda descartado al tener peor valor en los 3 indicadores que la Regresión, por su parte pasa algo similar entre XGBoost y los Árboles de Decisión, pues el segundo el más preciso, pero un 33% más lento. Sin embargo, la precisión de XGBoost es 0.2 puntos más baja que la del algoritmo de Árboles de Decisión, y sus predicciones tienen un 33% de mayor desviación.

Se considera que al ser un sistema donde la predicción no es el mayor cuello de botella para el tiempo de ejecución, es prioritario penalizar la ejecución y mejorar la precisión.

Entre la Regresión Lineal y los Árboles de Decisión, la elección es clara, el R^2 es inferior a 0.5 en el primer caso, lo que significa que está más cerca de no ajustar los datos al modelo en absoluto que de hacerlo a la perfección. Si bien el modelo de Regresión es prácticamente 20 veces más rápido, pero es preferible tener un sistema más lento pero que haga predicciones lógicas a un sistema que no funcione bien y apenas tenga utilidad pero sea muy rápido.

7. Estadías

El siguiente paso era realizar el mismo proceso para generar un modelo que sea capaz de predecir el tiempo de estadía, tiempo durante el que se realiza el paso de las mercancías de un medio de transporte a otro sin realizar desplazamiento.

Una estadía, es el tiempo que un transporte de mercancías se encuentra estacionado en un puerto o estación para realizar el paso de los productos transportados a otro medio de transporte. Esto es fundamental tenerlo en cuenta para tramos en los que se pase de transporte ferroviario a marítimo o al opuesto. Pues dependiendo de la cantidad de mercancías que haya que transferir, el tiempo puede aumentar enormemente, por ello es necesario tenerlo en cuenta en el cálculo del ETA de los tramos. Hay una serie de factores que afectan.

- **Tiempo de preparación:** Aunque solo se descargase un contenedor con un peso pequeño, se debe preparar la infraestructura, el amarre del barco o estacionamiento del tren, los operarios… Que lleva un tiempo de preparación fijo de unos 30 minutos aproximadamente.
- **El transporte que trae las mercancías**: Se tardará más en pasar las mercancías de un tren a un barco porque la ordenación de los contenedores en el barco es mucho más compleja que en un tren.
- **El peso que se transporta:** No es lo mismo transferir 50000 toneladas que 10, afectará al tiempo que se tarde en transferir las mercancías y hay que tenerlo en cuenta.

Teniendo estos 3 factores en cuenta se ha generado un *dataset* sintético que pueda servir de histórico teórico para desarrollar el sistema.

Al tratarse de un problema muy similar al previo, se han probado directamente los dos algoritmos predictivos que funcionaron bien en ese caso, para ver cuál de los dos funciona mejor en esta situación.

El procedimiento para crear y analizar el modelo predictivo del ETA de las estadías ha sido el mismo que para el ETA del transporte de mercancías. Sin embargo, en este caso ha sido XGBoost el que ha obtenido mejores resultados. Se ha probado con Árboles de Decisión potenciados por Regresión y con XGBoost, también basado en árboles, puesto que la Regresión Lineal no es útil en este caso, ya que al incorporar un tiempo de base. independientemente del peso, el aumento de tiempo según el peso no va a ser perfectamente lineal, pues nunca debería ser inferior a dicho umbral.

Para aumentar la precisión, se predice el tiempo en minutos y posteriormente se convierte en horas.

Algoritmo	**R^2**	**RMSE**
XGBoost	0.888	344.92
Árbol de Decisión	0.856	391.42

Como se observa, tanto el R^2 como el RMSE son mejores con XGBoost.

8. Despliegue

8.1 Obtención de clima

Una vez realizado todo el proceso de minería de datos y entrenamiento debemos pasar a desarrollar el sistema completo que lo usará.

El siguiente paso del sistema es obtener el clima en tiempo real en todas las coordenadas que se utilizan como *checkpoints* para generar una ruta posteriormente utilizando el modelo desarrollado.

Para ello se ha utilizado una API de uso público y gratuito llamada *Weather Forecast API*[40]. Este servicio es propiedad de *Open-Meteo*.

El funcionamiento de nuestro sistema es realizar una petición individual con cada una de las coordenadas del set al servicio web, solicitando el código de clima para ese punto en tiempo real. La duración del conjunto de todas estas llamadas para el set de coordenadas actual es de unos 5 minutos, por este motivo, sería conveniente cuando se plantee desplegar en un entorno de producción, utilizar un orquestador como Jenkins para lanzar este script cada cierto periodo de tiempo fijado y actualizar el clima del *dataset*, en lugar de lanzarlo a cada ejecución. De este modo, se utilizaría un clima con cierto desfase, cuyo máximo dependerá del tiempo de actualización, pero se reducirá en 5 minutos el cálculo de la ruta.

