Universidad de Valladolid

Escuela de Ingeniería Informática de Segovia

Grado en Ingeniería Informática de Servicios y Aplicaciones

# Predicción de los tiempos de llegada de los vuelos mediante aprendizaje automático

*Alumno*: Beatriz Arévalo Llorente

*Tutores*: Anibal Bregón Bregón
Miguel Ángel Martinez Prieto

# Agradecimientos

*Quería agradecer a mis padres el apoyo que me han dado en todo momento, ya que sin ellos no hubiera sido posible. Así como a mis tutores Miguel Ángel y Aníbal por haberme propuesto este proyecto con el que he adquirido conocimientos en nuevas tecnologías.*

# Resumen

La presencia de la inteligencia artificial ha crecido notoriamente estos últimos años, convirtiéndose en un aspecto presente en muchas de nuestras interacciones con el mundo tecnológico. Para su desarrollo necesitaremos acceder y procesar grandes cantidades de datos (Big Data), lo que nos permitirá sacar patrones y poder predecir hechos futuros. En nuestro caso haremos uso de técnicas de inteligencia artificial con el objetivo establecer predicciones acerca de los tiempos de aterrizaje de los vuelos.

En nuestro proyecto necesitaremos tratar la información recogida tanto por los planes de vuelo, como por el sistema Automático Dependiente de Vigilancia - Difusión (ADS-B). Esta última es una tecnología de vigilancia cooperativa utilizada por las aeronaves para determinar su posición mediante la navegación por satélite, permitiéndonos conocer datos referentes al estado del vuelo a través de un transmisor a bordo de la aeronave. Tendremos por un lado los datos relacionados con el vuelo (identificador, llegada prevista, tipo de avión, aeropuerto de salida y llegada, etc) y por otro los mensajes emitidos por el avión durante el vuelo, que nos proporcionaran información sobre el despegue y aterrizaje reales.

El tratamiento de los datos descritos anteriormente se hará utilizando Apache Spark, que es una plataforma desarrollada para agilizar el tratamiento de grandes cantidades de datos, más concretamente centrándonos en el uso de MLlib. Esta librería nos da la posibilidad de utilizar algoritmos de Machine Learning, con los que proponer y probar modelos de aprendizaje automático basados en los datos de los vuelos. Lo que nos dará como resultado una serie de predicciones de los tiempos de llegada de las aeronaves.

Por último, será necesario el desarrollo de una herramienta para la visualización y tratamiento de los datos, así como para la ejecución de los distintos algoritmos y posterior visualización de los resultados obtenidos. Esto implicará la creación de un dashboard utilizando soluciones basadas en Python.

**Palabras clave:** ADB-S, Spark, MLlib, Machine Learning, Python.

# Abstract

The presence of artificial intelligence has grown notoriously in recent years, becoming a present aspect in many of our interactions with the technological world. For its development we will need to access and process large amounts of data (Big Data), that will allow us to obtain patterns and predict future events. In this case, we will use artificial intelligence techniques in order to establish predictions about flight landings times.

In this project we will process the information collected by both the flight plans and the Automatic Dependent Surveillance - Broadcasting system (ADS-B). This latest is a cooperative surveillance technology used by aircrafts to determine their position by satellite navigation, allowing us to know data regarding the state of the flight through a transmitter aboard the aircraft. We will have, on the one hand, the data related to the flight (identifier, expected arrival, type of plane, airport of departure and arrival, etc) and on the other hand, the messages emitted by the plane during the flight, which will provide us with information about the real take-off and landing.

The data described above will be processed using Apache Spark, a platform developed to speed up the treatment of large amounts of data, more concretely focusing on the use of MLlib. This library gives us the possibility of using Machine Learning algorithms, with which to propose and test learning models automatic based on the flights data. This will give us a series of predictions of the arrival times of the aircraft.

Finally, it will be necessary to develop a tool for the visualization and treatment of the data, as well as for the execution of the different algorithms and subsequent visualization of the results obtained. This will involve creating a dashboard using solutions based on Python.

**Keywords:** ADB-S, Spark, MLlib, Machine Learning, Python.

# Índice general

# Índice de figuras

# Índice de tablas

# Capítulo 1

# Introducción

La inteligencia artificial (IA) está presente en multitud de aspectos en nuestras vidas, ya que es utilizada por numerosos sectores. Anteriormente, la capacidad de cómputo que existía era notablemente menor, generando resultados muy pobres en los problemas a los que se aplicaba, lo que provocó un cierto desinterés. Esto ha cambiado en la actualidad, ya que se dispone de dispositivos con mucha mayor capacidad de cómputo lo que ha impulsado al desarrollo de técnicas más avanzadas de inteligencia artificial. Un claro ejemplo está en los móviles con utilidades como la detección facial, el reconocimiento de la huella dactilar o la sugerencia de palabras en el teclado.

La necesidad de dar sentido a los numerosos datos de los que se dispone ha hecho que la combinación del Big Data y la IA generen un gran interés, dando pie a que cada vez haya más personas trabajando en estos campos y se desarrollen más herramientas con las que poder implementar técnicas relacionadas con la IA. Se prevé que en el año 2025 el 85 % de la interacción con los clientes sea gestionada con IA y que su mercado puede llegar a representar 127.000 millones de dólares [1].

Se benefician del uso de la IA sectores como el de las finanzas, para organizar operaciones, invertir en acciones y administrar propiedades [2]; el de la medicina, para la organización del personal o la asignación de camas, llegando a utilizar redes neuronales artificiales como sistemas de apoyo para decisiones clínicas en el diagnóstico médico [2]; en la industria pesada, asignando puestos de trabajo peligrosos a robots en vez de a humanos [2]; en la aviación, utilizando sistemas de simulación de vuelo y toma de decisiones en combate [2].

La aparición de Internet ha sido determinante, proporcionándonos infinidad de datos que son procesados por diferentes herramientas de Big Data y que al ser interpretados por la IA cobran un determinado sentido. Para que nos hagamos una idea, cada persona generamos en un día seis megabytes de información, y según la Unión Europea se generan 1.700 billones de bytes por minuto, se estima que en los próximos 5 años esta cifra se duplicará [1]. Tenemos casos como el de Google o redes sociales como Facebook o Instagram que recopilan toda la información que les proporcionamos ya sea, mediante fotos, perfiles a los que accedemos o búsquedas por Internet, y aprovechan estos datos para sugerirnos nueva información relacionada con nuestros

gustos o intereses. Un ejemplo le tenemos en el Gmail, cuando hacemos una reserva ya sea de un vuelo u hotel, Google muestra información en otras aplicaciones como Google Maps o Google Calendar acerca de dicha reserva.

## 1.1. Motivación

El uso del avión como medio de transporte se ha convertido en algo habitual en la actualidad, llegando en el año 2018 a los 4.300 millones de pasajeros según informa la Organización de Aviación Civil Internacional (OACI) [3]. El transporte aéreo de mercancías también se ha incrementado un 3,5 % respecto al 2017 [4], donde se transportaron cerca de 213.500 millones de toneladas [6]. En 2018 se superó el récord de vuelos comerciales en un día llegando a 202.157, con picos de hasta 19.000 aviones volando al mismo tiempo [5].

Saber si un avión va a llegar con antelación o retraso puede ser muy útil a la hora de organizar la operativa de un aeropuerto, logrando optimizar los tiempos de espera a la hora de asignar una pista para el aterrizaje. La fluctuación de tiempo de llegada en un vuelo puede generar un retraso en cadena, no solo del siguiente vuelo que va a despegar, sino de todos los vuelos de dicho aeropuerto, creando un efecto domino en los demás vuelos y aeropuertos. Los principales aeropuertos de la Unión Europea (UE) han llegado a anotar valores históricos en Julio del 2018 llegando a registrar una media de retrasos del 41 %, con un retraso medio de 45 minutos [7]. Estos retrasos pueden llevar a grandes pérdidas económicas para las compañías aéreas, que ven incrementadas en número de reclamaciones de los pasajeros.

Para controlar todo el tráfico aéreo y así evitar los retrasos, normalmente se utilizan radares, los cuales nos facilitan los datos relativos a la localización de un avión. Un radar primario localizado en tierra hace un barrido y detecta la posición aproximada haciendo uso de señales de radio. Más tarde un segundo radar se encarga de registrar el avión mediante un transponder, que es una pieza presente en todos los vuelos comerciales encargada de enviar un código de cuatro dígitos cuando reciben una señal de radar. También existe el protocolo ACARS (Aircraft Communications Addresing and Reporting System) encargado de enviar la información a tierra directamente a través de ondas de radio o por satélite, sin embargo la tecnología más novedosa son los sistemas llamados ADS-B (Automatic Dependent Surveillance Broadcast) actúan de la misma manera que un radar secundario pero al ser vía satélite evita la perdida de la señal por falta de cobertura.

El uso de los datos recogidos por los sistemas ADS-B nos proporcionan datos que pueden ser tratados mediante algoritmos de aprendizaje automático, para la predicción de tiempos de llegada y así evitar la problemática indicada anteriormente.

## 1.2. Objetivos

Este proyecto se ha realizado con el fin de hacer uso de técnicas de aprendizaje automático o machine learning en inglés, para lograr un modelo que nos permita elaborar predicciones sobre la hora en la que aterrizan unos determinados vuelos. Esto nos llevara al cumplimiento de una serie de objetivos que definiremos a continuación:

1. Estudio y entendimiento del tráfico aéreo, lo que nos llevará a interpretar correctamente los datos con los que se trabajará.
2. Aprendizaje de distintos algoritmos, con los que crear de un modelo de predicción de los tiempos de llegada de vuelos.
3. Creación de un programa con el que procesar los datos iniciales. Será el encargado de fusionar los datos obtenidos por el plan de vuelo, con los obtenidos por los sistemas ADS-B. También permitirá filtrar por el aeropuerto de destino.
4. Creación de una aplicación que nos permita visualizar los datos, modificaros y someterlos a los distintos algoritmos de aprendizaje, permitiendo realizar baterías de experimentos y mostrando los resultados.
5. Interpretación de los resultados obtenidos, con el fin de determinar con que grado de error es capaz de predecir el tiempo de llegada de un vuelo, así como ver que algoritmo de aprendizaje automático se adapta mejor a la problemática planteada.

## 1.3. Organización del documento

A continuación realizare un breve descripción de los capítulos que componen el documento, para que el lector tenga una idea de la composición del mismo.

- **Capítulo 1 - Introducción.** En este capítulo trataremos de contextualizar al lector sobre la motivación y objetivos que nos llevaron a la realización del proyecto.
- **Capítulo 2 - Gestión del tráfico aéreo.** En este capítulo nos centraremos en describir la problemática del control del tráfico aéreo, así como de realizar un análisis y estudio bibliográfico sobre otros trabajos relacionados con la predicción de tiempos de llegada de los aviones.
- **Capítulo 3 - Gestión y análisis del proyecto.** En el siguiente capítulo explicaremos la metodología utilizada, en la que nos basaremos para realizar un análisis para el desarrollo del proyecto. También se detallara la planificación y presupuesto del mismo.
- **Capítulo 4 - Marco teórico.** En este capítulo nos centraremos en ofrecer los contenidos teóricos sobre los que está basado el proyecto, que incluyen los algoritmos y tecnología utilizada para aplicarlos.

- **Capítulo 5 - Propuesta.** En este capítulo se describirán los datos con los que se va a trabajar, el programa de preprocesado de datos y la implementación de los distintos algoritmos.
- **Capítulo 6 - Dashboard.** En el siguiente capítulo se describirá el diseño de la aplicación, su posterior implementación y por último la realización de pruebas sobre la misma.
- **Capítulo 7 - Resultados.** En este capítulo se presentaran los resultados de la aplicación de los diferentes algoritmos, con la modificación de sus parámetros y el uso de diferentes datasets.
- **Capítulo 8 - Conclusiones y trabajo futuro.** En este capítulo haremos una evaluación del proyecto realizado, con el fin de poder proponer mejoras para futuros proyectos.

# Capítulo 2

# Gestión del tráfico aéreo

La gestión del tráfico aéreo se ha convertido en un tema de gran importancia en la actualidad, esto es debido a que cada vez son más la personas que utilizan este medio de transporte para sus desplazamientos, al mismo tiempo que aumenta el número de empresas que escogen el transporte aéreo como alternativa para el traslado de mercancías. Este año 2020 se estima un incremento del 137 % respecto al año 2004 lo que suponen 4.720 millones de pasajeros [15], y según International Air Transport Association (IATA) se estima que para el año 2035 se llegue a los 7.200 millones de pasajeros, necesitando que se incorporen 617.000 pilotos, 679.000 técnicos y 814.000 tripulantes de cabina[16]. En el ámbito nacional Aena ha registrado un incremento en tráfico del 5,8 % en pasajeros y operaciones, y del 9,9 % en mercancías [17].

Como consecuencia de del aumento del tráfico aéreo además de otras razones, se ha incrementado el número de retrasos en los vuelos. El año anterior, a 334 millones de pasajeros en Europa les retrasaron sus vuelos más de 15 minutos y a 23 millones se les cancelaron. Las causas fueron un 15 % debido a interrupciones, otro 25 % fue el resultado del clima, mientras que la capacidad y la escasez de personal de ATC (Control de Tráfico Aéreo) fueron la causa del 60 % de retrasos. Esto supone un gran coste para las aerolíneas que se ven en la obligación de asumir los costes extra de combustible, personal, indemnizaciones a los pasajeros, entre otras. En Europa el coste por minuto asciende a 100 euros, estimando que costarían a la economía 17.600 millones de euros en 2018 [19]. Para las aerolíneas de pasajeros en EEUU el coste por minuto en 2018 fue de 74,20 dólares, un 8,8 % más que el año anterior [18].

## 2.1. Contexto aéreo

La gestión del tráfico aéreo o ATM (Air Traffic Management) se encarga de controlar las aeronaves desde el despegue hasta el aterrizaje del mismo, permitiendo un cambio mínimo para transitar entre los espacios aéreos. En este proceso intervienen numerosos factores como la meteorología, el control de tránsito aéreo (ATC), sistemas de navegación, gestión del espacio (ASM), servicios (ATS) y control de flujo (ATFM).

El control de tránsito aéreo (ATC) divide el espacio en segmentos o secciones conocidas

como FIR (Flight Information Region), del que cada país es responsable del servicio correspondiente a su territorio, pero cuando el espacio se encuentra en aguas internacionales es controlada por una unidad llamada Centro de Control de Área [20]. Cada segmento del ATC irá informando al siguiente de la espera para la llegada de un vuelo y alguna modificación del mismo. El ATC tiene que recibir la información del vuelo recogida en el plan de vuelo (FP), que explicaremos más adelante, al menos con 30 minutos de antelación al despegue. Aparte del plan de vuelo podemos encontrar los Notam (Notification to Airmen), que nos informaran de cualquier cambio que pueda afectar a nuestro vuelo, y Weather document (WX), que nos proporciona información meteorológica. El ATC también será el encargado de comunicar datos como el tiempo, pistas en uso y otra información del aeródromo al avión.

Las aeronaves son supervisadas durante sus salidas, llegadas y sobrevuelos por los controladores. Los segmentos del ATC pueden subdividirse en sectores en los que 1 o 2 controladores se encargan de supervisar una cantidad aproximada de 15 aeronaves, las cuales son definidas vertical o geográficamente dependiendo de las necesidades de cada ruta. Los controladores pueden tener diferentes funciones cuando el avión se encuentra en ruta, entre las que está el controlador de planificación que es el encargado de asegurar que el tráfico entre y salga del sector a un nivel acordado, y el controlador ejecutivo que se asegura de mantener la separación de los límites del sector.

Las aeronaves pueden ser controladas de manera procesal, la cual está basada en informes de posición del piloto y estimaciones para planear niveles y rutas seguras. Esta forma de control se utiliza como refuerzo, ya que la manera principal de control se realiza mediante radares [21]. Generalmente se necesitan numerosos radares de largo alcance que permitan visualizar aeronaves a una distancia de 370km, si esto no fuera suficiente podrían obtener datos del control de terminal, que es un radar de menor alcance asociado con el aeropuerto [22].

El gran aumento del tráfico aéreo ha planteado la necesidad de construir una nueva generación de sistemas ATM. Proyectos como SESAR (Single European Sky ATM Research), en el ámbito europeo, y NextGen, en Estados Unidos, nacen con el objetivo de hacer uso de nuevos sistemas tecnológicos para la modernización de los sistemas ATM.

## 2.2. Plan de vuelo

Como hemos mencionado en la introducción de este capítulo, el plan de vuelo es un documento rellenado por el piloto o por la Autoridad de Aviación Civil, para ser enviado tiempo antes del despegue a la ATC. El plan de vuelo sigue el formato especificado en el documento ICAO 4444 [38]. Los datos presentes en el plan de vuelo contienen generalmente información básica sobre los puntos de despegue y aterrizaje, el tiempo estimado que el avión estará en ruta, aeropuertos alternativos en caso de que se presenten condiciones meteorológicas adversas, reglas de vuelo por instrumentos o Instrument flight rules (IFR) [1], información del piloto, nú-

[1]Conjunto de normas y procedimientos que regulan el vuelo de aeronaves en casos en los que la capacidad del piloto para ver y evitar colisiones esta reducida, o es inexistente.

mero de personas a bordo, e información sobre el avión. En la mayoría de países se requieren que los planes de vuelo estén sujetos a las IFR, pero se puede volar bajo las reglas de vuelo visual o Visual flight rules (VFR) [2] cuando se vuela sobre aguas internacionales. Se recomienda especialmente hacer uso de planes de vuelo cuando se viaja sobre áreas inhabitadas como los océanos, ya que pueden generar una alerta en caso de retraso. En los vuelos VFR solo se requiere proporcionar la información necesaria en caso de que sea necesario una búsqueda y rescate, o para el control del tráfico aéreo en caso de que se vuele por un área de reglas especiales de vuelo.

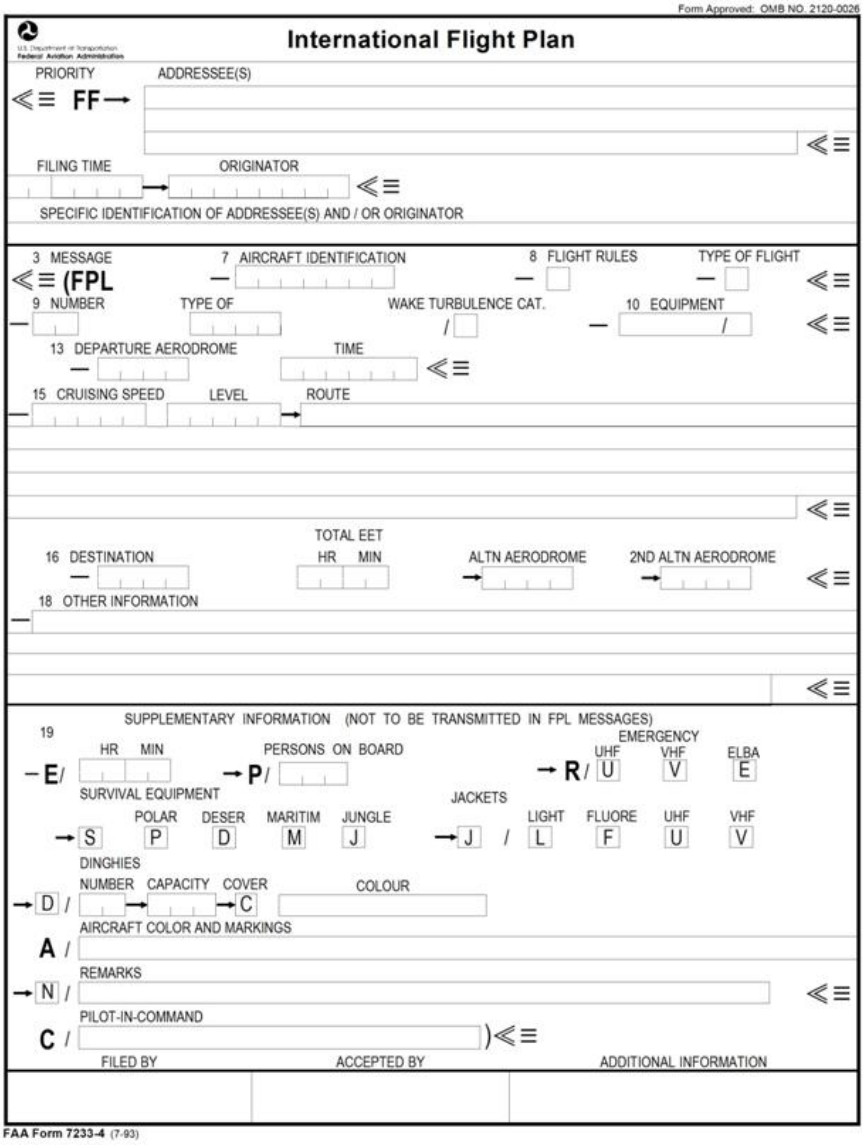



Figura 2.1: Plan de vuelo internacional.