Los códigos de interpretación de clima difieren de los utilizados por nuestro sistema, sin embargo, todos están contenidos en él, por ello se realiza una correspondencia a la

[40] https://open-meteo.com/en/docs

categoría climática más parecida de nuestro sistema cada vez que se recibe una respuesta, la correspondencia se observa en la Figura 23.

Code	Description
0	Clear sky
1, 2, 3	Mainly clear, partly cloudy, and overcast
45, 48	Fog and depositing rime fog
51, 53, 55	Drizzle: Light, moderate, and dense intensity
56, 57	Freezing Drizzle: Light and dense intensity
61, 63, 65	Rain: Slight, moderate and heavy intensity
66, 67	Freezing Rain: Light and heavy intensity
71, 73, 75	Snow fall: Slight, moderate, and heavy intensity
77	Snow grains
80, 81, 82	Rain showers: Slight, moderate, and violent
85, 86	Snow showers slight and heavy
95 *	Thunderstorm: Slight or moderate
96, 99 *	Thunderstorm with slight and heavy hail

```
Sunny = [0,1]
Cloudy = [2, 3]
Rain = [51,53,55,56,57,61,63,66,67,80]
Foggy = [45, 48]
Storm = [65, 81, 82, 95, 96]
Hurricane = [99]
Snow = [71,73,75,77, 85, 86]
```

Figura 23: Código de clima y correspondencia con el sistema desarrollado.

Si se recibe un código no contenido en nuestro sistema, se almacenará como *No Data*.

En definitiva, se ha realizado un proceso ETL (Extracción, transformación y carga) para obtener los datos del clima y utilizarlos en nuestro sistema, pues se han extraído los datos de *open-meteo* usando su API, se han transformado los datos para adaptarlos a nuestro sistema, y posiblemente se han cargado en nuestro conjunto de coordenadas para utilizarlo en el algoritmo predictivo.

8.2 Obtención de tramos transitables

El primer paso una vez cargadas las coordenadas que pueden ser transitables, es obtener la conexiones o tramos entre ellas. Al no haber ninguna restricción al respecto, la primera aproximación es obtener la matriz de conexión de todos los puntos con todos, sin embargo, esto supone un problema, pues la forma más rápida de ir de un punto X a cualquier punto Y, será la conexión directa entre estos dos puntos, la cual será una línea recta entre ambos. Por ello, es necesario reducir el volumen de conexiones o aristas del grafo, usando este grafo completamente conectado como punto de partida. (Grafo completo como el que vemos en la Figura 24).

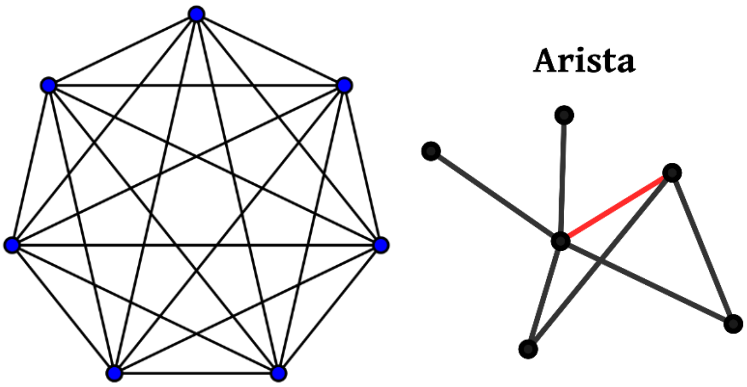

Figura 24: Grafo completo[41] (Izquierda), Arista de un grafo[42] (Derecha)

Para obtener la distancia entre cada par de coordenadas terrestres en este caso no recurrimos a la hipotenusa del triángulo, pues en este caso nos interesa la situación geográfica real y no solo la distancia, al trabajar con coordenadas geográficas decimales existe una problemática a resolver y esta es la distancia entre los dos extremos del plano, pues no tiene en cuenta que la tierra es una esfera, por tanto los dos puntos más lejanos del mapa, que en realidad se encuentran cercanos se calcularán como puntos totalmente distantes como se observa en la Figura 25, mientras que lo que buscamos es que se obtenga la distancia real en el globo terráqueo como en la Figura 26.

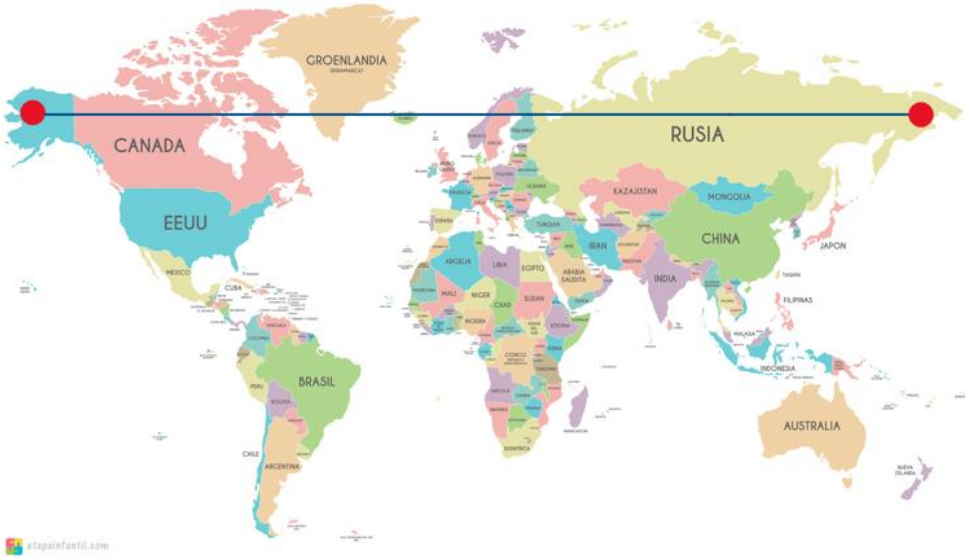

Figura 25: Distancia entre dos coordenadas de un mapa en un plano no esférico. Fuente Etapa Infantil.[43]

[41] (14 de enero de 2006) David Benbennick. https://es.wikipedia.org/wiki/Grafo_completo#/media/Archivo:Complete_graph_K7.svg
[42] (2 de abil de 2011) Pedro Sánchez. https://commons.wikimedia.org/wiki/File:Grafo_-_Arista.svg
[43] Etapa Infantil. https://www.etapainfantil.com/mapamundi-para-imprimir

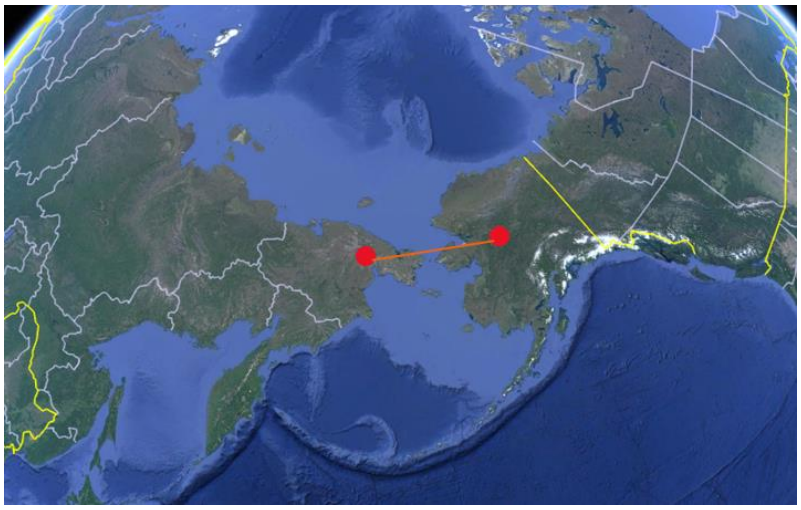

Figura 26: Distancia real entre esos dos puntos. Fuente Google Earth Pro.[44]

La solución que existe ante este conocido problema de cálculo es utilizar la fórmula de Harvesine, cuya visualización se encuentra en la Figura 27, esta fórmula obtiene la distancia más corta entre dos puntos de una esfera dada su longitud y latitud.

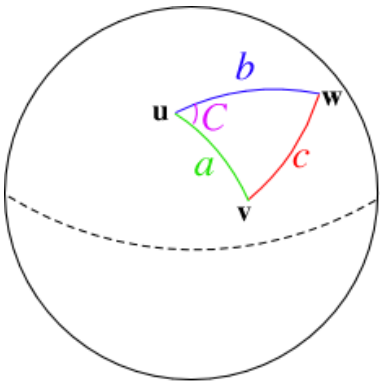

Figura 27: Visualización parámetros Harvesine. Fuente "Del Karukinká"[45]

El método de la Harvesine se compone de varios pasos, el primero es transformar las coordenadas decimales en radianes, de este modo convertimos los puntos en un mapa a los puntos en una esfera, posteriormente obtenemos la diferencia entre las latitudes y longitudes de ambos puntos, componiendo de este modo "a" y "b". A continuación, calculamos la mitad del cuadrado de la distancia entre los dos puntos y la distancia del ángulo en radianes, con esto obtenemos "c".