Para este proyecto será muy útil mucha de la información proporcionada por los planes de

[2]Conjunto de normas que establecen las condiciones suficientes para que un piloto pueda dirigir su aeronave con la única ayuda de la observación visual.

vuelo, tales como la aerolínea que opera el vuelo, el tipo de avión, los tiempos de inicio y final del vuelo entre otros. La Organización Europea para la Seguridad de la Navegación Aérea o EUROCONTROL pone a disposición del público la información relativa a los planes de vuelos que han accedido a la aérea europea, a través del Demand Data Repository (DDR) [3].

## 2.3. Sistema ADS-B

La tecnología ADS-B (Automatic Dependent Surveillance-Broadcast) nace con el fin de aumentar la capacidad del espacio aéreo, reduciendo la separación entre aeronaves de 30NM, que equivalen a 55,5 km, a tan solo 5NM, que equivalen a 9,26km [8]. Cada vez es mayor el número de aviones equipados con este sistema, aumentando también el número de estaciones receptoras. Según los criterios establecidos a partir de 2020 las aeronaves deberán portar el sistema ADS-B. No obstante en zonas como América del Sur, Asia o África existen muchas zonas exentas de receptores, lo que provoca la pérdida de información de muchos vuelos que pasen por dichas zonas. OpenSky es la mayor red encargada de recopilar datos sobre la vigilancia del tráfico aéreo, actualmente cuenta con más de 1.000 receptores concentrados principalmente en EEUU y Europa, recibiendo más de diez billones de mensajes ADS-B y Modo S [10]. También cabe destacar otras fuentes de obtención de datos como ADSBHub y Frambuesa, el primero cubre gran parte de Europa centrándose en la zona norte y el segundo se limita a cubrir parte del territorio español.

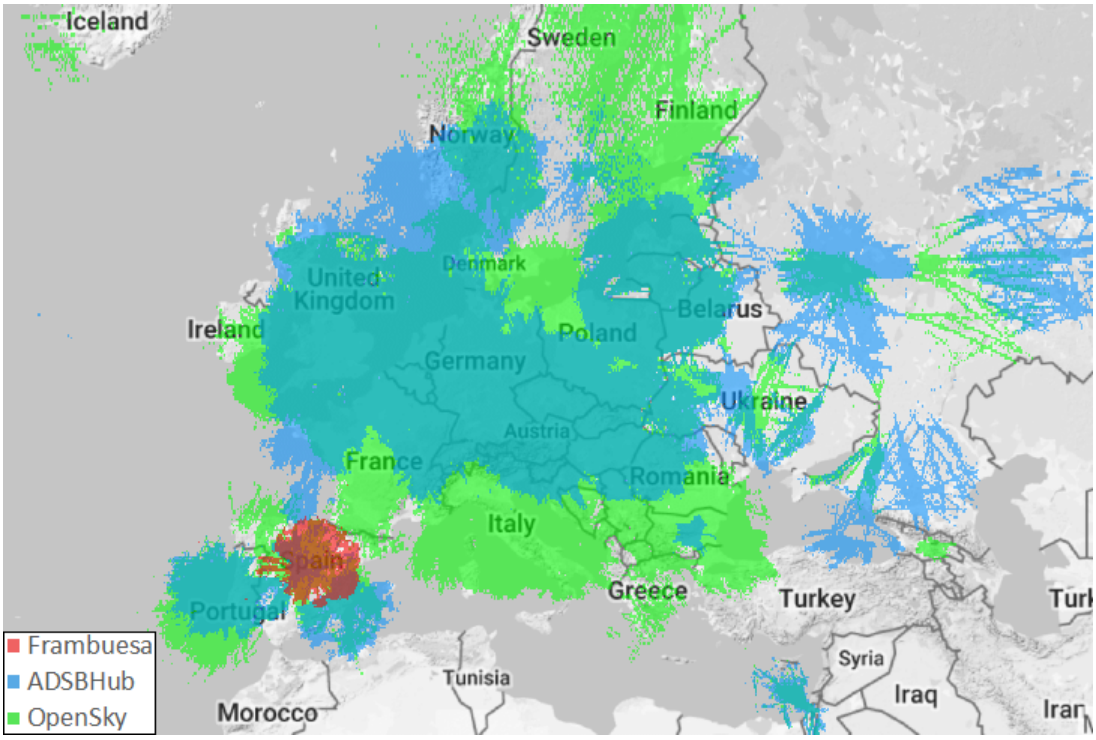



Figura 2.2: Cobertura en Europa ADS-B[11].

[3] https://www.eurocontrol.int/ddr

Hasta la actualidad los radares han sido los encargados de recoger la información acerca de las diferentes etapas de un vuelo, esto ha cambiado con la aparición del sistema ADS-B, el cual utiliza la señal GPS para determinar la posición de las aeronaves, que son enviadas automáticamente junto a otros datos del vuelo mediante radiodifusión. Esta información puede ser recibida e interpretada por cualquier receptor, se encuentre en tierra o aire. A continuación en la Figura 2.3 se muestra una imagen del funcionamiento de un sistema ADS-B.

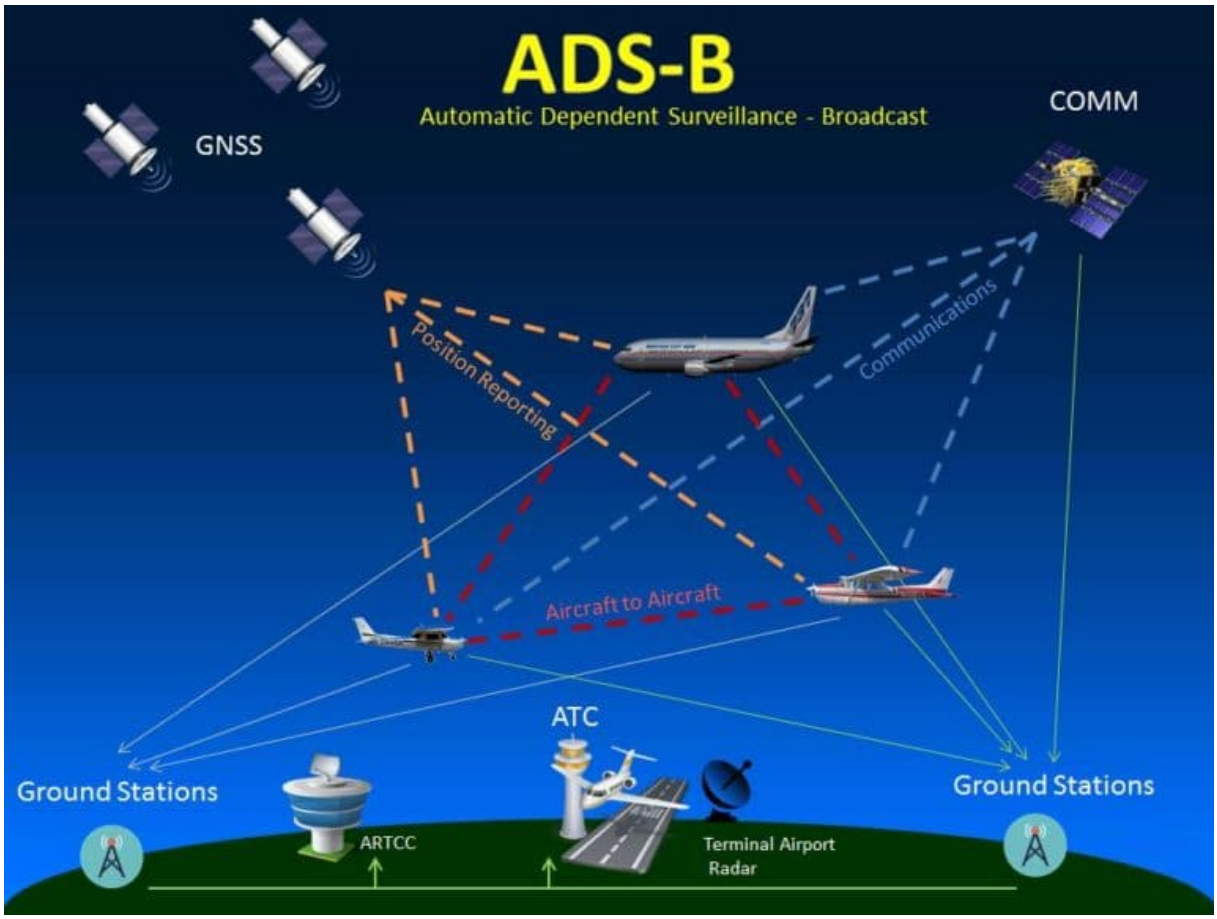



Figura 2.3: Funcionamiento ADS-B [9].

Los mensajes ADS-B se transmiten aproximadamente en 0,5 segundos y son recibidos en las estaciones terrestres compatibles, las cuales se encargan de enviarlos donde corresponda. ADS-B soporta varios tipos de mensajes, pero únicamente nos interesan los de tipo *MSG* en los que están contenidos la mayoría de los datos de la aeronave.

| # | Name | Description |
|---|---|---|
| 1 | Message | MSG,STA,ID,AIR,SEL,CLK. |
| 2 | Type | Type of MSG message: 1-8. |
| 3 | SessionID | Database session identifier. |
| 4 | AircraftID | Database aircraft record number. |
| 5 | HexIdent | ICAO 24-bit address (*Mode-S hexadecimal code*). |
| 6 | FlightID | Database flight record number. |
| 7 | Gen. Date | Generated date (from the aircraft). |
| 8 | Gen. Time | Generated time (from the aircraft). |
| 9 | Logged Date | Logged date (from the base station). |
| 10 | Logged Time | Logged time (from the base station). |
| 11 | Callsign | Flight identifier (ICAO airline code + flight number). |
| 12 | Altitude | Barometric altitude (25ft or 100 ft resolution). |
| 13 | Speed | Ground speed (knots). |
| 14 | Track | Derived from the velocity E/W and N/S. (degrees) |
| 15 | Latitude | North/East positive, South/West negative (degrees). |
| 16 | Longitude | North/East positive, South/West negative. |
| 17 | Vertical | Aircraft vertical rate (ft/sec). |
| 18 | Squawk | Assigned squawk code. |
| 19 | Alert | Flag to indicate the squawk value has changed. |
| 20 | Emergency | Flag to indicate emergency code has been set. |
| 21 | SPI (Ident) | Flag to indicate transponder Ident has been activated. |
| 22 | IsOnGround | Flag to indicate ground squat switch is active. |

Figura 2.4: Campos usados en los mensajes ADS-B [9].

| | 1...10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSG-1 | Mandatory | Call | | | | | | | | | | | |
| MSG-2 | | | Alt | Sp | Trk | Lat | Lon | | | | | | Gro |
| MSG-3 | | | Alt | | | Lat | Lon | | | Aler | Em | SPI | Gro |
| MSG-4 | | | | Sp | Trk | | | VSp | | | | | |
| MSG-5 | | | Alt | | | | | | | Aler | | SPI | Gro |
| MSG-6 | | | Alt | | | | | | Sq | Aler | Em | SPI | Gro |
| MSG-7 | | | Alt | | | | | | | | | | Gro |
| MSG-8 | | | | | | | | | | | | | Gro |

Figura 2.5: Campos usados por cada tipo de mensajes ADS-B [9].

La Figura 2.4 nos muestra los diferentes campos utilizados por los mensajes *MSG*, los primeros 10 campos son comunes a todos los tipos de mensajes. En la Figura 2.5 podemos ver los campos de cada tipo de mensaje *MSG*, siendo el 1 la identificación del vuelo, el 2 posición de la superficie, el 3 la posición en el aire, el 4 velocidad en el aire, el 5 altitud de vigilancia, el 6 identificación de vigilancia, el 7 mensaje aire-aire, y 8 todas las llamadas de respuesta [9].

## 2.4. Trabajo relacionado

A continuación se mencionaran trabajos relacionados con el objetivo de nuestro proyecto, de predecir el tiempo de llegada de aviones.

- **Predicting Estimated Time of Arrival for Commercial Flights** [23].

  En este documento los autores elaboran predicciones sobre el tiempo de llegada de aviones en los aeropuertos españoles, basándose en datos sobre la trayectoria, meteorología, el tráfico de los aeropuertos, y la información de sectores por los que vuela la aeronave, referente al espacio aéreo. Todos estos datos proporcionan un total de 34 características, con lo que se concluye finalmente que la más influyente es la del aeropuerto de llegada, seguida de las relacionadas con variables atmosféricas. En cuanto a modelos de predicción se utilizan métodos lineales como Linear Regression (LR), Lasso Regression (LASSO), y Elastic Net Regression (EN); no lineales como Classification and Regression Trees, Support Vector Regression (SVR), y k-Nearest Neighbors (KNN); "ensemble" como Adaptive Boosting (AdaBoost o AB), Gradient Boosting (GBM), Random Forest Regression (RF), y Extra Trees Regression (ET); y por ultimo redes neuronales recurrentes Long Short-Term Memory (LSTM). Los datos fueron ordenados cronológicamente por cada ruta aérea, utilizando un 80 % para entrenar los modelos y un 20 % para la validación de los mismos. Para el entrenamiento del modelo se utilizó validación cruzada con 10 iteraciones y se evaluaron los algoritmos utilizando la raíz del error cuadrático medio o Root Mean Squared Error (RMSE) en inglés. Los resultados nos muestran que los dos algoritmos que mejor predicen el tiempo son AB y GBM dando como resultado un error en torno a los 3 minutos.

- **A Tree-Based Ensemble Method for the Prediction and Uncertainty Quantification of Aircraft Landing Times** [24].

  El siguiente estudio hace uso de los datos de uno de los aeropuertos de Estados Unidos, el Dallas/Fort Worth International Airport (DFW). El algoritmo de predicción utilizado es Quantile Regression Forests algorithm (QRF), el cual se extiende del algoritmo Randon Forest (RF). Se han utilizado 11 variables, referentes a la posición del avión entre otras. La variable con más importancia es la distancia actual de la aeronave con respecto al aeropuerto, seguido de la altitud, y la que menos relevancia tiene es la hora del día. Respecto los datos se han utilizado los datos de 5 días, separando el 67 % para entrenar el modelo y un 33 % de prueba. Como resultados obtuvieron un error absoluto medio o mean Absolute Errors (MAE) por debajo de los 80 segundos con distancias de 60 millas náuticas, llegando a disminuir el error a menos de un minuto con distancias menores a 20 millas náuticas.

- **Data-driven aircraft estimated time of arrival prediction** [25].

  En este caso se hace uso de 18 características relacionadas con la información del vuelo, 4 con el tráfico aéreo, y 8 con la información meteorológica. La característica con mayor relevancia es el aeropuerto de destino, siendo las otras características mucho menos importantes con valores similares. Los datos constan de 24.787 vuelos comerciales dentro del territorio estadounidense, dividiéndose el 80 % para entrenamiento y el 20 % de prueba. El algoritmo de predicción utilizado es el Random Forest (RF), se determina que la mejor configuración es de 100 árboles con un tamaño mínimo de hoja de 5 instancias. Los resultados son comparados al de las predicciones hechas por Enhanced Traffic Management System (ETMS), solo en 12 casos el modelo RF tuvo un error absoluto más grande que el de ETMS por 12 minutos o más.

- **Predicting flight arrival times with a multistage model** [26].

  El siguiente documento se quiere predecir la llegada de los vuelos pertenecientes a EEUU, para lo que cuentan con 56 características diferentes provenientes de datos con la información de vuelo, la meteorología, y el tráfico aéreo entre otros. Los datos fueron tomados durante 109 días. Se utiliza un modelo de 6 fases en las que pasa por dos Ridge Regression (RR) y tres Gradient Boosting Machines (GBT), para predecir el error se utiliza la raíz del error cuadrático medio o Root Mean Squared Error (RMSE). Como conclusión extrae que la simplificación del modelo reduce el tiempo requerido en un 98,7 % y solo se incrementa el error en un 5 %.

En el primer y tercer estudio destacan que la variable del aeropuerto de llegada o destino es la que proporciona mayor información, mientras que en el segundo es la distancia actual de la aeronave con respecto al aeropuerto. En cuanto al porcentaje de datos utilizados para el entrenamiento de los modelos, se tiene que el primer y tercer estudio utilizan el 80 % de los datos, a diferencia del segundo que reduce este porcentaje a 67 %. Respecto a los modelos de predicción, podemos ver como el primer estudio a diferencia de los otros tres, que únicamente emplean algoritmos basados en árboles, utiliza también métodos lineales y redes neuronales recurrentes. Siendo los modelos basados en arboles los que mejores resultados arrojan. Los resultados obtenidos por el primer estudio fueron evaluados mediante el RMSE dando errores de 3 minutos aproximadamente. El segundo estudio, que evaluó los resultados mediante el MAE, da mejores resultados, estando por debajo los 80 segundos, pero está limitado distancias de 60 millas náuticas.

# Capítulo 3

# Análisis y gestión del proyecto

En este capítulo empezamos nombrando diferentes tipos de metodologías a utilizar para desarrollar un proyecto, describiendo la metodología escogida para este trabajo. Una vez elegida la metodología pasaremos a realizar un análisis del sistema a desarrollar, con el fin de obtener una planificación. Por último detallaremos el coste de los diferentes componentes del proyecto para elaborar un presupuesto final.

## 3.1. Metodología

La gestión de proyectos se compone de una serie de metodologías que nos guían para conseguir gestionar la evolución de un proyecto, tanto en su inicio como en su desarrollo. También nos ayudan a controlar y dar respuesta ante problemas que puedan surgir en un proyecto. Existen varios tipos de metodologías, destacando por un lado las tradicionales que están basadas en etapas secuenciales, en las que para empezar una fase tiene que haber terminado la anterior. A consecuencia de esto, los proyectos son muy rígidos, siendo necesario volver al inicio en caso de querer realizar algún cambio. Las principales metodologías dentro de esta categoría son Critical Path Method (CPM) y Critical Chain Project Management (CCPM). Otra metodología importante es la PMI o PMBOK, la cual consiste en seguir las cinco fases de gestión de proyectos descritas en la Guía del cuerpo de conocimiento de la gestión de proyectos o Guide to the Project Management Body of Knowledge (PMBOK) [40] en inglés.

Para la gestión de este proyecto vamos a fijarnos en metodologías ágiles, las cuales están siendo ampliamente utilizadas en la actualidad. Esto es debido a que la mayoría de proyectos hoy en día están sometidos a constantes cambios, causados entre otras cosas por los continuos avances tecnológicos que hacen que cambien los requisitos de un proyecto. Las metodologías ágiles proporcionan una mayor adaptabilidad a cambios que las metodologías anteriormente citadas. Los principios de estas metodologías, establecidos en el Manifiesto Ágil [41] se describen a continuación:

- Dar más prioridad al software funcional que a la documentación, consiguiendo así hacernos una pequeña idea de cómo será el funcionamiento de la aplicación final, sin necesidad

de desarrollarla por completo. Esto permite generar nuevas ideas sobre el proyecto, lo cual sería mucho más difícil que si solo tuviéramos los requisitos iniciales del proyecto.

- En las metodologías ágiles tiene una elevada importancia la iteración con el cliente, sometiendo al proyecto a evaluaciones continuas, en las que el cliente colabora aportando su punto de vista.

- Los cocimientos del equipo de trabajo son utilizados para alcanzar los niveles deseados de calidad, haciendo que predomine la participación de las personas involucradas en el proyecto antes que las herramientas usadas para el desarrollo, las cuales tampoco deben perder importancia.

- Es muy importante dar una respuesta adecuada a los cambios que puedan surgir en el proyecto, pasando a un segundo plano el seguimiento de un plan. Esto es así, debido a que cuando se tienen proyectos con requisitos inestables se valora más la capacidad de respuesta a los cambios surgidos que el seguimiento de los planes establecidos.

Las metodologías ágiles más utilizadas son Kanban, Programación extrema (XP), y Scrum. XP [42] tiene como base la simplicidad y como objetivo la satisfacción al cliente, basándose en un conjunto de reglas y buenas prácticas para el desarrollo software, el cual es efectuado en parejas, teniendo pruebas unitarias y corrección de errores continuas, en ambientes en los que los requisitos son muy cambiantes. Kanban [42] representa la planificación y asignación de actividades de forma muy simple, se hace uso de un tablero de mínimo tres columnas: Pendiente, En Progreso, y Terminado. En este tablero estarán descritos todos los procesos del flujo de trabajo. Por último, Scrum es la metodología ágil más popularizada, por lo que será la elegida para la gestión del proyecto, siendo descrita a continuación.

Scrum [42] es una metodología enfocada en un marco de trabajo de procesos ágiles, basándose en un ciclo de vida iterativo e incremental. Se parte de una visión general de la aplicación a desarrollar, a través de la cual se va especificando y detallando la funcionalidad a desarrollar. Las diferentes fases del desarrollo son denominadas sprints, suelen tener una duración máxima de un mes, y al finalizar debe entregarse una versión funcional del producto. Las reuniones son un elemento muy importante, realizándose tanto encuentros diarios con el fin de hacer un seguimiento del proyecto, como al finalizar cada sprint para valorar el resultado del mismo. En este proyecto las reuniones de seguimiento serán semanales en vez de diarias, debido a la disponibilidad de mis tutores.

En la Figura 3.1 se muestra el funcionamiento de scrum dividido en cuatro pasos:

1. Primeramente empleamos el Product Backlog para definir el alcance del proyecto. Normalmente se sigue el formato de historias de usuario.

2. Se seleccionan de cada sprint una serie de historias de usuario del Product Backlog para proceder a su desarrollo, lo que conoce como Sprint Backlog. El Sprint Backlog contendrá el plan de acción con las tareas a realizar cada sprint, en función de las historias de usuario a desarrollar.

3. El producto final va tomando forma en cada iteración, añadiéndose funcionalidades potencialmente entregables en cada sprint.

4. El cliente participa en las sucesivas iteraciones generando un feedback que sirve para seguir desarrollando la aplicación.

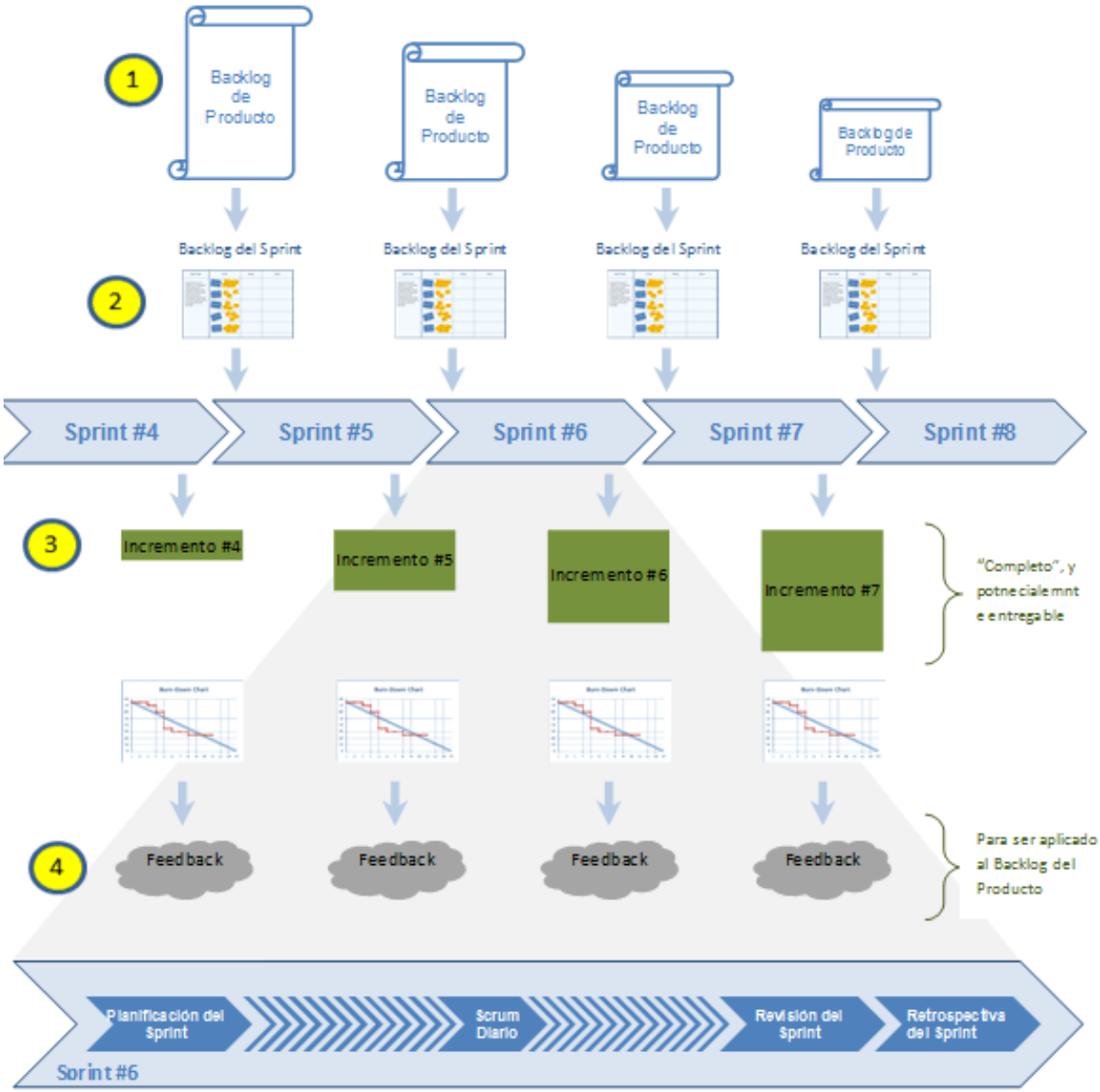



Figura 3.1: Funcionamiento de Scrum [37].

Para terminar de definir la metodología Scrum se presentaran en la Tabla 3.1 los diferentes roles de los participantes en el proyecto, los artefactos y eventos.

| Componente | Tipos | Descripción |
|---|---|---|
| Roles | Equipo scrum | Grupo de profesionales que realizan el incremento en cada sprint. |
| | Product Owner | Persona encargada de asegurar que el equipo aporte valor de negocio. |
| | Scrum Master | Responsable del cumplimiento de las reglas de un marco de Scrum técnico. |
| Artefactos | Product Backlog | El Product Backlog es la lista ordenada de todo aquello que el Product Owner cree que necesita el producto. |
| | Sprint Backlog | El Sprint Backlog es la lista de las tareas necesarias para construir las historias de usuario que se van a realizar en un sprint. |
| | Incremento | El incremento es la parte de producto producida en un sprint, y tiene como característica el estar completamente terminado y operativo. |
| Eventos | Sprint | El evento clave de Scrum para mantener un ritmo de avance continuo es el sprint, el periodo de tiempo acotado de duración máxima de 4 semanas, durante el que se construye un incremento del producto. |
| | Reunión de planificación del sprint | En esta reunión se toman como base las prioridades y necesidades de negocio del cliente, y se determinan cuáles y cómo van a ser las funcionalidades que se incorporarán al producto en el siguiente sprint. |
| | Scrum diario | Reunión diaria breve, de no más de 15 minutos, en la que el equipo sincroniza el trabajo y establece el plan para las siguientes 24 horas. |
| | Revisión del sprint | Reunión realizada al final del sprint para comprobar el incremento. |
| | Retrospectiva del sprint | Reunión en la que se realiza un autoanálisis de la forma de trabajar e identifica fortalezas y puntos débiles. |

Tabla 3.1: Descripción componentes de Scrum

## 3.2. Análisis

En esta sección se detallará el proceso de análisis del sistema desarrollado, el cual constará de una aplicación web sobre la que se realizarán diferentes operaciones con algoritmos de Machine Learning sobre los datos, ofreciéndonos una visualización de los resultados. Para este proyecto realizaremos un análisis de forma incremental, de manera que el conjunto completo de requisitos se considerará durante todo el proyecto.

### 3.2.1. Actores del Sistema

A continuación procederemos a especificar los actores que interactuarán con la aplicación, pudiendo ser tanto personas físicas como sistemas externos.

| ACT-01 | Usuario General |
|---|---|
| Versión | 1.0 (10/05/2019) |
| Descripción | Un usuario general será cualquier persona que tenga acceso a la aplicación e interactúe con ella, no siendo necesario registrarse. |
| Comentarios | Ninguno |

Tabla 3.2: Actor Usuario General

### 3.2.2. Product Backlog

El Product Backlog consiste en desarrollar un listado ordenado y priorizado de todos los requisitos y cambios a realizar en el proyecto. Estos requisitos serán representados mediante historias de usuario. Una parte de ellas describirán el proceso de aprendizaje y documentación necesario para este proyecto, mientras que la otra parte se centrará el comportamiento que deberá tener la aplicación para el usuario. Estas historias serán elaboradas de una manera simple que permita su comprensión para poder priorizar entre ellas. El cumplimiento de estas historias de usuario dará como resultado la materialización de los objetivos del proyecto.

A continuación se especificaran las historias de usuario que componen nuestro Product Backlog, dentro de la definición de las épicas que componen el proyecto. Únicamente hay una historia que no pertenece a ninguna épica, que se nombra a continuación.

| ID | Descripción |
|---|---|
| US-34 | Realización de los experimentos con diferentes datos y algoritmos. |

Tabla 3.3: Product Backlog

### Épicas e historias de usuario

Anteriormente se han descrito las historias de usuario como requisitos a cumplir para llevar a cabo el desarrollo del proyecto. Estas historias son descripciones simples, expresadas en un lenguaje coloquial y pretenden ser un recordatorio de la conversación con el cliente. Cuando una historia es lo suficientemente compleja como para descomponerse en más historias es llamada épica. Este proyecto estará compuesto por cuatro épicas, que se pasaran a definir en las siguientes Tablas.

| ID | EP-01 | Descripción | Instalación de componentes y adquisición de conocimientos previos |
|---|---|---|---|
| Historias de usuario | | | |
| US-01 | Instalación de LaTeX y aprendizaje de conocimientos básicos | | |
| US-02 | Estudio de la gestión del tráfico aéreo | | |
| US-03 | Análisis y documentación de estudios de predicción de tiempos de llegada de vuelos con algoritmos de machine learning | | |
| US-06 | Instalación de Python Anaconda | | |
| US-07 | Instalación, aprendizaje de Spark y posterior documentación | | |
| US-10 | Estudio de los algoritmos regresivos DT, GBT, y RF y sus diferentes parámetros en MLlib | | |

Tabla 3.4: Épica 1.

| ID | EP-02 | Descripción | Implementación de funciones con Spark |
|---|---|---|---|
| Historias de usuario | | | |
| US-08 | Implementación de preprocesado de datos mediante Spark SQL | | |
| US-12 | Estudio e implementación de la transformación de características con MLlib | | |
| US-13 | Configuración de los algoritmos: Decision tree, Gradient-boosted Tree y Random forest | | |

Tabla 3.5: Épica 2.

| ID | EP-03 | Descripción | Elaboración de una aplicación web que nos permita visualizar y ejecutar los algoritmos sobre una serie de datos |
|---|---|---|---|
| Historias de usuario | | | |
| US-15 | Aprendizaje de Dash para la elaboración del dashboard | | |
| US-16 | Análisis de las funcionalidades dashboard | | |
| US-18 | Carga de un dataset | | |
| US-19 | Visualización y selección de columnas del dataset | | |
| US-20 | Visualización de una tabla y un gráfico con información de una columna | | |
| US-21 | Selección de las columnas a utilizar en los algoritmos | | |
| US-22 | Borrado de filas de una determinada columna | | |
| US-23 | Generación de un dataset con las columnas y las filas modificadas anteriormente | | |
| US-24 | Selección y modificación de parámetros del algoritmo a utilizar, además de la elección del porcentaje de división del dataset y la columna a predecir | | |
| US-25 | Visualización los resultados obtenidos por el algoritmo utilizado | | |
| US-26 | Configuración de un nuevo dataset a la pestaña experimentos | | |
| US-27 | Eliminación de un dataset en la pestaña experimentos | | |
| US-28 | Configuración de un algoritmo a la pestaña experimentos | | |
| US-29 | Eliminación un algoritmo en la pestaña experimentos | | |
| US-30 | Visualización de un listado con los resultados, donde se podrá pinchar sobre uno y ver los resultados detalladamente | | |
| US-31 | Descripción de los parámetros de los algoritmos | | |
| US-32 | Probar la funcionalidad global del dasboard | | |

Tabla 3.6: Épica 3.

| ID | EP-04 | Descripción | Elaboración de una memoria que documente el proyecto realizado |
|---|---|---|---|
| Historias de usuario | | | |
| US-04 | Redacción del capítulo 1 “Introducción” | | |
| US-05 | Redacción del capítulo 2 “Gestión del tráfico aéreo” | | |
| US-09 | Gestión de riesgos | | |
| US-11 | Elaboración del capítulo 4 “Marco teórico” | | |
| US-14 | Redacción del capítulo 5 “Propuesta” | | |
| US-17 | Redacción del capítulo 3 “Análisis y gestión del proyecto” | | |
| US-33 | Redacción del capítulo 6 “Dashboard” | | |
| US-35 | Redacción del capítulo 7 “Resultados” | | |
| US-36 | Redacción del capítulo 8 “Conclusiones y trabajo futuro” | | |

Tabla 3.7: Épica 4.

En la Tabla 3.8 se muestra la plantilla mediante la cual vamos a definir más detalladamente las historias de usuario mencionadas en el Product Backlog. Dentro de las historias de usuario tendremos la definición de las mismas en base al usuario que va dirigida, lo que quiere conseguir y para que lo quiere conseguir. Otra parte de suma importancia serán los criterios de aceptación, e cumplimento de estos criterios será necesario para dar por finalizada la historia. En las siguientes tablas se muestra a definición de las historias de usuario descritas en el Product Backlog.

| **ID** | Identificador | **TÍTULO** | Título descriptivo | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | Para determinar el orden de implementación | **ESTIMACIÓN** | Estimación en tiempo ideal de implementación |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario interesado en la acción | | | | |
| Quiero... | Acción que queremos que ocurra | | | | |
| Para... | El beneficio que queremos lograr (opcional) | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| Número | Título del escenario | | | | |
| Dado que | Precondición / Descripción | | | | |
| Cuando | Acción que el usuario ejecuta | | | | |
| Entonces | Comportamiento del sistema | | | | |

Tabla 3.8: Modelo de definición de historia.

| **ID** | US-1 | **TÍTULO** | Instalación de LaTeX y aprendizaje de conocimientos básicos | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Aprender e instalar LaTeX | | | | |
| Para... | Redactar la memoria de la aplicación | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Aprendizaje e instalación de LaTeX | | | | |
| Dado que | Instalo y aprendo LaTeX | | | | |
| Cuando | Redacte la memoria | | | | |
| Entonces | Obtendré una memoria redactada en el formato LaTeX | | | | |

Tabla 3.9: Historia de usuario 1.

| **ID** | US-2 | **TÍTULO** | Estudio de la gestión del tráfico aéreo | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Estudiar la gestión del tráfico aéreo | | | | |
| Para... | Conocer el origen de los datos tratados | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Conocimiento de la gestión del tráfico aéreo | | | | |
| Dado que | Estudio la gestión del tráfico aéreo | | | | |
| Cuando | Trate los datos | | | | |
| Entonces | Podré hacer una correcta interpretación de los datos | | | | |

Tabla 3.10: Historia de usuario 2.

| **ID** | US-3 | **TÍTULO** | Análisis y documentación de estudios de predicción de tiempos de llegada de vuelos con algoritmos de machine learning | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Documentar estudios de predicción de tiempos de llegada de vuelos con algoritmos de machine learning | | | | |
| Para... | Conocer la metodología y resultados obtenidos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Documentación de estudios de predicción de tiempos de llegada de vuelos con algoritmos de machine learning | | | | |
| Dado que | Busco estudios acerca de la predicción de vuelos con algoritmos de machine learning | | | | |
| Cuando | Documento esos estudios | | | | |
| Entonces | Podré sacar unas conclusiones acerca de los algoritmos utilizados y los resultados obtenidos | | | | |

Tabla 3.11: Historia de usuario 3.

| **ID** | US-4 | **TÍTULO** | Redacción del capítulo 1 “Introducción” | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Tener un capitulo introductorio del proyecto en la memoria | | | | |
| Para... | Introducirme en el proyecto realizado | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Redacción del capítulo 1 “Introducción” | | | | |
| Dado que | Leo la memoria | | | | |
| Cuando | Me encuentre en el capítulo 1 | | | | |
| Entonces | Podré tener una visión general acerca del proyecto y sus objetivos | | | | |

Tabla 3.12: Historia de usuario 4.

| **ID** | US-5 | **TÍTULO** | Redacción del capítulo 2 “Gestión del tráfico aéreo” | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Tener un capítulo sobre gestión del tráfico aéreo en la memoria | | | | |
| Para... | Introducirme en el origen de los datos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Redacción del capítulo 2 “Gestión del tráfico aéreo” | | | | |
| Dado que | Leo la memoria | | | | |
| Cuando | Me encuentre en el capítulo 2 | | | | |
| Entonces | Podré tener una visión general acerca de la gestión del tráfico aéreo | | | | |

Tabla 3.13: Historia de usuario 5.

| **ID** | US-6 | **TÍTULO** | Instalación de Python Anaconda | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Instalar Python Anaconda | | | | |
| Para... | Poder programar en Python | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Instalación de Python Anaconda | | | | |
| Dado que | Instalo Python Anaconda | | | | |
| Cuando | Use el programa Spyder | | | | |
| Entonces | Podré programar y ejecutar programas en Python | | | | |

Tabla 3.14: Historia de usuario 6.

| **ID** | US-7 | **TÍTULO** | Instalación, aprendizaje de Spark y posterior documentación | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Documentarme acerca del origen y funcionamiento de Spark | | | | |
| Para... | Aprender a utilizar sus componentes | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Instalación, aprendizaje de Spark y posterior documentación | | | | |
| Dado que | Me instalo Spark y aprendo acerca de su funcionamiento | | | | |
| Cuando | Necesite utilizar sus componentes | | | | |
| Entonces | Tendré nociones acerca de su funcionamiento | | | | |

Tabla 3.15: Historia de usuario 7.

| **ID** | US-8 | **TÍTULO** | Implementación de preprocesado de datos mediante Spark SQL | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Disponer de un programa de preprocesado de datos mediante Spark SQL | | | | |
| Para... | Poder obtener un CSV con los datos pertenecientes a los vuelos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Implementación de preprocesado de datos mediante Spark SQL | | | | |
| Dado que | Dispongo de dos CSV con datos relacionados con los vuelos | | | | |
| Cuando | Introduzca los dos CSV y el aeropuerto de destino por el que quiero filtrar los datos | | | | |
| Entonces | Obtendré un solo CSV con los datos filtrados | | | | |

Tabla 3.16: Historia de usuario 8.

| **ID** | US-9 | **TÍTULO** | Gestión de riesgos | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Disponer de un plan de gestión de riesgos | | | | |
| Para... | Poder evitar posibles retrasos en el proyecto | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Gestión de riesgos | | | | |
| Dado que | El desarrollo del proyecto puede estar afectado por una serie de riesgos | | | | |
| Cuando | Alguno de esos riesgos ocurra | | | | |
| Entonces | Sabré como actuar para no afectar a la planificación | | | | |

Tabla 3.17: Historia de usuario 9.

| **ID** | US-10 | **TÍTULO** | Estudio de los algoritmos regresivos DT, GBT, y RF y sus diferentes parámetros en MLlib | |
|---|---|---|---|---|
| | | **PRIORIDAD** | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | |
| Como... | Desarrollador | | | |
| Quiero... | Estudiar los algoritmos regresivos DT, GBT, y RF y sus diferentes parámetros en MLlib | | | |
| Para... | Poder utilizarlos en la aplicación | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | |
| 1 | Estudio de los algoritmos regresivos DT, GBT, y RF y sus diferentes parámetros en MLlib | | | |
| Dado que | La aplicación hará uso de los algoritmos DT, GBT, y RF | | | |
| Cuando | Necesite aplicarlos a los datos | | | |
| Entonces | Sabré como poder utilizarlos en conjunto con sus parámetros | | | |

Tabla 3.18: Historia de usuario 10.

| **ID** | US-11 | **TÍTULO** | Elaboración del capítulo 4 “Marco teórico” | |
|---|---|---|---|---|
| | | **PRIORIDAD** | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | |
| Como... | Usuario | | | |
| Quiero... | Tener un capítulo con los conceptos teóricos necesarios para la realización del proyecto en la memoria | | | |
| Para... | Poder entender los algoritmos y tecnologías aplicadas | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | |
| 1 | Elaboración del capitulo 4 “Marco teórico” | | | |
| Dado que | Leo la memoria | | | |
| Cuando | Me encuentre en el capítulo 4 | | | |
| Entonces | Podré tener una definición de los conceptos teóricos aplicados | | | |

Tabla 3.19: Historia de usuario 11.

| **ID** | US-12 | **TÍTULO** | Estudio e implementación de la transformación de características con MLlib | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Estudiar e implementar de la transformación de características con MLlib | | | | |
| Para... | Poder transformar los datos para ser procesador por los algoritmos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Estudio e implementación de la transformación de características con MLlib | | | | |
| Dado que | La aplicación hará uso de los algoritmos DT, GBT, y RF | | | | |
| Cuando | Necesite aplicarlos a los datos | | | | |
| Entonces | Deberán haberse transformado los datos para poder aplicar los algoritmos | | | | |

Tabla 3.20: Historia de usuario 12.

| **ID** | US-13 | **TÍTULO** | Configuración de los algoritmos: Decision tree, Gradient-boosted Tree y Random forest | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Desarrollar funciones con Spark y MLib | | | | |
| Para... | Poder aplicar los algoritmos Decision tree, Gradient-boosted Tree y Random forest | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Configuración de los algoritmos: Decision tree, Gradient-boosted Tree y Random forest | | | | |
| Dado que | La aplicación hará uso de los algoritmos DT, GBT, y RF | | | | |
| Cuando | Necesite utilizarlos en la aplicación | | | | |
| Entonces | Deberán haberse desarrollado funciones con las que aplicar los algoritmos | | | | |

Tabla 3.21: Historia de usuario 13.

| **ID** | US-14 | **TÍTULO** | Redacción del capítulo 5 “Propuesta” | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Tener un capítulo que explique las funciones desarrolladas en Spark para el tratamiento de los datos | | | | |
| Para... | Poder entender los procedimientos utilizados | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Redacción del capítulo 5 “Propuesta” | | | | |
| Dado que | Leo la memoria | | | | |
| Cuando | Me encuentre en el capítulo 5 | | | | |
| Entonces | Podré tener una explicación de las funciones desarrolladas para el tratamiento de los datos | | | | |