La distancia final se calcula como la multiplicación de esta distancia ("c") por el diámetro terrestre.

En la Figura 28 observamos esta fórmula para una esfera genérica.

[44] Google Earth PRO. https://www.google.com/intl/es/earth/
[45] (28 de agosto de 2017) https://delkarukinka.wordpress.com/2017/08/28/calcular-distancia-entre-dos-coordenadas-en-la-tierra/

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Figura 28: Fórmula completa de Harvesine. [46]

Una primera forma de hacer la reducción es buscar el árbol de expansión mínimo o *Minimal Spanning Tree*, el cual conforma la ruta de peso mínimo que atraviesa todos los nodos del grafo, sin embargo esto trae un problema, solo habría un modo de ir de un punto a otro, lo cual en algunos casos puede ser la ruta más rápida, pero en otros puede suponer tener que rodear completamente el destino o desviarte para posteriormente llegar a él, como observamos en la Figura 29, donde el *Spanning Tree* solo permitiría atravesar la ruta de peso 12 para llegar al destino, mientras que la ruta de peso 9 es más rápida.

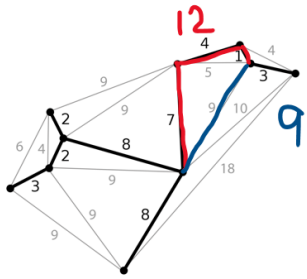

Figura 29: Visualización de *Spanning Tree*[47] donde la ruta sería mucho más corta por otra conexión.

El objetivo por tanto en este punto es formar un grafo de conexiones entre las coordenadas conexo, de modo que desde cualquier punto se pueda llegar a cualquier otro, reduciendo la cantidad de conexiones o aristas para aumentar la precisión o cantidad de conexiones que haya que atravesar para llegar a un punto, pero sin dejar demasiadas como para penalizar la velocidad del sistema, o demasiadas pocas para que las rutas que genere impliquen dar vueltas absurdas.

La siguiente aproximación que se realiza, es obtener la arista con mayor peso (distancia) dentro del *Minimal Spanning Tree* y eliminar todas aquellas conexiones con mayor distancia. Sin embargo, al comprobar esto prácticamente, observamos que, a pesar de quedar el grafo conexo como es lógico, pues garantiza que al menos el *Minimum Spanning Tree* va a estar dentro del grafo resultante, la distancia es tan grande que es posible llegar a cualquier punto en apenas 2 o 3 saltos y el sistema pierde mucha precisión.

---

[46] (16 de junio de 2017) Daniil Sydorenko. Harvesine formula.
https://github.com/DaniilSydorenko/haversine-geolocation/issues/1
[47] (31 de diciembre de 2005).
https://commons.wikimedia.org/wiki/File:Minimum_spanning_tree.svg

La solución final y que mejor funciona que se haya ideado es buscar de partida el punto más lejano a todos los demás, el punto con la mayor distancia a su punto más cercano, de este modo nos aseguramos de que todos los nodos estarán al menos conectados con otro y que en el caso del más lejano será por su camino más corto únicamente. Sin embargo, esto resulta en 2 grafos conexos, pero inconexos entre sí. Esto se debe a que si bien todos los nodos están conectados con otro, eso no implica que todos lo estén conectados entre sí, pues podría darse el caso de que haya 2 nodos con una distancia pequeña entre ambos, pero que para conectarse con un tercero esta distancia sea mucho mayor y escape del rango donde se ha hecho el corte. Llegados a este punto, se ha comprobado empíricamente cuanto se debía aumentar esta distancia para obtener la distancia mínima a la que el grafo sigue conexo.

La distancia resultante es 1400 kilómetros aproximadamente, lo cual implica que en el peor de los casos la distancia entre dos saltos será de esa distancia. Es una buena solución para un sistema que no pretende dar indicaciones en cada calle como sería un *Google Maps*, si no que se utilizará para rutas internacionales, indicando la trayectoria global por la que debería realizarse un transporte de mercancías o para conocer si atravesar un continente en tren es más rápido que por la costa, por ejemplo.

Finalmente resultan 20568 aristas, que frente a las 1525223 de inicio supone una reducción del 98.65% de las aristas, lo cual hace al sistema mucho más preciso y rápido, además, el grafo, el cual se visualiza en la Figura 30, se mantiene conexo, tal y como buscábamos.