Tabla 3.22: Historia de usuario 14.

| **ID** | US-15 | **TÍTULO** | Aprendizaje de Dash para la elaboración del dashboard | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Aprender conocimientos de desarrollo de aplicaciones en Dash | | | | |
| Para... | Poder desarrollar el Dashboard | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Aprendizaje de Dash para la elaboración del dashboard | | | | |
| Dado que | El dashboard utilizará el framework Dash | | | | |
| Cuando | Implemente el dashboard o aplicación | | | | |
| Entonces | Deberán aplicarse los conocimientos adquiridos en Dash | | | | |

Tabla 3.23: Historia de usuario 15.

| **ID** | US-16 | **TÍTULO** | Análisis de las funcionalidades dashboard | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Desarrollador | | | | |
| Quiero... | Obtener una visión global del diseño de la aplicación | | | | |
| Para... | Poder empezar el desarrollo | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Análisis de las funcionalidades dashboard | | | | |
| Dado que | Se quiere empezar el desarrollo de la aplicación | | | | |
| Cuando | Se concrete el diseño global de la aplicación | | | | |
| Entonces | Se estudiará la viabilidad de la misma y se empezará a desarrollar el diseño del menú de pestañas que compondrá la aplicación | | | | |

Tabla 3.24: Historia de usuario 16.

| **ID** | US-17 | **TÍTULO** | Redacción del capítulo 3 “Análisis y gestión del proyecto” | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Tener un capítulo en el que explique cómo se ha llevado a cabo la gestión y análisis del proyecto | | | | |
| Para... | Poder entender la metodología utilizada, los riesgos y el presupuesto del mismo | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Redacción del capítulo 3 “Análisis y gestión del proyecto” | | | | |
| Dado que | Leo la memoria | | | | |
| Cuando | Me encuentre en el capítulo 3 | | | | |
| Entonces | Podré tener una explicación de la gestión y análisis que se han llevado a cabo | | | | |

Tabla 3.25: Historia de usuario 17.

| **ID** | US-18 | **TÍTULO** | Carga de un dataset | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Poder buscar o arrastrar un dataset | | | | |
| Para... | Cargar los datos del dataset en la aplicación | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Carga dataset | | | | |
| Dado que | Me encuentro en la pestaña de datos | | | | |
| Cuando | Arrastro o pincho y busco un dataset | | | | |
| Entonces | Se carga la información del dataset seleccionando la primera columna y apareciendo un texto con el nombre del dataset cargado | | | | |

Tabla 3.26: Historia de usuario 18.

| **ID** | US-19 | **TÍTULO** | Visualización y selección de columnas del dataset | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Poder visualizar todas las columnas dataset | | | | |
| Para... | Seleccionar una de las columnas | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Visualización y selección de columnas | | | | |
| Dado que | Me encuentro en la pestaña de datos | | | | |
| Cuando | Cargo un dataset | | | | |
| Entonces | Podré visualizar el listado de las columnas y seleccionar una para visualizarla | | | | |

Tabla 3.27: Historia de usuario 19.

| **ID** | US-20 | **TÍTULO** | Visualización de una tabla y un gráfico con información de una columna | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Poder visualizar una tabla y un gráfico de una de las columnas | | | | |
| Para... | Poder obtener información de los datos de la columna | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Visualización de un gráfico y una tabla de una columna | | | | |
| Dado que | Me encuentro en la pestaña de datos | | | | |
| Cuando | Selecciono una columna y pulso en visualizar | | | | |
| Entonces | Podré visualizar una tabla y un gráfico con información de una columna | | | | |

Tabla 3.28: Historia de usuario 20.

| **ID** | US-21 | **TÍTULO** | Selección de las columnas a utilizar en los algoritmos | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Seleccionar las columnas | | | | |
| Para... | Utilizar en los algoritmos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Selección de columnas | | | | |
| Dado que | Me encuentro en la pestaña de ML | | | | |
| Cuando | Selecciono unas columnas y pulso en seleccionar | | | | |
| Entonces | Aparecerán las opciones para eliminar filas y para seleccionar la columna a predecir | | | | |

Tabla 3.29: Historia de usuario 21.

| **ID** | US-22 | **TÍTULO** | Borrado de filas de una determinada columna | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Seleccionar las filas de una columna | | | | |
| Para... | Eliminar dichas filas | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Eliminación de filas | | | | |
| Dado que | Me encuentro en la pestaña de ML | | | | |
| Cuando | Selecciono una columna y posteriormente me aparecerán los valores de las filas de dicha columna donde podré hacer una selección y pulsar el botón eliminar | | | | |
| Entonces | Aparecerá un texto informando de los elementos borrados | | | | |

Tabla 3.30: Historia de usuario 22.

| **ID** | US-23 | **TÍTULO** | Generación de un dataset con las columnas y las filas modificadas anteriormente | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Obtener un dataset con las columnas y las filas modificadas anteriormente | | | | |
| Para... | Poder disponer del dataset modificado en formato csv | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Creación de un csv con el dataset modificado | | | | |
| Dado que | Me encuentro en la pestaña de ML | | | | |
| Cuando | Selecciono el botón CREAR CSV | | | | |
| Entonces | Aparecerá un link de descarga con el csv creado | | | | |

Tabla 3.31: Historia de usuario 23.

| **ID** | US-24 | **TÍTULO** | Selección y modificación de parámetros del algoritmo a utilizar, además de la elección del porcentaje de división del dataset y la columna a predecir | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Seleccionar el algoritmo a utilizar, modificar sus parámetros, elegir el porcentaje de división del dataset y la columna a predecir | | | | |
| Para... | Pulsar el botón ejecutar y que se procesen los datos según el algoritmo y sus parámetros establecidos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Selección del algoritmo y sus variables. | | | | |
| Dado que | Me encuentro en la pestaña de ML | | | | |
| Cuando | Selecciono un algoritmo | | | | |
| Entonces | Aparecerán los parámetros predefinidos pudiéndose modificar, junto con una descripción de los mismos al poner el ratón sobre el nombre, también se elegirá el porcentaje de división del dataset y la columna a predecir | | | | |

Tabla 3.32: Historia de usuario 24.

| **ID** | US-25 | **TÍTULO** | Visualización los resultados obtenidos por el algoritmo utilizado | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Poder seleccionar un dataset | | | | |
| Para... | Poder eliminarlo | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Eliminar dataframe | | | | |
| Dado que | Me encuentro en la pestaña de Experimentos | | | | |
| Cuando | Selecciono el dataset que quiero borrar y pulso el botón eliminar en la sección datasets | | | | |
| Entonces | Se eliminaran los datos del dataset | | | | |

Tabla 3.33: Historia de usuario 25.

| **ID** | US-26 | **TÍTULO** | Configuración de un nuevo dataset a la pestaña experimentos | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Visualizar los resultados obtenidos y una gráfica con las predicciones | | | | |
| Para... | Poder interpretar los resultados de los algoritmos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Visualización de los resultados | | | | |
| Dado que | Me encuentro en la pestaña de ML | | | | |
| Cuando | Selecciono el botón ejecutar | | | | |
| Entonces | Aparecerán los resultados del algoritmo y un gráfico donde se podrán seleccionar las columnas a visualizar, teniendo como valores predeterminados la columna a predecir y las predicciones | | | | |

Tabla 3.34: Historia de usuario 26.

| **ID** | US-27 | **TÍTULO** | Eliminación de un dataset en la pestaña experimentos | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Añadir varios datasets | | | | |
| Para... | Poder ejecutarlos con diferentes algoritmos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Añadir dataframe | | | | |
| Dado que | Me encuentro en la pestaña de Experimentos | | | | |
| Cuando | Selecciono el botón nuevo en la sección datasets | | | | |
| Entonces | Aparecerán una ventana emergente donde cargar un dataset y seleccionar la columna a predecir y el porcentaje de división | | | | |

Tabla 3.35: Historia de usuario 27.

| ID | US-28 | TÍTULO | Configuración de un algoritmo a la pestaña experimentos | | |
|---|---|---|---|---|---|
| | | PRIORIDAD | | ESTIMACIÓN | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Poder añadir varios algoritmos | | | | |
| Para... | Poder procesarlos con distintos datasets | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Añadir algoritmo | | | | |
| Dado que | Me encuentro en la pestaña de Experimentos | | | | |
| Cuando | Selecciono el botón nuevo en la sección algoritmos | | | | |
| Entonces | Aparecerán una ventana emergente donde seleccionar un algoritmo y modificar sus parámetros, junto con una descripción de los mismos al poner el ratón sobre el nombre | | | | |

Tabla 3.36: Historia de usuario 28.

| ID | US-29 | TÍTULO | Eliminación un algoritmo en la pestaña experimentos | | |
|---|---|---|---|---|---|
| | | PRIORIDAD | | ESTIMACIÓN | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Poder seleccionar un algoritmo | | | | |
| Para... | Poder eliminarlo | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Eliminar algoritmo | | | | |
| Dado que | Me encuentro en la pestaña de Experimentos | | | | |
| Cuando | Selecciono el algoritmo que quiero borrar y pulso el botón eliminar en la sección algoritmos | | | | |
| Entonces | Se eliminaran los datos del algoritmo | | | | |

Tabla 3.37: Historia de usuario 29.

| **ID** | US-30 | **TÍTULO** | Visualización de un listado con los resultados, donde se podrá pinchar sobre uno y ver los resultados detalladamente |
|---|---|---|---|
| | | **PRIORIDAD** | **ESTIMACIÓN** |
| **DEFINICIÓN** | | | |
| Como... | Usuario | | |
| Quiero... | Poder visualizar un listado con los resultados obtenidos y una gráfica con las predicciones | | |
| Para... | Poder interpretar los resultados de los algoritmos | | |
| **CRITERIOS DE ACEPTACIÓN** | | | |
| 1 | Visualización del listado de resultados | | |
| Dado que | Me encuentro en la pestaña de Experimentos | | |
| Cuando | Selecciono el botón ejecutar | | |
| Entonces | Aparecerá un listado con la combinación de los diferentes algoritmos y datasets, donde se podrá seleccionar uno y ver sus resultados, además de un gráfico donde se podrán seleccionar las columnas a visualizar teniendo como valores predeterminados la columna a predecir y las predicciones | | |

Tabla 3.38: Historia de usuario 30.

| **ID** | US-31 | **TÍTULO** | Descripción de los parámetros de los algoritmos |
|---|---|---|---|
| | | **PRIORIDAD** | **ESTIMACIÓN** |
| **DEFINICIÓN** | | | |
| Como... | Usuario | | |
| Quiero... | Poder visualizar una descripción de los parámetros de los algoritmos | | |
| Para... | Poder saber la implicación que tiene cada uno | | |
| **CRITERIOS DE ACEPTACIÓN** | | | |
| 1 | Descripción de los parámetros de los algoritmos | | |
| Dado que | Me encuentro en la pestaña ML o Experimentos | | |
| Cuando | Pongo el ratón sobre el nombre de un parámetro | | |
| Entonces | Aparecerá una descripción de dicho parámetro | | |

Tabla 3.39: Historia de usuario 31.

| **ID** | US-32 | **TÍTULO** | Probar la funcionalidad global del dasboard | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Realizar pruebas sobre la aplicación | | | | |
| Para... | Poder comprobar que funciona correctamente | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Probar la funcionalidad global del dashboard | | | | |
| Dado que | Ejecuto la aplicación | | | | |
| Cuando | Elaboro una serie de pruebas | | | | |
| Entonces | Podré asegurarme de que todas sus funcionalidades están correctamente desarrolladas | | | | |

Tabla 3.40: Historia de usuario 32.

| **ID** | US-33 | **TÍTULO** | Redacción del capítulo 6 “Dashboard” | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Tener un capítulo que explique cómo se ha desarrollado el dashboard | | | | |
| Para... | Poder entender su funcionamiento | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Redacción del capítulo 6 “Dashboard” | | | | |
| Dado que | Leo la memoria | | | | |
| Cuando | Me encuentre en el capítulo 6 | | | | |
| Entonces | Podré tener una explicación del diseño, implementación y pruebas del dashboard | | | | |

Tabla 3.41: Historia de usuario 33.

| **ID** | US-34 | **TÍTULO** | Realización de los experimentos con diferentes datos y algoritmos | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Realizar experimentos con diferentes datos y algoritmos | | | | |
| Para... | Poder ver qué capacidad de predicción tienen los modelos generados | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Experimentación con diferentes datos y algoritmos | | | | |
| Dado que | Realizo diferentes experimentos con varios datos y algoritmos | | | | |
| Cuando | Obtengo los resultados | | | | |
| Entonces | Podré saber que algoritmo se adapta mejor a los datos y que capacidad predictiva tiene el modelo generado | | | | |

Tabla 3.42: Historia de usuario 34.

| **ID** | US-35 | **TÍTULO** | Redacción del capítulo 7 “Resultados” | | |
|---|---|---|---|---|---|
| | | **PRIORIDAD** | | **ESTIMACIÓN** | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Tener un capítulo que muestre los diferentes experimentos realizados | | | | |
| Para... | Poder ver los resultados obtenidos | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Redacción del capítulo 7 “Resultados” | | | | |
| Dado que | Leo la memoria | | | | |
| Cuando | Me encuentre en el capítulo 7 | | | | |
| Entonces | Podré ver los resultados de los diferentes experimentos realizados | | | | |

Tabla 3.43: Historia de usuario 35.

| ID | US-36 | TÍTULO | Redacción del capítulo 8 “Conclusiones y trabajo futuro” | | |
|---|---|---|---|---|---|
| | | PRIORIDAD | | ESTIMACIÓN | |
| **DEFINICIÓN** | | | | | |
| Como... | Usuario | | | | |
| Quiero... | Tener un capítulo que explique las conclusiones y el trabajo futuro | | | | |
| Para... | Poder entender las deducciones a las que nos ha llevado la realización de este proyecto | | | | |
| **CRITERIOS DE ACEPTACIÓN** | | | | | |
| 1 | Redacción del capítulo 5 “Propuesta” | | | | |
| Dado que | Leo la memoria | | | | |
| Cuando | Me encuentre en el capítulo 8 | | | | |
| Entonces | Podré tener una explicación de las conclusiones que hemos sacado acerca de este proyecto y los posibles desarrollos que se podrían llevar a cabo para ampliar y mejorar el mismo | | | | |

Tabla 3.44: Historia de usuario 36.

### 3.2.3. Sprint Backlog

A continuación pasaremos a especificar las historias de usuario realizadas en cada uno de los Sprints en los que está dividido el desarrollo de nuestro proyecto. Ha sido necesario la realización de 6 Sprints, los cuales tendrán una duración de 1 mes cada uno. Podemos apreciar que al usar la metodología Scrum haremos una estimación en puntos de historia. Estos estiman las tareas según su tamaño o dificultad, tomando como referencia una tarea sencilla a la que se le asignará el valor de 1 punto de historia y a partir de esta referencia se determinara el valor en puntos de historia de las demás tareas. En nuestro caso un la correspondencia a un punto de historia sera de 5 a 6 horas, con esta relación seremos capaces de inferir las horas que nos ha llevado realizar cada tarea. En nuestro caso cada historia de usuario solo dará origen a una tarea.

| **Sprint 1** | | | |
|---|---|---|---|
| **Tarea** | **Descripción** | **Horas Estimadas** | **P. Historia** |
| T-01 | Instalación de LaTeX y aprendizaje de conocimientos básicos. | 3 | 0,5 |
| T-02 | Estudio de la gestión del tráfico aéreo. | 9 | 1,5 |
| T-03 | Análisis y documentación de estudios de predicción de tiempos de llegada de vuelos con algoritmos de Machine Learning. | 12 | 2 |
| T-04 | Redacción del capítulo 1 “Introducción”. | 9 | 1,5 |
| T-05 | Redacción del capítulo 2 “Gestión del tráfico aéreo”. | 12 | 2 |
| **Total** | | 45 | 7,5 |
| **Fecha inicio: 01-06-2019** | | **Fecha fin: 01-07-2019** | |

Tabla 3.45: Sprint 1

| **Sprint 2** | | | |
|---|---|---|---|
| **Tarea** | **Descripción** | **Horas Estimadas** | **P. Historia** |
| T-06 | Instalación de Python Anaconda. | 2 | 0,25 |
| T-07 | Instalación, aprendizaje de Spark y posterior documentación. | 15 | 2,5 |
| T-08 | Implementación de preprocesado de datos mediante Spark SQL. | 30 | 5 |
| T-09 | Gestión de riesgos. | 8 | 1,5 |
| **Total** | | 55 | 9,25 |
| **Fecha inicio: 18-09-2019** | | **Fecha fin: 18-10-2019** | |

Tabla 3.46: Sprint 2

| **Sprint 3** | | | |
|---|---|---|---|
| **Tarea** | **Descripción** | **Horas Estimadas** | **P. Historia** |
| T-10 | Estudio de los diferentes parámetros de los algoritmos DT, GBT, y RF en MLlib. | 14 | 2,5 |
| T-11 | Elaboración del capítulo 4 “Marco teórico”. | 18 | 3 |
| T-12 | Estudio e implementación de la transformación de características con MLlib. | 8 | 1,5 |
| T-13 | Configuración de los algoritmos: Decision tree, Gradient-boosted Tree y Random forest. | 17 | 3 |
| **Total** | | 57 | 10 |
| **Fecha inicio: 19-10-2019** | | **Fecha fin: 19-11-2019** | |

Tabla 3.47: Sprint 3

| **Sprint 4** | | | |
|---|---|---|---|
| **Tarea** | **Descripción** | **Horas Estimadas** | **P. Historia** |
| T-14 | Redacción del capítulo 5 “Propuesta”. | 8 | 1,5 |
| T-15 | Aprendizaje de Dash para la elaboración del dashboard. | 12 | 2 |
| T-16 | Análisis de las funcionalidades del dashboard. | 11 | 2 |
| T-17 | Redacción del capítulo 3 “Análisis y gestión del proyecto”. | 15 | 2,5 |
| T-18 | Carga de un dataset. | 6 | 1 |
| T-19 | Visualización y selección de columnas del dataset. | 3 | 0,5 |
| T-20 | Visualización de una tabla y un gráfico con información de una columna. | 8 | 1,5 |
| **Total** | | 63 | 11 |
| **Fecha inicio: 20-11-2019** | | **Fecha fin: 20-12-2019** | |

Tabla 3.48: Sprint 4

| **Sprint 5** | | | |
|---|---|---|---|
| **Tarea** | **Descripción** | **Horas Estimadas** | **P. Historia** |
| T-21 | Selección de las columnas a utilizar en los algoritmos. | 3 | 0,5 |
| T-22 | Borrado de filas de una determinada columna. | 3 | 0,5 |
| T-23 | Generación de un dataset con las columnas y las filas modificadas anteriormente. | 13 | 2,25 |
| T-24 | Selección y modificación de parámetros del algoritmo a utilizar, además de la elección del porcentaje de división del dataset y la columna a predecir. | 12 | 2 |
| T-25 | Visualización los resultados obtenidos por el algoritmo utilizado. | 6 | 1 |
| T-26 | Configuración de un nuevo dataset a la pestaña experimentos. | 3 | 0,75 |
| T-27 | Eliminación de un dataset en la pestaña experimentos. | 9 | 1,75 |
| T-28 | Configuración de un algoritmo a la pestaña experimentos. | 3 | 0,5 |
| T-29 | Eliminación un algoritmo en la pestaña experimentos. | 3 | 0,5 |
| T-30 | Visualización de un listado con los resultados, donde se podrá pinchar sobre uno y ver los resultados detalladamente. | 11 | 2 |
| T-31 | Descripción de los parámetros de los algoritmos. | 2 | 0,25 |
| **Total** | | 68 | 12 |
| **Fecha inicio: 08-01-2020** | | **Fecha fin: 08-02-2020** | |

Tabla 3.49: Sprint 5

| Sprint 6 | | | |
|---|---|---|---|
| **Tarea** | **Descripción** | **Horas Estimadas** | **P. Historia** |
| T-32 | Probar la funcionalidad global del dasboard. | 6 | 1 |
| T-33 | Redacción del capítulo 6 “Dashboard”. | 14 | 2,5 |
| T-34 | Realización de los experimentos con diferentes datos y algoritmos. | 18 | 3 |
| T-35 | Redacción del capítulo 7 “Resultados”. | 11 | 2 |
| T-36 | Redacción del capítulo 8 “Conclusiones y trabajo futuro”. | 11 | 2 |
| **Total** | | 60 | 10,5 |
| **Fecha inicio: 09-02-2020** | | **Fecha fin: 09-03-2020** | |

Tabla 3.50: Sprint 6

### 3.2.4. Gestión de Riesgos

El desarrollo de un proyecto software puede ser afectado por una gran cantidad de riesgos, pudiendo afectar al orden cronológico de desarrollo del mismo. Esto hace que sea de vital importancia identificar y analizar los posibles riesgos con el fin de afrontarlos teniendo el menor impacto posible. Establecer correctamente todos los riesgos es una tarea con una elevada dificultad, ya que identificar completamente todos los riesgos que pueden afectar a un proyecto es prácticamente imposible. Para ayudarnos a llevar acabo una buena gestión de los riesgos se elaborará un plan donde se definirán los siguientes aspectos:

- Técnicas y herramientas a utilizar.
- Actividades de control de riesgos y periodicidad de las mismas.
- Plantillas estandarizadas para la identificación y gestión de riesgos.
- Un organigrama para la gestión de riesgos.
- El proceso de identificación y análisis de riesgos.

En este proyecto solo se tendrán en cuenta algunos aspectos de los anteriormente citados. En la Tabla 3.51 se indican los posibles riesgos que pueden afectar a este proyecto.

| ID | Título |
|---|---|
| RISK-01 | Planificación optimista. |
| RISK-02 | Desconocimiento de las tecnologías utilizadas. |
| RISK-03 | Cambios en los requisitos. |
| RISK-04 | Problemas con el SO utilizado. |
| RISK-05 | Retrasos en la planificación debido a problemas de salud. |

Tabla 3.51: Listado de riesgos

Después de haber identificado los riesgos que pueden afectar al proyecto, pasaremos a cuantificar el impacto que podría tener cada uno de ellos en caso de que se produjeran. Para esto haremos uso de la herramienta denominada Matriz de Probabilidad Impacto, la cual prioriza los riesgos en función de la probabilidad de que ocurran y del impacto que generarían. En la Tabla 3.52 mostramos nuestros riesgos priorizados por la Matriz de Probabilidad Impacto.

| ID | Título | Impacto | Probabilidad | Prioridad |
|---|---|---|---|---|
| RISK-01 | Planificación optimista. | Alto | Media | 2 |
| RISK-02 | Desconocimiento de las tecnologías utilizadas. | Alto | Alto | 1 |
| RISK-03 | Cambios en los requisitos. | Medio | Baja | 3 |
| RISK-04 | Problemas con SO utilizado. | Medio | Baja | 5 |
| RISK-05 | Retrasos en la planificación debido a problemas de salud. | Alto | Baja | 4 |

Tabla 3.52: Listado de riesgos priorizados

Después de haber priorizado los riesgos, lo siguiente que haremos es llevar a cabo medidas para permitirnos gestionar los riesgos. Realizaremos dos tipos de planes:

- Plan de mitigación: Definida en el PMBOK como el conjunto de acciones a través de las cuales se pretende reducir la probabilidad de ocurrencia de un riesgo.
- Plan de contingencia: Conjunto de acciones realizadas como respuesta a la producción de un riesgo. Es el ejemplo perfecto de un plan reactivo de gestión de riesgos, es decir, trata de reducir el impacto del riesgo una vez este ya ha ocurrido.

En las siguientes tablas describiremos los planes definidos anteriormente, para cada uno de nuestros riesgos.

| ID | Título | Descripción |
|---|---|---|
| RISK-01 | Planificación optimista. | La planificación en muchos casos puede ser optimista, lo que puede provocar tardar más de lo previsto en realizar las tareas, generando retrasos en la planificación. |
| **Gestión del riesgo** | | |
| Estrategia | | Intentar prevenir el riesgo. |
| Plan de mitigación | | Con razón de evitar problemas derivados de los retrasos haremos una planificación pesimista dejando 4 días de holgura en cada sprint. |
| Plan de contingencia | | Aumentar el número de horas diarias de trabajo. |

Tabla 3.53: Risk-01. Planificación optimista.

| ID | Título | Descripción |
|---|---|---|
| RISK-02 | Desconocimiento de las tecnologías utilizadas. | En este proyecto se hará uso de tecnologías que no han sido vistas previamente en la carrera, pudiendo surgir problemas desconocidos. |
| **Gestión del riesgo** | | |
| Estrategia | | Elaborar una estrategia para lograr evitar el riesgo. |
| Plan de mitigación | | Con el fin de familiarizarnos con las nuevas tecnologías a utilizar, los primeros sprints se dedicaran a documentarnos para adquirir los conocimientos necesarios. |
| Plan de contingencia | | Se hará un incremento de las horas diarias, ademas de contactar con los tutores del proyecto en busca de ayuda. |

Tabla 3.54: Risk-02. Desconocimiento de las tecnologías utilizadas.

| ID | Título | Descripción |
|---|---|---|
| RISK-04 | Cambios en los requisitos. | Al tratarse de un proyecto de investigación pueden surgir nuevos casos de estudio, teniendo que modificar los requisitos iniciales. |
| **Gestión del riesgo** | | |
| Estrategia | | Elaborar una estrategia para lograr evitar el riesgo. |
| Plan de mitigación | | La medida de prevención utilizada es el uso de una metodología ágil, que como hemos dicho anteriormente tiene una gran adaptabilidad a cambios. |
| Plan de contingencia | | Cambiar los requisitos del proyecto que sean necesarios. |

Tabla 3.55: Risk-03. Cambios en los requisitos.

| ID | Título | Descripción |
|---|---|---|
| RISK-04 | Problemas con el SO utilizado. | El SO puede fallar en cualquier momento debido a problemas con las actualizaciones, entre otras cosas. |
| **Gestión del riesgo** | | |
| Estrategia | | Elaborar una estrategia para lograr evitar el riesgo. |
| Plan de mitigación | | Para prevenir la perdida de información se almacenara toda la información en la nube, teniendo la posibilidad de recuperarla en caso de pérdida. |
| Plan de contingencia | | Aumentar el número de horas diarias, con el fin de recuperar el tiempo perdido en la reinstalación del SO y programas. |

Tabla 3.56: Risk-04. Problemas con el SO utilizado.

| ID | Título | Descripción |
|---|---|---|
| RISK-05 | Retrasos en la planificación debido a problemas de salud. | Las personas que desarrollan el proyecto pueden padecer problemas de salud, llegando a derivar en una baja. |
| **Gestión del riesgo** | | |
| Estrategia | | En este caso aceptaremos el riesgo, ya que es poco probable que en el corto periodo de tiempo se desarrolla nuestro proyecto ocurra alguna baja. |
| Plan de mitigación | | Ante la imposibilidad de predecir futuros problemas de salud, lo único que podemos hacer es aconsejar tomar precauciones e ir a revisiones médicas. |
| Plan de contingencia | | Adaptar la planificación a los retrasos producidos por problemas de salud. |

Tabla 3.57: Risk-05. Retrasos en la planificación debido a problemas de salud.

## 3.3. Planificación

La planificación de este proyecto se ha elaborado utilizando la metodología ágil Scrum, la cual ha sido descrita en la sección 3.1. El desarrollo del proyecto ha sido dividido en el 6 Sprints de 1 mes de duración cada uno. Las tareas realizadas en cada sprint están detalladas en el Sprint Backlog ubicado en la sección 3.2 de análisis.

Este proyecto al estar centrado en investigación, ha provocado que muchas de las tareas estén centradas en el aprendizaje de nuevos entornos, como el de la aviación y Machine Learning, y también de nuevas tecnologías como Spark. En el primer sprint nos encontramos con que la mayoría de las tareas consisten en la investigación del espacio aéreo y búsqueda de estudios similares al que queremos realizar, con la posterior documentación en la memoria. En el segundo y tercer sprint seguimos encontrándonos tareas de aprendizaje e investigación, esta vez de la tecnología a utilizar, lo que da lugar a las primeras fases de implementación de algoritmos. En el sprint 4 continúan las tareas de implementación y se empieza el análisis del dashboard y su implementación, que se concluirá en el quinto sprint. Por último en el sprint 6 se realizan varias pruebas del dashboard desarrollado, se ejecutaran una serie de experimentos, analizando los diferentes resultados obtenidos, y se terminaran de redactar los últimos capítulos de la memoria.

Respecto a los retrasos en la planificación cabe destacar que se ha dotado a todos los sprints de la holgura necesaria en caso de que se produzca una planificación optimista. Aun así en el quinto sprint se produjo un fallo en el sistema operativo, siendo necesario su reinstalación y la de los programas necesarios. Este riesgo ya se había previsto en la gestión de riesgos, presente

en la sección 3.2 de análisis. Este hecho genero un retraso aproximado de 6 horas que se aplacó añadiendo horas extras. En el siguiente cuadro podemos ver un resumen de las horas y puntos de historia de los sprints.

| Sprint | Duración horas | Puntos de historia | Retraso horas | Retraso puntos |
|---|---|---|---|---|
| 1 | 45 | 7,5 | 0 | 0 |
| 2 | 55 | 9,25 | 0 | 0 |
| 3 | 57 | 10 | 0 | 0 |
| 4 | 63 | 11 | 0 | 0 |
| 5 | 68 | 12 | 6 | 1 |
| 6 | 60 | 10,5 | 0 | 0 |

Tabla 3.58: Planificación temporal de los sprints.

En Figura 3.2 se muestra el gráfico de los puntos de historia por cada sprint. En este gráfico podemos observar como la carga de los sprints varía entre 7,5 y 12 puntos de historia. La carga de trabajo va aumentando a medida que los sprints avanzan, teniendo la mayor carga de trabajo en el sprint 5, el cual se dedicó en su totalidad al desallorro del dashboard.

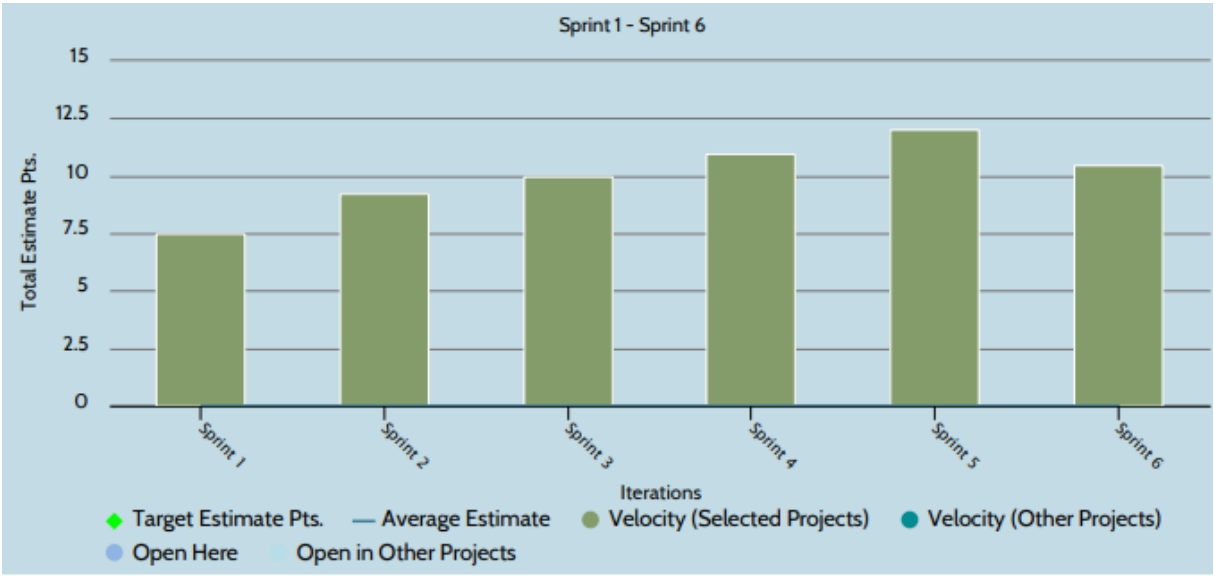



Figura 3.2: Puntos de historia por sprint.

## 3.4. Presupuesto

En este proyecto, al ser de investigación, será muy difícil elaborar un presupuesto inicial. A consecuencia de esto pasaremos a detallar el presupuesto final que hemos obtenido como resultado al acabar nuestro proyecto. Estará dividido en tres categorías: hardware, software, y mano de obra.

- **Presupuesto Software.** En la Tabla 3.59 tenemos los costes de los diferentes productos software utilizados. Estos costes estarán prorrateados en función del tiempo de uso.

| Software | Coste Total | Vida Útil | % de uso | Coste real |
|---|---|---|---|---|
| Windows 10 | 145€ | 10 años | 5 % | 7,25€ |
| Anaconda Python | 0 € | | | 0 € |
| Apache Spark | 0€ | | | 0€ |
| OneDrive | 0€ | | | 0€ |
| Google Chrome | 0€ | | | 0€ |
| Java SE | 0€ | | | 0€ |
| TexStudio | 0€ | | | 0€ |
| VersionOne | 0€ | | | 0€ |

Tabla 3.59: Coste Software

- **Presupuesto Hardware.** En la Tabla 3.60 se destallan los componentes hardware y su coste asociado.

| Hardware | Coste Total | Vida Útil | % de uso | Coste real |
|---|---|---|---|---|
| Ordenador personal | 800 € | 5 años | 10 % | 80€ |
| Conexión a Internet | 60 €/mes | 6 meses | 100 % | 360 € |

Tabla 3.60: Coste componentes Hardware

- **Presupuesto de mano de obra.** Los costes asociados a la mano de obra de un ingeniero informático están compuestos por:
  - 70 % de sueldo bruto, el cual es rondará los 21.000 €.

- 20 % de Seguridad Social, corresponderá a 6.000 €.
- 10 % de indemnización, pestación social o gasto en formación, corresponderá a 3.000 €.

Después tendremos que calcular el gasto total en función de las horas invertidas en el proyecto, para lo que haremos un cálculo de los euros por hora que cuesta tener contratado a un ingeniero informático y lo multiplicaremos por las horas invertidas en el proyecto.

$$30.000\ \textit{euros/año} \longrightarrow 14,82\ euros/hora$$

$$Sueldototal = precio/hora * horas \longrightarrow 14,82\ euros/hora * 348horas$$

El sueldo total a percibir por la persona encargada de desarrollar el proyecto es: 5.158,1 €

Teniendo el cálculo de todos los costes asociados al proyecto, pasaremos a hacer el cálculo del presupuesto final, que se puede observar en la Tabla 3.61.

| Coste software | Coste hardware | Coste personal | Coste total |
|---|---|---|---|
| 7,25 € | 440 € | 5.158,1 € | 5.605,35 € |

Tabla 3.61: Presupuesto final.

# Capítulo 4

# Marco teórico

En este capítulo nos centraremos primeramente en describir los diferentes algoritmos de Machine Learning utilizados para la predicción del tiempo de llegada de un conjunto de vuelos. También se detallaran los métodos de evaluación de los modelos generados por los algoritmos, así como las métricas utilizadas para medir los resultados. Como segundo punto relevante de este capítulo, tendremos la descripción de la tecnología utilizada para el tratamiento de los datos y su posterior ejecución de los algoritmos.

## 4.1. Algoritmos de predicción

En esta sección se detallaran el funcionamiento de los diferentes algoritmos regresivos de los que haremos uso para la predicción de los tiempos de aterrizaje de una serie de vuelos. En este proyecto se han utilizado algoritmos basados en arboles de decisión, ya que según estudios anteriormente realizados son los que mejor se adaptan a nuestra problemática. Más concretamente se han utilizado los siguientes algoritmos: Decision Tree o Árbol de Decisión, Random Forest y Gradient-boosted Trees.

### 4.1.1. Árbol de decisión

Los arboles de decisión son utilizados en el análisis de datos como modelos predictivos de aprendizaje supervisado. Cuando la variable a predecir es un conjunto de valores finitos estaremos hablando de árboles de clasificación, pero si por el contrario se trata de un conjunto de valores continuos se denominaran arboles de regresión. En nuestro caso utilizaremos un árbol de regresión, al tratarse de predicción de tiempos de llegada de vuelos.

El funcionamiento de un árbol de decisión consiste en partir de un nodo raíz, que representa una característica, e ir evaluando los posibles valores de ese nodo, dando lugar a otros nodos internos que representarán otra característica de los datos, o a nodos hoja que corresponderán a uno de los valores de la característica a predecir. En la Figura 4.1 se muestra un ejemplo en el

que partimos de las características cielo, humedad y viento para determinar si salimos a hacer deporte.

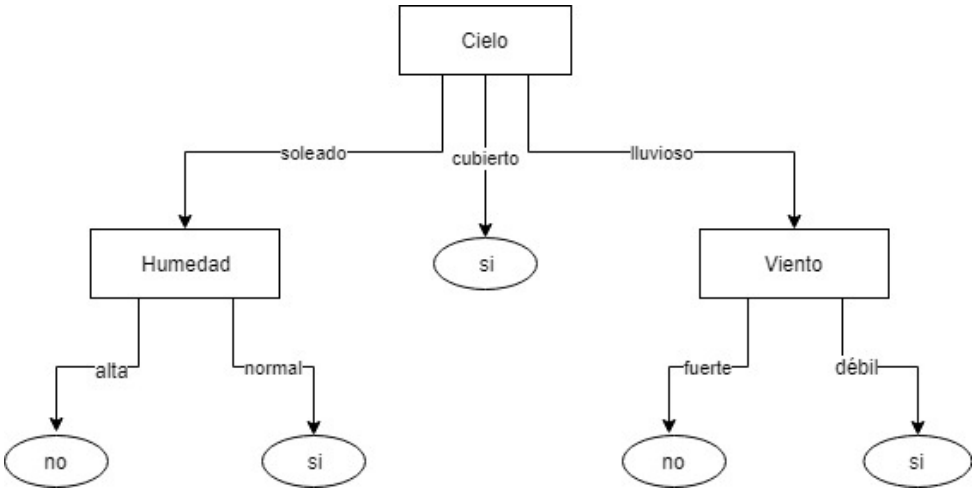



Figura 4.1: Ejemplo de árbol de decisión.

El algoritmo del árbol realiza recursivamente una partición binaria, si la característica es continua, o de cada clase, si la variable es categórica. El árbol predice la misma etiqueta para cada partición inferior o nodo hoja. Para las variables continuas se escoge como partición la mejor división entre el posible conjunto de divisiones, con el objetivo de aumentar la ganancia de información del nodo de un árbol. En otras palabras, la división escogida para cada nodo se elige del conjunto *argmaxIG(D,s)* donde *IG(D,s)* es la información ganada cuando la división *s* es aplicada al dataset *D*.

### Ganancia de información e impureza del nodo

La impureza del nodo es una forma de medir la homogeneidad de las etiquetas en el nodo. Tenemos dos medidas de impureza para la clasificación (Gini y entropía) y una medida de impureza para la regresión (varianza). En nuestro caso la impureza del nodo estará representada por la varianza, que tiene la siguiente fórmula:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)^2$$

donde $y_i$ es el valor resultado (label) de la observación, $N$ es el número de observaciones y $\mu$ es la media dada por $\frac{1}{N}\sum_{i=1}^{N} y_i$ .

La ganancia de información es la diferencia entre la impureza del nodo padre y la suma ponderada de las dos impurezas de los nodos hijos.

**Reglas de parada**

El algoritmo de construcción de árbol se detiene si se dan alguna de estas tres condiciones:

- Si se llega a la profundidad máxima especificada.
- Si la división de candidatos no aporta una ganancia de información suficiente.
- Ningún candidato produce al menos un mínimo de nodos establecido.

**Sobreajuste**

El sobreajuste u overfitting se da cuando un modelo se ajusta demasiado a los datos de entrenamiento, haciendo que se reduzca la capacidad predicativa del modelo ante nuevos datos. Además el hecho de que los datos contengan valores atípicos hace que el modelo pueda tomarlos como referencia empeorando aún más su capacidad predictiva.

En el caso de los arboles su facilidad para ramificarse adquiriendo estructuras complejas hacen que termine ajustándose a los datos de entrenamiento generando un nodo terminal por cada observación. Para impedir que esto suceda será necesario controlar el tamaño del árbol, ajustando los parámetros mencionados anteriormente como reglas de parada.

### 4.1.2. Random Forest

Random Forest es uno de los modelos de aprendizaje automático más exitosos, tanto en problemas de clasificación como de regresión. Utiliza un conjunto de árboles de decisión que son entrenados de manera individual y ejecutados en paralelo. En el proceso de entrenamiento los datos son sometidos a un submuestreo en cada iteración para conseguir un conjunto de datos diferente, además de considerar diferentes subconjuntos aleatorios de características para dividir en cada nodo del árbol, después de esto los árboles son entrenados de la misma manera que los de decisión. Obteniendo como resultado un modelo final formado por la suma de los modelos simples. La combinación de las predicciones de cada árbol reduce la varianza de las predicciones, mejorando el rendimiento en los datos de prueba.

Cada árbol es construido usando el siguiente algoritmo:

1. Sea N el número de casos de prueba, M es el número de variables en el clasificador.
2. Sea m el número de variables de entrada a ser usado para determinar la decisión en un nodo dado; m debe ser mucho menor que M.
3. Elegir un conjunto de entrenamiento para este árbol y usar el resto de los casos de prueba para estimar el error.
4. Para cada nodo del árbol, elegir aleatoriamente m variables en las cuales basar la decisión. Calcular la mejor partición del conjunto de entrenamiento a partir de las m variables.

Como hemos dicho anteriormente las predicciones del modelo se basan en la suma de los árboles de decisión, siendo diferente para los casos de clasificación y regresión. Mientras que en clasificación la variable predicha es la que más votos obtiene, entendiendo como votos los resultados de cada árbol independiente, en regresión la predicción es el promedio de los valores predichos por los arboles individuales.