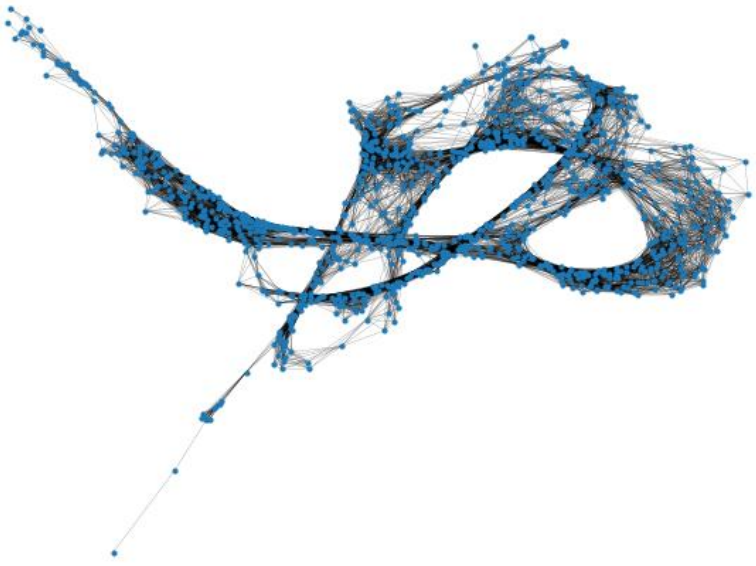

Figura 30: Grafo resultante

8.3 Cálculo de pesos

Para que el sistema pueda saber cuál es el coste de atravesar cada tramo y por tanto escoger la ruta óptima, se utiliza un grafo con pesos, cuyo peso hasta este punto era únicamente la distancia, sin embargo, queremos tener en cuenta una predicción de tiempo en función del clima y la fecha actual. Para ello, calculamos el mes actual, y usando el clima previamente obtenido y los modelos predictivos diseñados, realizamos la predicción del ETA para cada tramo, según sea un tramo ferroviario o marítimo.

Para los tramos híbridos, se utiliza el modelo predictivo previamente desarrollado, se indica el peso que se transporta y predice los ETAs del tiempo que tomará el intercambio de las mercancías, el cual se suma al ETA previamente calculado.

El sistema recibe las coordenadas de inicio y de final del trayecto. Y a partir de ellas obtiene el punto más cercano de nuestro *dataset* de coordenadas a cada una de ellas, el sistema calculará la ruta entre estos dos puntos.

Si bien se está omitiendo el tiempo que se tarda en realizar el transporte de las mercancías desde la coordenada dada hasta el punto más cercano, esto se ha tenido en cuenta en una primera implementación, sin embargo, se ha descartado posteriormente al no tener sentido pues se desconoce el medio de transporte que se utilizará. Al estar mapeado el conjunto de rutas ferroviarias y de coordenadas marítimas y puertos, una localización que diste mucho de cualquiera de los puntos más cercanos del *dataset* deberá usar camiones u otro medio de transporte como punto de partida, lo cual no se soporta en este sistema. Sin embargo, de cara a una futura mejora por parte de la empresa se podría tener en cuenta.

8.4 Cálculo de la ruta óptima y ETA

Una vez realizada toda esta preparación, la obtención de la ruta óptima es rápida, se utiliza el algoritmo de Dijkstra.

Este algoritmo busca iterativamente los caminos más cortos para llegar a los nodos vecinos de los nodos descubiertos, de este modo se garantiza que una vez encontrado un camino al nodo destino, este será el camino más corto posible como vemos en la Figura 31, ya que si la ruta mínima para llegar de A a C atraviesa B, esta contendrá el camino mínimo para llegar de A a B igualmente, esto se conoce como principio de optimalidad.[48]

[48] Gloria Sánchez Torrubia y Víctor M. Lozano Terrazas. Universidad Politécnica de Madrid. Algoritmo de Dijkstra Un Tutorial Interactivo.
http://bioinfo.uib.es/~joemiro/aenui/procJenui/ProcWeb/actas2001/saalg223.pdf

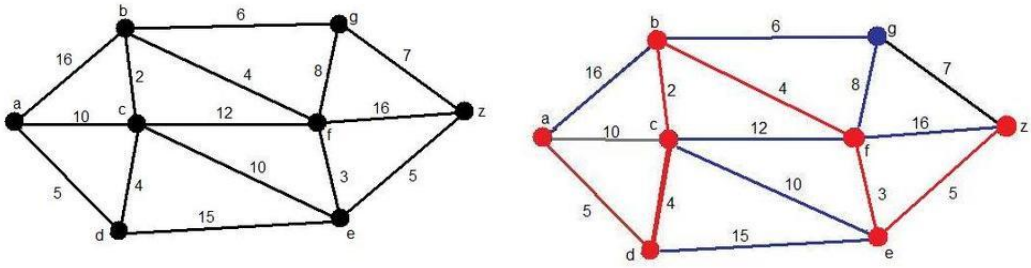

Figura 31: Algoritmo Dijkstra para ir del punto A al punto B. (Izquierda inicio, derecha final) [49]

Indicando el nodo origen y el nodo final, y pasando el grafo generado, el algoritmo de Dijkstra nos devuelve los nodos que componen la ruta óptima. Si sumamos el peso de las aristas que unen estos nodos, obtenemos el tiempo total que predice el sistema que se tardará en realizar el trayecto.

Una vez obtenida la ruta se exporta a un CSV con un formato apto para ser visualizado por una herramienta externa.

8.5 Visualización de las rutas

Para la visualización de las rutas se ha utilizado la herramienta software Power Bi con el complemento *Route map*, desarrollado por Weiwei Cui, el cual permite visualizar un conjunto de coordenadas en un mapa, unirlos y segmentarlos en función de variables o darles diferentes colores en función de otra variable.

Un ejemplo de ruta sería desde Taymyrsky Dolgano-Nenetsky District, Krai de Krasnoyarsk, Rusia coordenadas (67.66845481747058, 88.49166348406456) hasta El Cuy, Río Negro, Argentina, coordenadas (-40.25902471612411, -68.76538856544546), éstas pueden ser obtenidas a través de *Google Maps* como en la Figura 32.

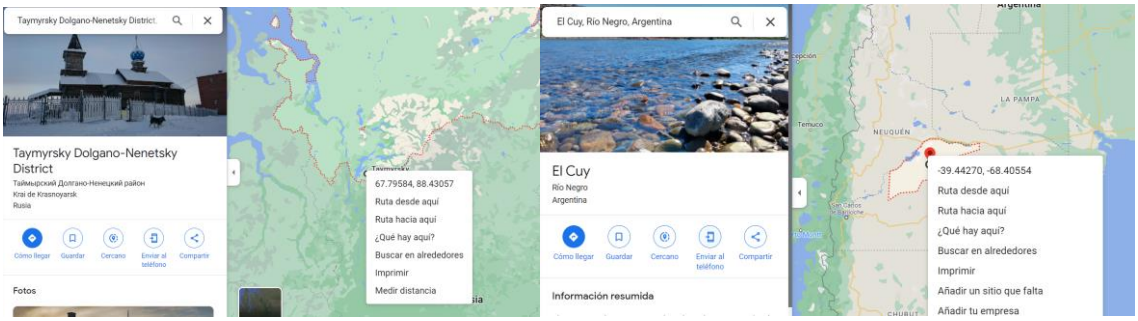

Figura 32: Origen (Izquierda) y llegada (Derecha) de la ruta en Google Maps.[50]

[49] (26 de mayo de 2009) Jipsy. https://commons.wikimedia.org/wiki/File:Dijkstrapaso8.jpg
[50] Google Maps. Enlace a la ruta. https://www.google.es/maps/dir/67.66845481747058,++88.49166348406456/El+Cuy,+R%C3%ADo+Negro,+Argentina/@-0.4614985,-51.2836918,3z/data=!4m12!4m11!1m3!2m2!1d88.4916635!2d67.6684548!1m5!1m1!1s0x9609002cb371de1d:0x237206c00e3d64e8!2m2!1d-68.3339275!2d-39.6854845!3e2

El sistema nos arroja el resultado que se muestra en la Figura 33:

```
[974, 859.0, 712.0, 932.0, 869.0, 531.0, 676.0, 501.0, 568.0, 147.0, 629.0, 641.0, 389.0, 1051.0, 950.0, 890.0]
596.6536610104439
```

Figura 33: Listado de nodos que se atraviesan en la ruta y ETA de la ruta en horas.

Habrá que atravesar los puntos correspondientes al listado superior, 16 coordenadas de referencia para construir la ruta. Tardará en realizarla 596.65 horas lo cual se corresponde a 24.86 días, una cifra adecuada si se va a llevar 10000 toneladas entre dos extremos el planeta.

La visualización de la ruta en Power BI se observa en la Figura 34.

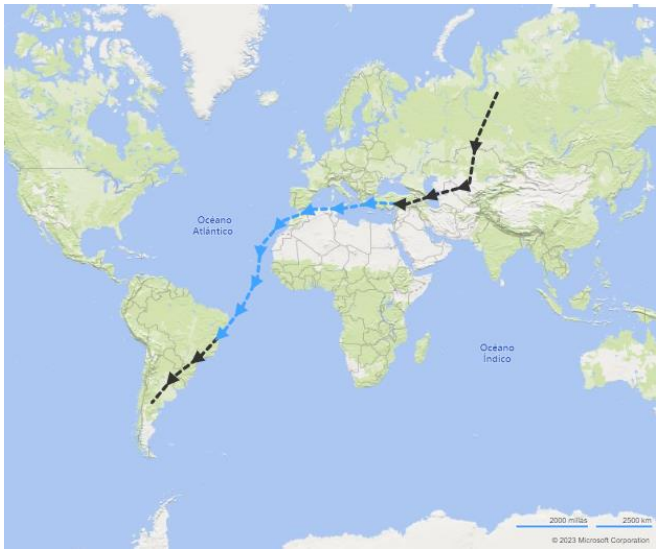

Figura 34: Visualización de la ruta en Power BI.[51]