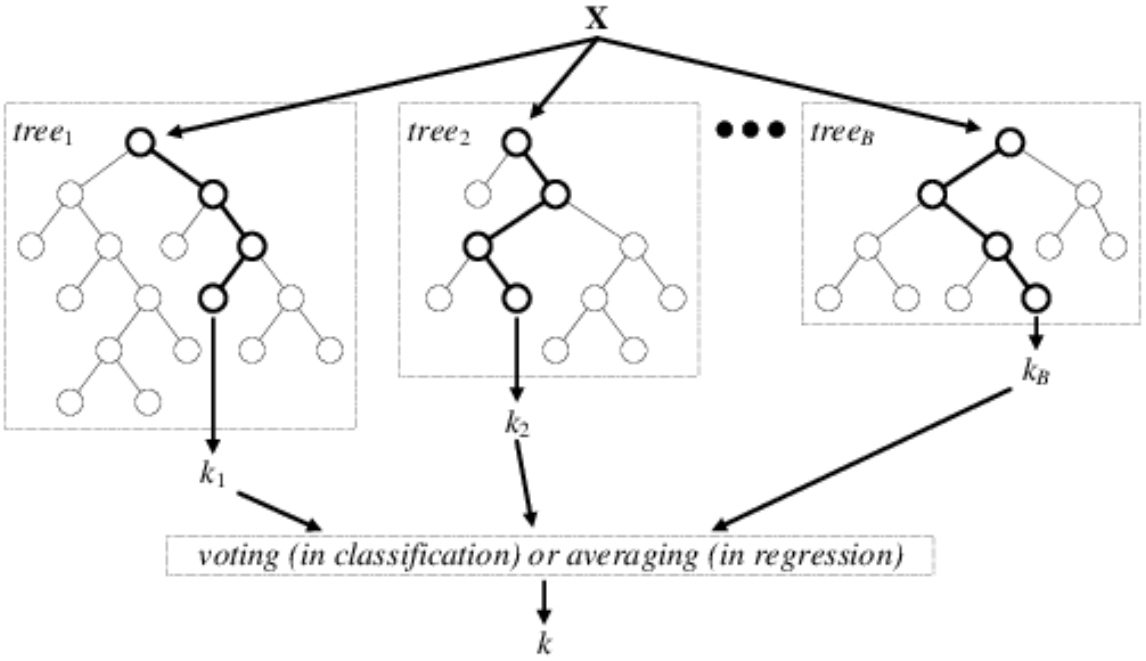



Figura 4.2: Ejemplo de Random Forest [14].

### 4.1.3. Gradient-boosted Trees

Este algoritmo como los anteriores también es utilizado por problemas tanto de clasificación como de regresión. Su funcionamiento está basado en generar numerosos modelos de árboles de decisión utilizando un proceso iterativo y secuencial. Para cada nuevo modelo que se va a entrenar se le asigna un peso mayor a las muestras mal clasificadas por el anterior, así en la siguiente iteración se ayudara al árbol de decisión a corregir los errores previos. Para esto se utiliza una función de perdida que se optimiza, eligiendo en cada paso el árbol que más reduce, produciendo un gradiente descendiente. El entrenamiento del modelo se para cuándo se alcanza un nivel aceptable de la función de pérdida o esta no mejora al aplicarse un conjunto externo de datos. Para modelos de regresión, como es nuestro caso, se pueden utilizar estas dos funciones de pérdida:

- Error cuadrático L2 o MSE (se explica en el apartado 4.3.1)

$$\sum_{i=1}^{N}(y_i - F(x_i))^2$$

- Error absoluto L1 o MAE (se explica en el apartado 4.3.3)

$$\sum_{i=1}^{N} |y_i - F(x_i)|$$

Notación: $N$ = número de instancias, $y_i$ = etiqueta de la instancia $i$, $x_i$ = características de la instancia $i$, $F(x_i)$ = etiqueta predicha por el modelo para la instancia $i$.

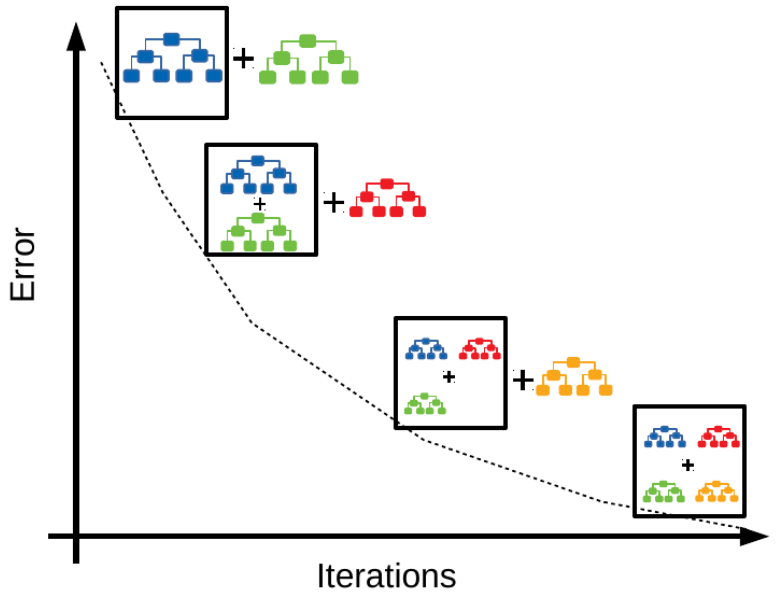



Figura 4.3: Funcionamiento de GBT [27].

El algoritmo Gradient-Boosted puede presentar problemas de sobreajuste, para evitarlo se debe restringir el crecimiento de los árboles y agregar un mayor número de iteraciones.

## 4.2. Métodos de evaluación del modelo: Cross-Validation

Los algoritmos anteriormente descritos se someterán a un entrenamiento con el fin de generar un modelo. La técnica Cross-Validation o validación cruzada es una técnica utilizada para estimar la precisión del modelo que luego se llevara a cabo en la práctica. Esta técnica es ampliamente utilizada en proyectos inteligencia artificial.

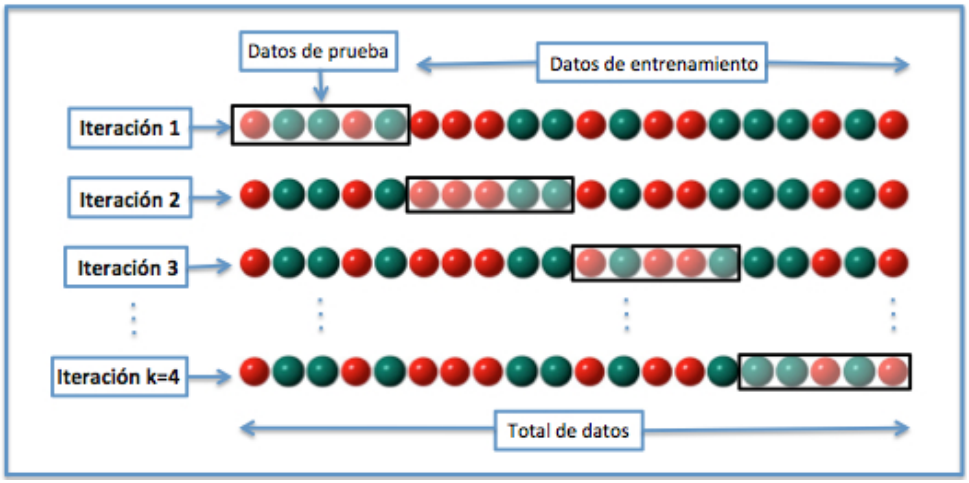



Figura 4.4: K-fold cross-validation con K= 4 [28].

El objetivo de la validación cruzada es evaluar un modelo con diferentes parámetros, de tal manera que en el proceso de ajuste se optimicen los parámetros haciendo que el modelo se adapte a los datos de entrenamiento. Existen varios tipos de validación cruzada, a nosotros nos interesa la validación cruzada de K iteraciones o K-fold cross-validation. Como se muestra en la figura los datos son divididos en K subconjuntos, uno de ellos es utilizado como datos de prueba y el resto (K-1) para entrenar el modelo. Este proceso se repite para cada uno de los subconjuntos, para finalmente realizar una media aritmética de los resultados de cada iteración para obtener el resultado final.

## 4.3. Métricas de regresión

A continuación pasaremos a nombrar las distintas métricas utilizadas para la evaluación de nuestros modelos regresivos.

### 4.3.1. Error cuadrático medio (MSE)

Se encarga de medir el error cuadrado promedio de las predicciones. Calcula la diferencia cuadrada entre las predicciones y el objetivo, para luego promediar esos valores. Está definido por la siguiente ecuación:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

donde $y_i$ es el valor resultado real esperado y $\hat{y}_i$ es la predicción del modelo.

El valor resultante nunca es negativo, ya que se realiza el cuadrado de los errores. En caso de que un modelo sea perfecto nos daría como resultado un 0, según se vaya incrementando el

resultado peor será el modelo de predicción.

- **Ventaja**: Es útil en caso de tener valores inesperados que podrían ser de interés.
- **Desventaja**: En caso de tener datos ruidosos podemos tener un MSE alto a pesar de que el modelo sea bueno, esto es debido a que al elevar el error al cuadrado este empeora más y puede sesgar la métrica sobrestimando la maldad del modelo. Por otro lado, si los errores son menores a 1, se puede producir el efecto contrario, subestimando así la maldad del modelo.

### 4.3.2. Raíz del error cuadrático medio (RMSE)

El error RMSE se obtiene calculando la raíz cuadrada de MSE. El uso de la raíz cuadrada es debido a que se intenta conseguir que la escala de errores sea igual que la escala de los objetivos. Está definido por la siguiente ecuación:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

donde $y_i$ es el valor resultado real esperado y $\hat{y}_i$ es la predicción del modelo.

El RMSE otorga un peso relativamente alto a los errores grandes. Esto significa que el RMSE debería ser más útil cuando los errores grandes son particularmente indeseables.

### 4.3.3. Error absoluto medio (MAE)

Este error calcula el promedio de los valores absolutos de las diferencias de los valores objetivo y las predicciones. Es una puntuación lineal, lo que significa que la diferencia entre 10 y 0 será el doble de la diferencia entre 5 y 0. Se calcula utilizando la fórmula:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

donde $y_i$ es el valor resultado real esperado y $\hat{y}_i$ es la predicción del modelo.

Una ventaja de MAE respecto a MSE es que no es tan sensible a valores atípicos, lo que significa que penaliza mejor los errores grandes.

### 4.3.4. R al cuadrado ($R^2$)

$R^2$ o coeficiente de determinación mide si nuestro modelo es mejor comparándolo con una línea base constante. Esto hace que a diferencia de las anteriores esté libre de escala, dando como resultado normalmente valores entre 0 y 1. Viene definido por la siguiente formula:

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)}$$

La MSE del modelo es calculada de la forma dicha anteriormente, mientras que la MSE de la linea de base se obtiene del modelo más simple posible.

El modelo más simple posible será predecir el promedio de todas las muestras. Si el resultado es cercano a 1 implica que el modelo tiene un error cercano a 0, mientras que si el valor se acerca a 0 querrá decir que el modelo se arroja resultados cercanos a los obtenidos por la línea de base.

## 4.4. Spark y MLlib

Actualmente ha surgido la necesidad de trabajar con grandes cantidades de datos, lo que ha dado lugar a la creación de nuevas tecnologías que nos permitan trabajar con ellos. Spark es desarrollado en el AMPLab de la Universidad Berkeley en 2009, un año después fue liberado como código abierto. El proyecto se hizo notorio en 2013 cuando fue donado a la Apache Software Foundation, para convertirse un año después en un Top-Level Apache Project.

Apache Spark es un framework creado para mejorar la velocidad y rendimiento de las aplicaciones Big Data. Se observó que el modelo de programación de MapReduce era ineficiente para procesos de algoritmos interactivos o consultas interactivas, por lo que Spark se diseñó con el objetivo de dar soporte para persistencia en memoria y un eficiente sistema de tolerancia a fallos. Ademas también se buscó dar soporte en el mismo entorno de ejecución a aplicaciones que requerían de diversos y separados sistemas distribuidos [29].

En comparación con Hadoop, Spark agiliza mucho la creación de proyectos Big Data, ya que la inclusión de tecnologías Hadoop requería centenares de líneas de código. Esto se debe en parte a que Hadoop está escrito en Java, mientas que Spark está escrito en Scala, que es un lenguaje mucho más conciso. Otra gran ventaja de Spark frente a Hadoop es la velocidad, esto se debe a que Spark ejecuta sus procesos en memoria principal mientras que Hadoop lo hace en disco. Pero esto hace que Spark necesite más memoria para el almacenamiento y funcione peor con aplicaciones pesadas.

La usabilidad de Spark es otra ventaja frente a Hadoop, ya que para el manejo de esta última es necesario tener un nivel avanzado de MapReduce o Java. Spark no tiene este problema al ofrecer APIs para la programación en Scala, Java, Python y Spark SQL. También se pueden

escribir comentarios instantáneos sobre consultas u otras acciones, siendo de gran ayuda tanto para los desarrolladores como para los usuarios [32].

El proyecto Apache Spark está compuesto por los diferentes módulos que vamos a describir a continuación, en la Figura 4.5 se muestra un esquema de los diferentes módulos de Spark.

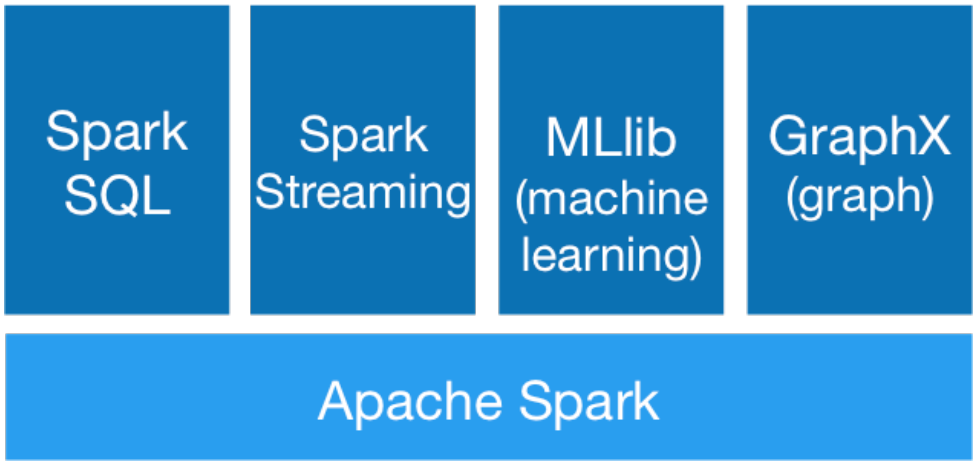



Figura 4.5: Componentes de Spark [31].

### Spark Core

Contiene la funcionalidad principal de Spark, es el encargado de la planificación de tareas, gestión de memoria, recuperación ante fallos, gestión de almacenamiento, etc. Sobre el Spark Core se asientan el resto de elementos, y es donde reside el API de los RDD.

### Spark Streaming

Componente encargado de dar respuesta al procesamiento de datos en tiempo real de forma escalable, con alto rendimiento y tolerancia a fallos. Su funcionamiento consiste en tomar un flujo de datos continuo, y lo convierte en DStream.


Figura 4.6: Funcionamiento de Spark Streaming.

DStream es una abstracción proporcionada por Spark Streaming que representa una secuencia de RDDs ordenados en el tiempo en los que cada uno de ellos guarda datos de un intervalo concreto. Con esto se consigue que Spark Core(Spark Engine) lo analice sin enterarse de que está procesando un flujo de datos, ya que saprk Streaming es el encargado crear y conccordinar estos RDDs [30].

#### Spark SQL

Los datos estructurados, que son información etiquetada y organizada en filas y columnas, están presentes en gran cantidad de bases de datos. Spark SQL permite trabajar con este tipo de datos a través de DataFrames. Los DataFrames son conceptualmente equivalentes a las tablas de una base de datos relacional, y pueden extraer la información de diferentes fuentes de datos como JDBC, ficheros csv, JSON, etc. Los datos también pueden provenir de un RDD existente.

La gran ventaja es que este módulo nos permite unificar información distribuida en diferentes entornos, además de proporcionarnos un alto nivel de abstracción mediante lenguaje SQL [31]. Este módulo también nos permite hacer operaciones sobre los datos sin utilizar lenguaje SQL, ofreciéndonos métodos como *groupBy(), select()*, o *show()* entre otras.

#### Spark GraphX

GraphX está destinado a la manipulación de grafos. Nos permite realizar operaciones distribuidas sobre ellos, como subgraph, joinVertices, y mapReduceTriplets. Ademas incluye una librería con los algoritmos típicos de los grafos. Al igual que los anteriores componentes GraphX también hace uso de los RDD. Dentro de las operaciones que podemos realizar esta la creación de un grafo dirigido con propiedades arbitrarias que enlazan a cada uno de sus vértices o arcos [29].

### 4.4.1. MLlib

MLlib es una librearía destinada al Machine Learning, nos da la posibilidad de implementar y entrenar modelos de clasificación y regresión, clusterización y filtrado colaborativo. A parte de eso nos ofrece herramientas tanto para modelar las características con las que vamos a trabajar, como para evaluar un modelo mediante Cross-Validation. Nos la posibilidad de crear "pipelines" que sirven como flujo de trabajo donde se establecen todos los pasos a realizar con los datos. En cuanto a la persistencia podemos guardar algoritmos, modelos y "pipelines".

MLlib implemneta los siguientes algoritmos:

- Clasificación: Logistic regression (Binomial logistic regression y Multinomial logistic regression), Decision tree, Random forest, Gradient-boosted Tree, Multilayer perceptron, Linear Support Vector Machine, One-vs-Rest (a.k.a. One-vs-All), y Naive Bayes.
- Regresión: Linear regression, Generalized linear regression, Decision tree, Random forest, Gradient-boosted Tree, Survival regression, y Isotonic regression.
- Clúster: K-means, LDA, Bisecting k-means y GMM.
- Filtrado colaborativo: ALS.

MLlib incluye una API basada en RDD y otra basada en DataFrame. Actualmente la API basada en RDD no agregará nuevas características, manteniéndose solo en uso hasta que la API basada en DataFrame alcance la paridad de características con la API basada en RDD, una vez

que suceda esto la API basada en RDD será eliminada. Spark ML se utiliza para referirse a la librería MLlib basada DataFrame API, que esta implementada dentro del módulo Saprk SQL citado anteriormente.

Un DataFrame es un Spark Dataset, es decir una colección de datos distribuida organizada en columnas con nombre representando variables. Son conceptualmente equivalentes a una tabla de una base de datos relacional o un dataframe en R o Python, pero con un conjunto de optimizaciones implícitas [33]. Los principales beneficios que tiene utilizar la DataFrame API son:

- API uniforme: no es necesario adaptarse a las diferencias de cada lenguaje de programación.
- Spark SQL: nos permite acceder y manipular los datos a través de consultas SQL.
- Optimizaciones: implementación de una serie de optimizaciones que hacen que el Dataset nos proporcione un mayor rendimiento al manejar los datos.
- Fuentes de datos: se puede construir un Dataset a partir de bases de datos externas, RDD existentes, archivos CSV, JSON y multitud de datos estructurados.

**Parámetros de los arboles de decisión en MLlib**

En nuestro proyecto vamos a hacer uso de los algoritmos Decision tree, Random forest, Gradient-boosted Tree, los cuales hemos descrito anteriormente. MLlib nos brinda una serie de parámetros que nombraremos a continuación [34][35][36].

- **Profundidad máxima o maxDepth**: Profundidad máxima del árbol. El valor por defecto es “5”.
- **Instancias por nodo o minInstancesPerNode**: Para que un nodo se divida aún más, cada uno de sus hijos debe recibir al menos este número de instancias de entrenamiento. Valor por defecto “1”.
- **Ganancia de información o minInfoGain**: Para que un nodo se divida aún más, la división debe mejorar al menos esta cantidad (en términos de ganancia de información). Valor por defecto “0,0”.
- **Número de contenedores o maxBins**: Número de contenedores utilizados al discretizar características. El aumento maxBins permite que el algoritmo considere más candidatos divididos y tome decisiones divididas detalladas. Sin embargo, también aumenta la computación y la comunicación. Valor por defecto “32”.
- **Memoria máxima o maxMemoryInMB**: Para un procesamiento más rápido, el algoritmo del árbol de decisión recopila estadísticas sobre grupos de nodos para dividir (en lugar de 1 nodo a la vez). El número de nodos que se pueden manejar en un grupo está

determinado por los requisitos de memoria (que varían según las características). El maxMemoryInMB especifica el límite de memoria en términos de megabytes que se puede usar para estas estadísticas. Valor por defecto "256".

- **División de datos o subsamplingRate**: Este parámetro especifica el tamaño del conjunto de datos utilizado para entrenar cada árbol, como una fracción del tamaño del conjunto de datos original. Se recomienda el valor predeterminado "1,0", disminuir esta fracción puede acelerar el entrenamiento. Utilizado en Radom Forest y GBT.

- **Impureza o impurity**: El parámetro de impureza representa la métrica (ganancia de información) para determinar si el modelo se debe dividir en un nodo hoja particular con un valor particular o mantenerlo como está. La única métrica actualmente soportada para los árboles de regresión es la varianza.