Se ha repetido la misma ejecución al día siguiente para comprobar que la ruta obtenida si bien es la misma, el coste calculado es diferente debido a las condiciones actuales, observamos este cambio en la Figura 35.

```
[974, 859.0, 712.0, 932.0, 869.0, 531.0, 676.0, 501.0, 568.0, 147.0, 629.0, 641.0, 389.0, 1051.0, 950.0, 890.0]
675.1943296481918
```

Figura 35: Listado de nodos que se atraviesan en la ruta y ETA de la ruta en horas.

El ETA estimado ha pasado de 596.65 a 675.19 horas.

9. Video Demostración

En el siguiente enlace se puede visualizar una demo del sistema en funcionamiento:

https://youtu.be/4sP_KVzSNi4

[51] PowerBI. https://powerbi.microsoft.com/es-es/

10. Código en Github

En el siguiente proyecto de Github se puede ver el código realizado durante el desarrollo de este sistema:

https://github.com/albertoM02/MultimodalRoutesAndETA_TFG_AlbertoMoro

11. Conclusiones

Una vez terminado de implementar completamente el sistema hasta el alcance propuesto, se concluye que el sistema realizado satisface los objetivos propuestos, se trata de un sistema que funciona mejor cuanto mayor es la distancia entre el inicio y final de la ruta, hay que tener en cuenta que no se trata de un sistema de rutas tipo *Google Maps* que indica cada pocos metros el giro que se debe realizar o el camino a seguir, si no que se utiliza para obtener a grandes rasgos por donde se tardaría menos en realizar una ruta internacional, atravesando un país o bordeándolo por mar, un continente, ir de EEUU a China atravesando Europa o por el Océano Pacífico… Además hay que tener en cuenta que existen múltiples restricciones a nivel legislativo por las que un determinado buque no podrá utilizar un puerto, porque no tengan contrato con él si es privado, porque no haya disponibilidad… Además de tener que seguir la ruta marítima predefinida, por lo que no necesitan indicaciones cada km. El caso de uso principal que se plantea para este sistema es el del encargado de escoger la ruta que realizará un buque, lo use como línea base o directriz de partida, y desde este punto contacte con las estaciones, puertos y autoridades pertinentes para comprobar la disponibilidad de las estaciones marítimas y rutas ferroviarias que se correspondan con la ruta, y a partir de las restricciones que se impongan modifique la ruta.

Otras conclusiones obtenidas son los grandes requerimientos computacionales que requiere un sistema así, para obtener el clima en todo el mundo en las aproximadamente 1200 coordenadas transitables que se utilizan actualmente, tarda más de 5 minutos, para obtener la matriz de conexiones entre todas las coordenadas inicial otros 5 minutos, aumentando el tiempo exponencialmente con el tamaño en este caso, y la inmensa cantidad de coordenadas que se necesita para realizar un sistema de precisión similar a *Google Maps* o cualquier otro sistema de rutas de corta distancia.

Respecto al trabajo realizado en la empresa y el aprendizaje que me ha supuesto, en primer lugar, ha sido un gran reto para mí, puesto que al cursar la especialidad de *Software*, he tenido que realizar un gran aprendizaje sobre los procesos de ciencia de datos, los cuáles desconocía, pues nunca había trabajado con ellos. También el lenguaje de programación *Python* era prácticamente nuevo para mí, por lo que he tenido que familiarizarme tanto con su sintaxis como sus librerías y métodos más habituales. Por otro lado, he debido comprender, hasta donde me ha sido posible, las matemáticas y

razonamientos que hay detrás de los algoritmos empleados. En definitiva, se ha tratado de un aprendizaje constante a cada avance que debía realizar. Además, en la empresa he aprendido los procesos de negocio típicos, así como poner en práctica los procesos de desarrollo de software que había aprendido en el grado de Ingeniería Informática. Cada dos semanas, siempre que fuera posible, tenía una reunión con mi jefe, para mostrarle los avances que llevaba en el proyecto e indicarle lo que pensaba realizar las próximas dos semanas, esto se conoce como *Sprints*. También he aprendido a desarrollar reuniones, los medios de comunicación que se emplean para concretar estas reuniones o contactar a los demás empleados y las tecnologías que se utilizan para computar horas de trabajo o registrar las tareas realizadas.