- **Nodos en caché o cacheNodeIds**: De forma predeterminada su valor es "False", el algoritmo comunica el modelo actual a los ejecutores para que estos puedan hacer coincidir las instancias de entrenamiento con los nodos de árbol. Cuando esta configuración está activada, el algoritmo en su lugar almacenará en caché esta información, evitando pasar el modelo actual (árbol o árboles) a los ejecutores en cada iteración. Esto puede ser útil con árboles profundos, acelerando el cálculo, y para Random Forest, reduciendo la comunicación en cada iteración.

- **Frecuencia de almacenamiento o checkpointInterval**: Frecuencia de almacenamiento de los nodos en caché. Si se establece en un valor demasiado bajo, se producirá una sobrecarga adicional, y si es demasiado alto puede causar problemas si los ejecutores fallan. Valor por defecto "10".

- **numTrees**: Número de árboles. Solo aplicable en Random Forest. Valor predeterminado "20".

- **Número de características o featureSubsetStrategy**: Número de características para usar como candidatos para la división en cada nodo de árbol. El número se especifica como una fracción o función del número total de características. Disminuir este número acelerará el entrenamiento, pero a veces puede afectar el rendimiento si es demasiado bajo. Puede tener como valores "all" (usa todas), "onethird" (usa 1/3), "sqrt"(la raiz cuadrada), "log2" (usa el logaritmo de 2), y "n" (si n está en el rango (0, 1,0] usa n * número de características, y si no usa n características). En GBT el valor por defecto es "all" y en Random Forest "auto".

- **Función de pérdida o lossType**: Función de pérdida que GBT intenta minimizar. Opciones: "squared", que es el valor por defecto y "absolute".

- **maxIter**: Número máximo de iteraciones a utilizar en GBT. Valor por defecto "20".

- **Contribución de estimadores o stepSize**: Parámetro de tamaño que se utilizará para para reducir la contribución de cada estimador. Su valor está en el intervalo (0, 1], siendo "0,1" por defecto.

# Capítulo 5

# Propuesta

En este capítulo se encuentra dividido en tres secciones, la primera de descripción de los datos, la segunda del preprocesado de los datos y por último la aplicación de los algoritmos. En estas secciones se describen en un orden cronológico los pasos seguidos desde que se reciben los datos hasta que se crea un modelo de predicción, el cual nos proporcionará los resultados sobre el tiempo de llegada de los vuelos.

El objetivo principal de este capítulo es ofrecer una descripción de los datos con los que se va a trabajar, explicar los procesos a los que han sido sometidos los datos iniciales para conseguir un único dataset y por ultimo detallar todas las funciones realizadas con el objetivo de aplicar los algoritmos descritos anteriormente en el capítulo 4.

## 5.1. Descripción de los datos

Anteriormente, en el capítulo 2 de la memoria explicábamos tanto la información incluida en los planes de vuelo, detallado en la sección 2.2, como los datos recogidos por el sistema ADS-B, descritos en la sección 2.3. Los datos obtenidos a través de estos dos medios nos dan como resultado los dos datasets con los que vamos a trabajar.

Las tablas mostradas a continuación hacen una descripción de los datos que conforman dos datasets mencionados con anterioridad. Las dos tablas son:

- Tabla 5.1, la cual describe los datos proporcionados por el plan de vuelo, correspondientes al archivo leg.csv.
- Tabla 5.2, la cual describe los datos sacados de los mensajes en los sistemas ADS-B, correspondientes al archivo message.csv.

| Nombre | Descripción | Ejemplo |
|---|---|---|
| leg_id | Identificador del vuelo. | ANE8322_3442CC_ 1517580582_1517585124 |
| leg_callsign | Código ICAO de la operadora y número de vuelo. | ANE8322 |
| operator | Código ICAO de la compañía operadora del vuelo. | ANE |
| tiempo_inicio_leg | Fecha y hora del inicio del vuelo. | 1517580840 |
| tiempo_final_leg | Fecha y hora del final del vuelo. | 1517588845 |
| cdm_taxi_time | Tiempo de rodaje del avión en pista antes de despegar. | 001600 |
| airport_origin | Aeropuerto de origen. | LIPE |
| departure_runway | Puerta de salida del vuelo. | LIPE12 |
| airport_destination | Aeropuerto de destino. | LEMD |
| arrival_runway | Puerta de llegada del vuelo. | LEMD36R |
| leg_hexident | Dirección ICAO de 24-bit (Código Modo S hexadecimal) | 34530E |
| aircrafttype | Tipo de avión. | CRJX |

Tabla 5.1: Datos de *leg.csv*.

| Nombre | Descripción | Ejemplo |
|---|---|---|
| message_id | Identificador del mensaje. | fr24-34530E-1517587278 |
| timestamp | Fecha y hora en la que ha sido emitido el mensaje. | 1517587278 |
| latitude | Norte/Este positiva, Sur/Oeste negativa (grados) | 44.23577 |
| longitude | Norte/Este positiva, Sur/Oeste negativa (grados) | 9.55458 |
| altitude | Altitud barométrica. | 26925 |
| speed | Velocidad. | 450 |
| vspeed | Velocidad vertical. | -960 |
| ground | Es true cuando el avión está en tierra y false cuando no lo está. | false |

Tabla 5.2: Datos de *message.csv*.

## 5.2. Preprocesado de los datos

Una vez descritos los datos que componen los dos archivos csv con los que vamos a trabajar, pasaremos a explicar el proceso seguido por nuestra función de preprocesado de datos.


Figura 5.1: Ejecución del programa de preprocesado.

Este programa se encarga de recibir los datos presentados anteriormente y proporcionar como salida un único csv que será el utilizado en la aplicación para su tratamiento con los algoritmos. El programa nos pedirá introducir la ruta de los dos csv descritos en la sección anterior, así como el aeropuerto de destino por el que se van a filtrar los vuelos. En la Figura 5.1 se muestra un ejemplo de la ejecución.

Primeramente creamos dos spark dataframe, *leg* y *message*, utilizando los datos de los archivos leg.csv y message.csv. Para la creación de las dataframes utilizaremos la función *read.format()* donde especificaremos el tipo de archivo del fichero que contiene los datos. Por último se la pasará a la función *load()* las variables con las rutas de los ficheros. Una vez creados los dos dataframes pararemos a añadirles las cabeceras con los nombres de cada columna, dichos nombres estarán contenidos en un array de strings, *leg_headers* y *message_headers*. La función *toDF()* modificará el dataframe según el esquema que se le ha pasado en las variables anteriores.

```
#lectura de csv

leg = spark.read.format("csv")\
.load(route_leg)

message = spark.read.format("csv")\
.load(route_message)

#cabeceras

leg_headers= ['leg_id', 'leg_callsign', 'operator', '
   tiempo_inicio_leg',
'tiempo_final_leg', 'cdm_taxi_time', 'rtfm_consumed_fuel',
'rtfm_route_charges', 'airport_origin', 'departure_runway',
'airport_destination', 'arrival_runway', 'fp_registration',
'leg_hexident', 'aircrafttype']

message_headers =['message_id','timestamp','latitude','longitude',
'altitude', 'speed','vspeed','squawk','track','ground', 'leg_id']

leg = leg.toDF(*leg_headers)
message = message.toDF(*message_headers)
```

Filtramos los datos del dataframe *leg* por el aeropuerto de destino introducido, que se encuentra en la variable *airport_destination*. En nuestro caso nos quedaremos únicamente con los vuelos que tienen como destino el Aeropuerto de Madrid-Barajas Adolfo Suárez.

```
leg.createOrReplaceTempView("leg")

leg = spark.sql("SELECT * FROM leg WHERE airport_destination= '" +
   airport_destination + "'")
```

Haremos uso del dataframe *message*, el cual contiene los mensajes proporcionados por ADS-B, para extraer los tiempos reales de despegue y aterrizaje de los vuelos.

Los tiempos de despegue los obtendremos agrupando los mensajes por vuelo, mediante *groupBy('leg_id')*, de los que cogeremos el tiempo del primer mensaje (*f.min('timestamp')*) que tenga la columna *ground* a true (*filter(message['ground']=='true')*). La función *agg()* añadirá a cada fila una columna nueva llamada *take_off* (*alias('take_off')*) con los tiempos calculados anteriormente, que junto con *leg_id* nos dará como resultado el dataframe *take_off*.

Lo mismo ocurrirá para los tiempos de aterrizaje a diferencia de que se cogerá el último mensaje que tenga la columna *ground* a true. En este caso dará como resultado el dataframe *touchdown* con dos columnas: *leg_id* y *touchdown*. Por ultimo uniremos, mediante *leg_id* con la función *join()*, los dataframes *take_off* y *touchdown* con el dataframe *leg* que contenía los datos del plan de vuelo.

```
take_off = message.filter(message["ground"]=="true").groupBy('leg_id'
    ).agg(f.min('timestamp').alias('take_off'))

touchdown = message.filter(message["ground"]=="true").groupBy('leg_id
    ').agg(f.max('timestamp').alias('touchdown'))

flight = leg.join(take_off,on='leg_id')

flight = flight.join(touchdown,on='leg_id')
```

Obtendremos como resultado un dataframe con la siguiente estructura.

```
|-- leg_id: string (nullable = true)
|-- leg_callsign: string (nullable = true)
|-- operator: string (nullable = true)
|-- tiempo_inicio_leg: string (nullable = true)
|-- tiempo_final_leg: string (nullable = true)
|-- cdm_taxi_time: string (nullable = true)
|-- rtfm_consumed_fuel: string (nullable = true)
|-- rtfm_route_charges: string (nullable = true)
|-- airport_origin: string (nullable = true)
|-- departure_runway: string (nullable = true)
|-- airport_destination: string (nullable = true)
|-- arrival_runway: string (nullable = true)
|-- fp_registration: string (nullable = true)
|-- leg_hexident: string (nullable = true)
|-- aircrafttype: string (nullable = true)
|-- take_off: integer (nullable = true)
|-- touchdown: integer (nullable = true)
```

Lo último que hará nuestro programa de preprocesado será generar un fichero csv con el dataframe resultante. Para ello se utilizaran las funciones: *repartition(1)* hará que el dataframe este distribuido en una sola partición de tal forma que solo se genere un archivo con todos los datos, en *write.format()* especificaremos el formato del archivo a generar, en nuestro caso será un csv, *option('header', 'true')* hará que se incluyan las cabeceras en el archivo generado, y por último en *save()* se especificará el nombre de la carpeta que contendrá el archivo, en nuestro caso será la fecha y hora actual.

```

#guardamos el dataset en una carpeta con la fecha de creacion

save_folder = datetime.datetime.now().strftime("%Y-%m-%d_%H.%M.%S")

flight.repartition(1)\
        .write.format("com.databricks.spark.csv")\
        .option("header", "true")\
        .save(save_folder)
```

## 5.3. Aplicación de los algoritmos

A continuación se explicarán los pasos necesarios en la preparación de los datos para su posterior entrenamiento y creación de diferentes modelos a partir de los algoritmos descritos en las subsecciones 4.1.1, 4.1.2 y 4.1.3. Los resultados proporcionados por los modelos serán evaluados a partir de distintas métricas explicadas anteriormente en la sección 4.3.

Lo primero que haremos será preparar los datos de las columnas que se van a utilizar para elaborar las predicciones. Para que los algoritmos puedan procesar se deben transformar las columnas con datos de tipo texto a numérico, la función *StringIndexer* codifica una columna con valores de tipo texto, asignándoles a cada clase o palabra diferente un índice de tipo numérico. Una vez tenemos todas las columnas de tipo numérico, generaremos un vector por cada fila con los datos de las columnas. Esto lo conseguiremos con la función *VectorAssembler* que tiene como parámetros: *inputCols* al que le pasamos un array de texto con las columnas a procesar y *outputCol* al que le pasamos el nombre de la columna que contendrá los vectores.

```
stringIndexer_0 = StringIndexer(inputCol=string_cols[0], outputCol=
   string_cols[0]+"_index").fit(df)

vectorAssembler_features = \
VectorAssembler(inputCols= cols_vector , outputCol="features")
```

A continuación podemos ver los resultados, en primer lugar del uso de *StringIndexer* y en segundo lugar del uso de *VectorAssembler*.

```
+--------+--------------+
|operator|operator_index|
+--------+--------------+
|     IBE|           0.0|
|     AAL|          18.0|
|     AAL|          18.0|
|     DAL|          32.0|
|     LPE|          48.0|
|     ACA|          61.0|
+--------+--------------+

+-----------------------------------------------------------+
|features                                                   |
+-----------------------------------------------------------+
|[1.5175271E9,1.5175271E9,1.51755824E9,0.0,9.0,28.0]        |
|[1.517527789E9,1.517527789E9,1.517552678E9,18.0,6.0,95.0]|
|[1.517529105E9,1.517529105E9,1.51755757E9,18.0,27.0,62.0]|
|[1.51752964E9,1.51752964E9,1.51755286E9,32.0,11.0,66.0]  |
|[1.51753047E9,1.51753047E9,1.517572224E9,48.0,11.0,54.0] |
|[1.517530584E9,1.51753031E9,1.517554625E9,61.0,8.0,100.0]|
+-----------------------------------------------------------+
```

El siguiente paso a realizar es una división de los datos de manera aleatoria en datos de entrenamiento y datos de prueba, para ello usamos la función *randomSplit* para dividir de manera aleatoria el dataset. En la variable *split_train* tendremos el porcentaje de los datos elegido para el entrenamiento, en *split_test* el porcentaje de datos de prueba y por último una semilla para la generación aleatoria. Tendremos por un lado un dataframe *train* con los datos de entrenamiento y por otro un dataframe *test* con los datos de prueba. Para la aplicación de los algoritmos tendremos las funciones *DecisionTreeRegressor*, *RandomForestRegressor* y *GBTRegressor*, que tendrán como parámetros: *featuresCol* indica el nombre de la columna con los vectores generados anteriormente, *labelCol* indica el nombre de la columna a predecir y por último tendremos como parámetros todas las opciones definidas en la subsección 4.4.1.

```
# division de los datos en test o entrenamiento
split_test = 1-split_train
splits = df.randomSplit([split_train, split_test], 1234)
train = splits[0]
test = splits[1]


dt = DecisionTreeRegressor(featuresCol="features",labelCol=label,
maxDepth = maxDepth, maxBins = maxBins, minInstancesPerNode =
   minInstancesPerNode, minInfoGain = minInfoGain, maxMemoryInMB =
   maxMemoryInMB, cacheNodeIds = cacheNodeIds, checkpointInterval =
   checkpointInterval)

```

```
11
12 rf = RandomForestRegressor(featuresCol="features",labelCol=label,
13 maxDepth = maxDepth, maxBins = maxBins, minInstancesPerNode =
      minInstancesPerNode, minInfoGain = minInfoGain,
14 maxMemoryInMB = maxMemoryInMB, cacheNodeIds = cacheNodeIds,
      checkpointInterval = checkpointInterval,
15 subsamplingRate=subsamplingRate, numTrees=numTrees,
16 featureSubsetStrategy= featureSubsetStrategy)
17
18
19 gbt = GBTRegressor(featuresCol="features",labelCol=label, maxDepth =
      maxDepth, maxBins = maxBins, minInstancesPerNode =
      minInstancesPerNode, minInfoGain = minInfoGain, maxMemoryInMB =
      maxMemoryInMB, cacheNodeIds = cacheNodeIds, checkpointInterval =
      checkpointInterval, subsamplingRate=subsamplingRate,
      featureSubsetStrategy= featureSubsetStrategy, lossType =lossType,
      maxIter = maxIter, stepSize = stepSize)
```

La variable *stages* contendrá un array con una serie de estados que se ejecutaran de manera ordenada para la creación de un flujo mediente *Pipeline()*. En nuestro caso los estados vendrán definidos, primero por *stringIndexer_i*, segundo por *vectorAssembler_features* y por último los distintos algoritmos *dt, rf,* o *gbt*. Usaremos la función *pipeline.fit()* con los datos de entrenamiento, contenidos en el dataset *train*, con el fin de generar un modelo. Una vez generado el modelo se utilizaran los datos de prueba para elaborar las predicciones con la función *transform*. Esta función convertirá el dataframe *test* con los datos de prueba en un nuevo dataframe que tendrá una columna adicional, llamada *prediction*, con los resultados de las predicciones.

```
1 stages=[stringIndexer_operator, stringIndexer_aircrafttype,
      vectorAssembler_features,dt]
2 pipeline = Pipeline(stages = stages)
3
4 # entrenamiento del modelo
5 model = pipeline.fit(train)
6
7 # elaboracion de las predicciones
8 predictions = model.transform(test)
```

```
+--------------------+-------------+--------------------+
|            features|        label|          prediction|
+--------------------+-------------+--------------------+
|[1.5175271E9,1.51...|1.517557974E9|1.5175570856111112E9|
|[1.517527789E9,1....|  1.5175506E9|   1.5175498041875E9|
|[1.51752964E9,1.5...| 1.51755231E9|1.5175525837058823E9|
|[1.517530584E9,1....|1.517554136E9|1.5175543195454545E9|
+--------------------+-------------+--------------------+
```

Una vez hechas las predicciones el siguiente paso será la evaluación de los resultados obtenidos para ello usaremos la función *RegressionEvaluator*, le pasaremos la columna con las predicciones (*predictionCol*), la columna a predecir (*labelCol*) y la métrica que se utilizara (*metricName*). En nuestro caso utilizaremos las métricas definidas en la sección 4.3. Una vez definido el evaluador se aplicara sobre la columna de las predicciones (*evaluator_rmse.evaluate(predictions)*), obteniendo el resultado numérico correspondiente.

```
1
2  evaluator_rmse = RegressionEvaluator(labelCol= label, predictionCol="
       prediction", metricName="rmse")
3  rmse = evaluator_rmse.evaluate(predictions)
4
5  evaluator_mae = RegressionEvaluator(labelCol="touchdown",
       predictionCol="prediction", metricName="mae")
6  mae = evaluator_mae.evaluate(predictions)
7
8  evaluator_mse = RegressionEvaluator(labelCol="touchdown",
       predictionCol="prediction", metricName="mse")
9  mse = evaluator_mse.evaluate(predictions)
10
11 evaluator_r2 = RegressionEvaluator(labelCol="touchdown",
       predictionCol="prediction", metricName="r2")
12 r2 = evaluator_r2.evaluate(predictions)
```

```
Root Mean Squared Error (RMSE) on test data = 2598.17
Mean Absolute Error (MAE) on test data = 1265.15
Mean Squared Error (MSE) on test data = 6.75047e+06
Coefficient of Determination (R2) on test data = 0.981977
```

# Capítulo 6

# Dashboard

En este capítulo se describirá la aplicación web realizada. La cual nos permitirá visualizar los datos en tablas y gráficos, modificar los datos pudiendo generar un nuevo csv, seleccionar un algoritmo configurando todos sus parámetros, realizar baterías de experimentos y por ultimo visualizar los resultados en un gráfico.

El capítulo constará de tres secciones. En la primera se mostrarán los bocetos iniciales sobre los que está basado el diseño de nuestra aplicación, donde se pueden ver los diferentes componentes de las pestañas que la componen. En la segunda sección se explicaran las tecnologías utilizadas para llevar a cabo el desarrollo de nuestra aplicación, para más tarde explicar detalles de la implementación llevada a cabo. Por último se presentaran los resultados asociados a las pruebas a las que hemos sometido a nuestra aplicación para asegurarnos de su correcto funcionamiento.

## 6.1. Diseño

Como hemos dicho en la introducción a este capítulo, en esta sección nuestro objetivo es mostrar una serie de bocetos iniciales que nos sirvieron para establecer el diseño de nuestra aplicación. En este caso nuestra aplicación está compuesta por un menú de tres pestañas: Datos, ML y Experimentos. A continuación se mostrará el diseño de cada una de ellas.

### 6.1.1. Pestaña Datos

En esta pestaña el usuario tendrá la posibilidad de cargar un dataset, una vez cargado podrá seleccionar una de las columnas que lo componen y así visualizar los datos de dicha columna. Estos datos se mostraran mediante una tabla y un gráfico.