Otro aspecto que resulta interesante es el haber realizado el proyecto en paralelo al curso de especialización en ciberseguridad de la universidad, puesto que el poseer conocimiento de otros ámbitos siempre arroja nuevos puntos de vista o enfoques a los trabajos que se realizan, y este ha sido el caso. Información que en otro caso hubiera pasado desapercibida, en este caso ha cobrado importancia y ha sido sujeto de análisis. Un ejemplo claro es lo relevante que es preservar la información e inteligencia de la empresa que sea vulnerable o privada, lo cual es complejo cuando se quiere exponer el trabajo realizado a una entidad externa a la empresa o de manera pública como es el caso, para esto existen múltiples técnicas y procedimientos que permiten realizar el trabajo de la mejor manera posible sin exponer información sensible. Una de estas técnicas es precisamente la generación de *datasets* sintéticos, lo cual permite crear datos ficticios equivalentes a los reales, de manera que tengan la misma calidad pero no expongan información real. Otro aspecto crucial es conocer los términos detrás del uso de herramientas externas, pues en varios casos se determina que los datos que se procesen con dicha herramienta podrán ser utilizados y almacenados por la empresa propietaria de esa tecnología, con lo cual se debe buscar una alternativa o bien ser muy cuidadoso con los datos e información que se procesan con la herramienta.

12. Trabajo futuro

De cara a la futura evolución de este proyecto, la directriz principal sería añadir transporte por carretera al sistema de rutas con las restricciones que eso implica, no se ha valorado para el alcance de este proyecto, puesto que aparte de tener unas restricciones muy diferentes al transporte marítimo y ferroviario, está actualmente siendo desarrollado por otro empleado, de modo que en el futuro se plantea integrar ambos sistemas para poseer un sistema completo.

También sería interesante estudiar las opciones de despliegue del sistema, pues al requerir tanto tiempo en obtener el clima en todo el mundo, se podría gestionar el flujo de datos con un orquestador tipo “Jenkins” que ejecute cada hora o cada cierto tiempo determinado esa parte del código y genere el grafo actualizado de pesos que el algoritmo de Dijkstra utilizará cuando se le solicite una ruta. De este modo una petición al sistema

tardará en ejecutarse lo que tarde el algoritmo de Dijkstra (que está en torno a 0.1 – 0.2 décimas con este volumen de datos) en lugar de los 10 minutos del sistema completo.

Otra opción de evolución en el futuro sería aumentar la precisión de las rutas y desplegarlo en un sistema de alto computo, como puede ser un *cluster*.

Por último, también sería interesante exponerlo como un servicio web al que se le introduzcan las coordenadas de origen y destino y el peso a transportar y devuelva la ruta.

13. Bibliografía

[1] Silaparasetty, N. (2020). Introduction to Jupyter Notebook. In: Machine Learning Concepts with Python and the Jupyter Notebook Environment. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-5967-2_6

[2] Ortega-Arranz, H., Llanos, D.R., Gonzalez-Escribano, A. (2015). Classical Algorithms. In: The Shortest-Path Problem. Synthesis Lectures on Theoretical Computer Science. Springer, Cham. https://doi.org/10.1007/978-3-031-02574-7_3

[3] Alexei Novikov. (15 de mayo de 2019). Creating sea routes from the sea of AIS data. https://towardsdatascience.com/creating-sea-routes-from-the-sea-of-ais-data-30bc68d8530e

[4] Tenkanen, H., Toivonen, T. Longitudinal spatial dataset on travel times and distances by different travel modes in Helsinki Region. Sci Data 7, 77 (2020). https://doi.org/10.1038/s41597-020-0413-y

[5] Huang, H., Pouls, M., Meyer, A., Pauly, M. (2020). Travel Time Prediction Using Tree-Based Ensembles. In: Lalla-Ruiz, E., Mes, M., Voß, S. (eds) Computational Logistics. ICCL 2020. Lecture Notes in Computer Science(), vol 12433. Springer, Cham. https://doi.org/10.1007/978-3-030-59747-4_27

[6] Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, pp. 3146–3154 (2017)

[7] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[8] Roell, M., Viarouge, A., Hilscher, E. et al. Evidence for a visuospatial bias in decimal number comparison in adolescents and in adults. Sci Rep 9, 14770 (2019). https://doi.org/10.1038/s41598-019-51392-6

[9] Gutierrez. J, Tirnauca. C, García. D. Aplicación de técnicas de ciencia de datos para mejorar la gestión de flotas de transporte. (2022) http://hdl.handle.net/10902/25814