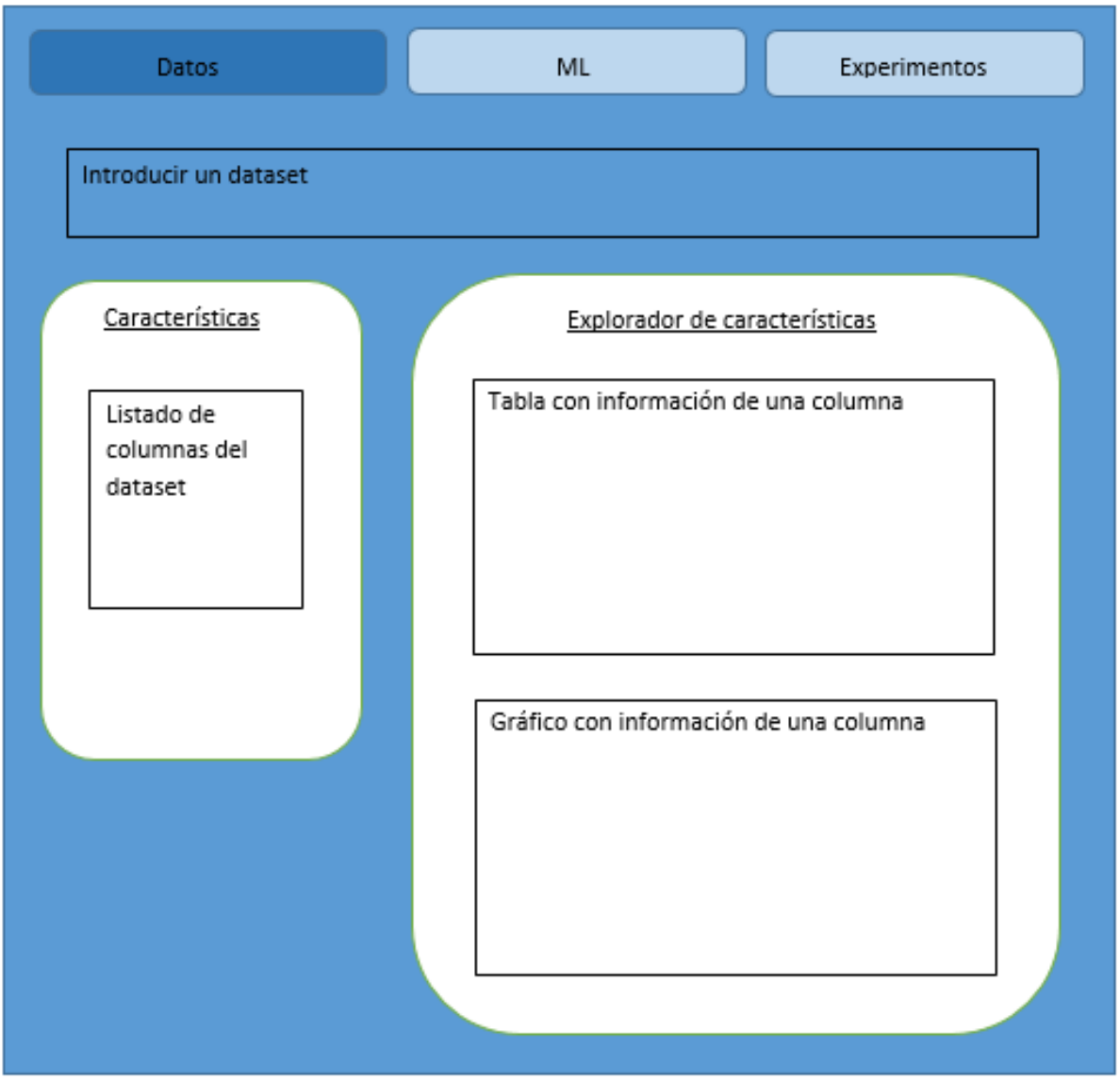



Figura 6.1: Diseño de la pestaña Datos.

### 6.1.2. Pestaña ML

La funcionalidad de esta pestaña es la preparación de los datos que van a ser procesados por el algoritmo elegido y la especificación de sus parámetros, además de presentar los resultados mediante métricas y un gráfico. También se permitirá la eliminación de una serie de filas y la creación de un nuevo csv.

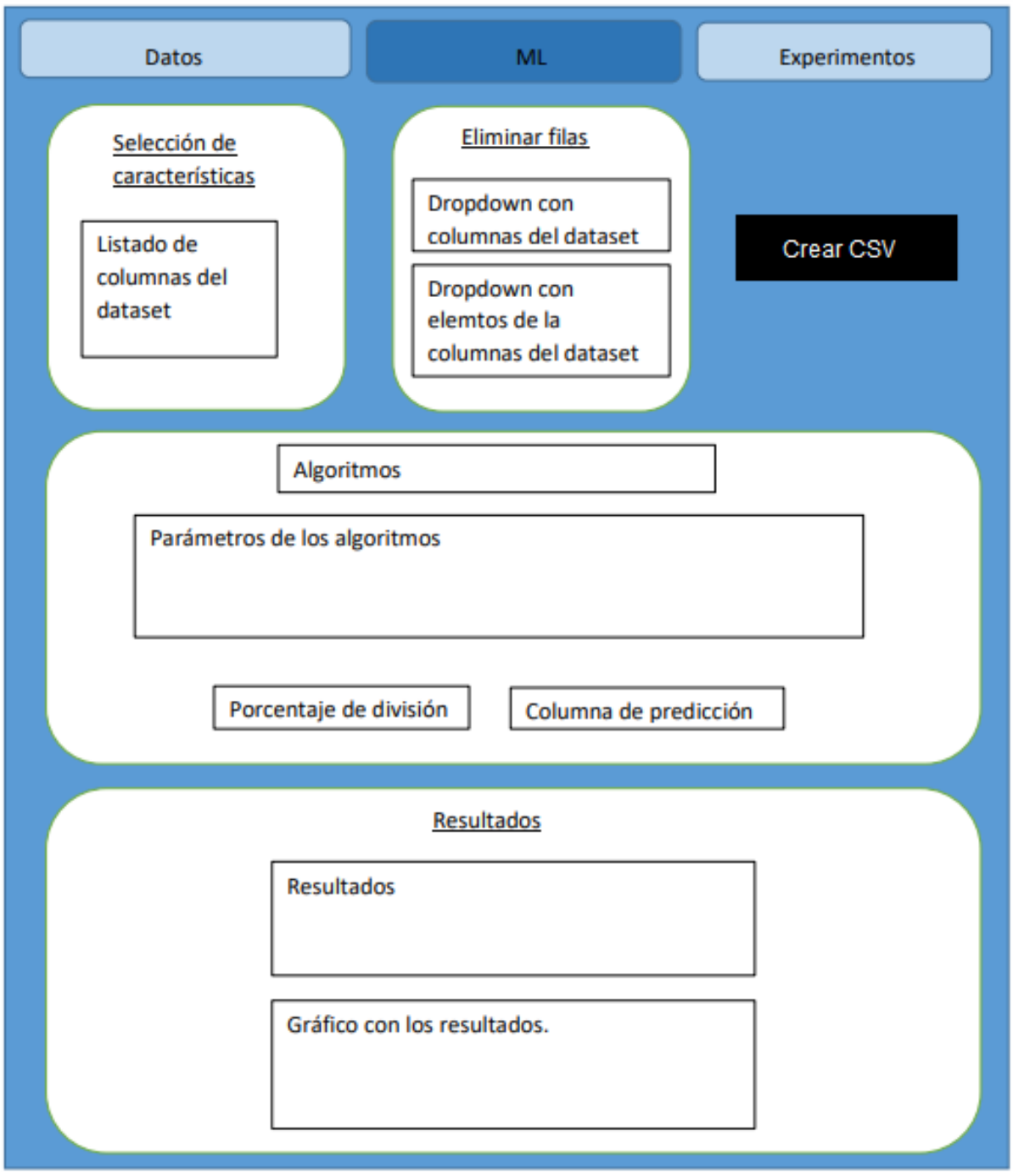



Figura 6.2: Diseño de la pestaña ML.

### 6.1.3. Pestaña Experimentos

El diseño de la pestaña experimentos tiene como objetivo poder cargar diferentes datasets y algoritmos, mediante pestañas emergentes, para poder ser ejecutados y mostrar una lista de todos los resultados. Siendo la presentación de los resultados muy parecida a la mencionada anteriormente en la pestaña ML.

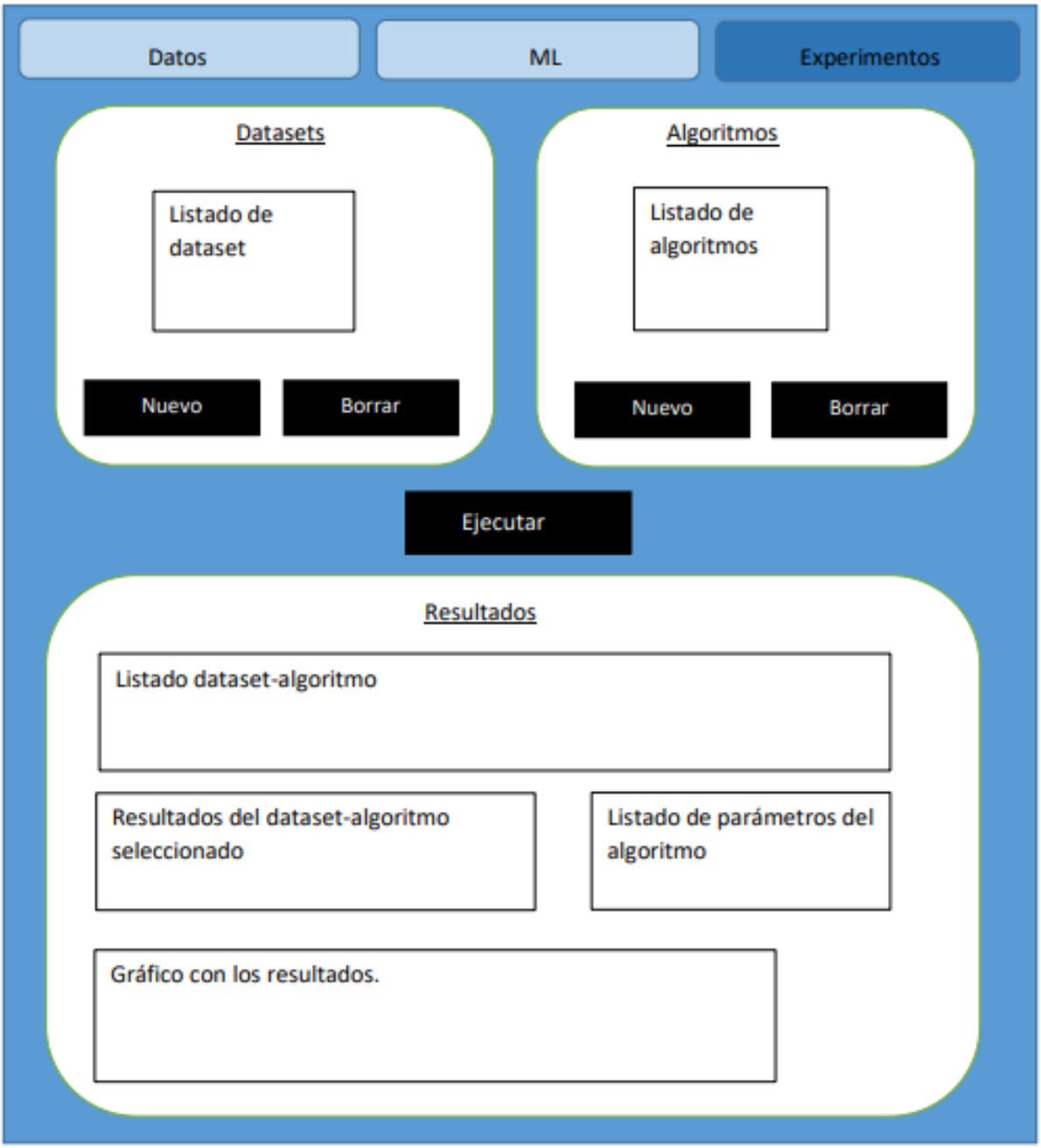



Figura 6.3: Diseño de la pestaña Experimentos.

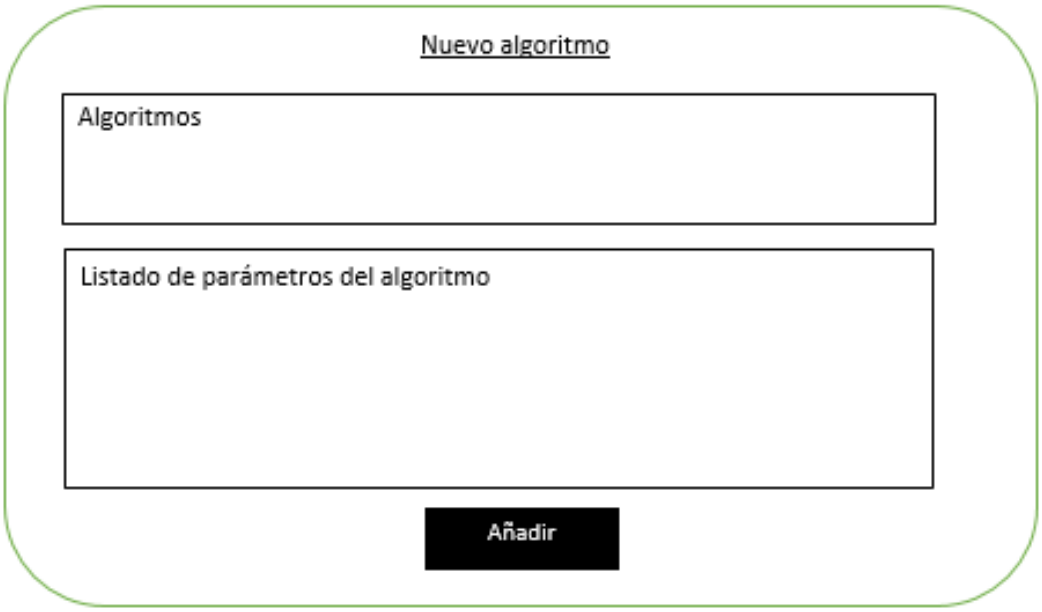



Figura 6.4: Diseño de la ventana emergente Nuevo Algoritmo.

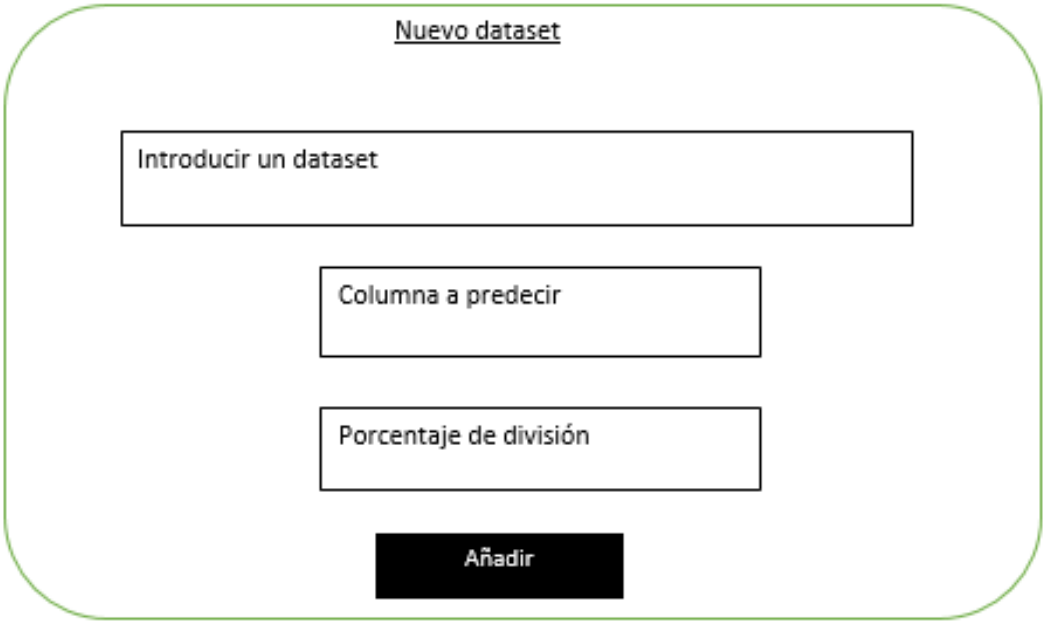



Figura 6.5: Diseño de la ventana emergente Nuevo Dataset.

## 6.2. Implementación

Para la implementación de la aplicación realizada se decidió utilizar Dash. Dash es un framework hecho para construir aplicaciones web de una forma rápida y sencilla, está desarrollado sobre Flask, Plotly.js y React.js. Los principales motivos que nos llevaron a utilizar este framework son, que es una plataforma de código abierto que nos permitía desarrollar aplicaciones de visualización de datos altamente personalizables con el lenguaje Python, además de ofrecer multitud de recursos y una documentación detallada[1]. Dash soporta Python 2 y 3. La instalación de Dash es muy sencilla teniendo solo que introducir el siguiente comando.

```
pip install dash==1.9.1
```

Como se ha mencionado antes Dash está compuesto por Flask, Plotly.js y React.js. Flask es un micro framework utilizado para crear aplicaciones web. React.js una librería de interfaces de usuario escrita en Javascript y mantenida por Facebook. Plotly.js es otra librería que proporciona componentes para la creación de gráficos. Las aplicaciones Dash corren bajo los servidores web de Flask comunicando paquetes JSON para cada petición HTTP. Las aplicaciones son renderizadas y almacenadas en el navegador, pudiendo configurar sesiones diferentes para varios usuarios al mismo tiempo.

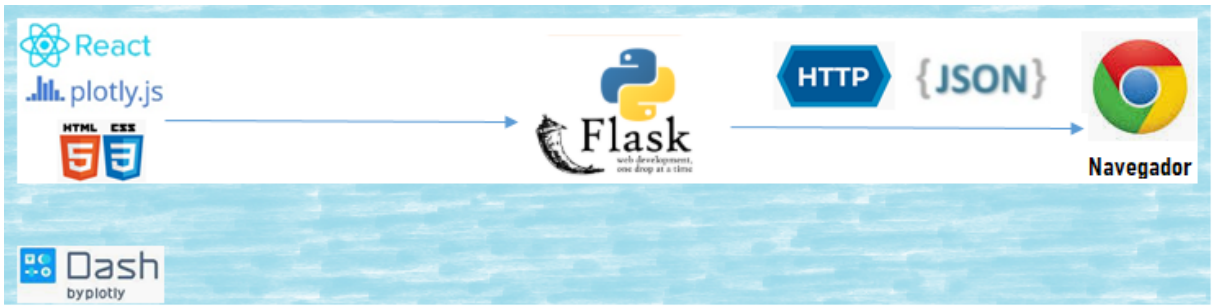



Figura 6.6: Arquitectura de Dash.

Las aplicaciones creadas con Dash permiten una alta personalización de sus componentes, teniendo una librería que nos permite utilizar etiquetas del lenguaje HTML, también se puede insertar código CSS en cada uno de los componentes. A continuación se muestra el código de una aplicación básica en Dash [39]. En las primeras líneas se importan las librerías con los componentes principales de dash y la librería de html mencionada anteriormente. Se declara la variable *app* a la que se le añadirá un fichero css (*external_stylesheets*). La variable *app* contendrá un *layout* con los componentes que se mostraran en la aplicación. Por último se llamará al método *app.run_server(debug=True)* dentro del método main de la aplicación, para que el servidor despliegue la aplicación.

[1]Documentación de Dash https://dash.plot.ly

```
import dash
import dash_core_components as dcc
import dash_html_components as html

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__, external_stylesheets=external_stylesheets)

app.layout = html.Div(children=[
        html.H1(children='Hello Dash'),

        html.Div(children='''
                Dash: A web application framework for Python.
        '''),

        dcc.Graph(
                id='example-graph',
                figure={
                        'data': [
                                {'x': [1, 2, 3], 'y': [4, 1, 2], '
                                   type': 'bar', 'name': 'SF'},
                                {'x': [1, 2, 3], 'y': [2, 4, 5], '
                                   type': 'bar', 'name': 'Montreal'},
                        ],
                        'layout': {
                                'title': 'Dash Data Visualization'
                        }
                }
        )
])


if __name__ == '__main__':

app.run_server(debug=True)
```

### 6.2.1. Desarrollo de la aplicación

A continuación pasaremos a explicar cómo se han implementado diferentes partes de la aplicación. En primer lugar explicaremos las carpetas y archivos que componen la aplicación. Una vez explicados se expondrán fragmentos de código para hacernos una idea de cómo ha sido la implementación de la aplicación.

En el archivo app.py se importa dash, se declaran las variables principales de la aplicación, y se definen todos los métodos implementados con spark. A continuación se pueden ver todas las

importaciones tanto de dash como de spark, ademas de la definición de las variables app y server de la aplicación y la variable de inicio de sesión en spark. El resto de funciones implementadas son las relacionadas con los algoritmos descritas en la sección 5.3.

```
import dash
import datetime

app = dash.Dash(__name__)
server = app.server
app.config.suppress_callback_exceptions = True

import findspark
findspark.init()

from pyspark.sql import SparkSession
from pyspark.sql.types import StringType,StructType,DoubleType
from pyspark.sql.functions import col, abs
from pyspark.ml import Pipeline
from pyspark.ml.feature import StringIndexer, VectorAssembler
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.regression import DecisionTreeRegressor,
   RandomForestRegressor, GBTRegressor

# inicio de sesion en spark
spark = SparkSession \
.builder \
.appName("TFG") \
.master("local[*]")\
.getOrCreate()


# Enable Arrow-based columnar data transfers
spark.conf.set("spark.sql.execution.arrow.enabled", "true")

# creacion de un Spark Dataframe vacio
schema = StructType([])
empty = spark.createDataFrame([], schema)
```

El segundo archivo index.py será el que contendrá el método main encargado de arrancar la aplicación, y el layout principal. En el siguiente código se ve como se ha configurado el menú con nuestras tres pestañas de tal modo que al pulsar la pestaña Datos nos referencia a la variable layout_datos del archivo data.py, al pulsar la pestaña ML menciona a la variable layout_ml del archivo anterior, y por último la pestaña Experimentos referencia a la variable layout del archivo experiment.py. Al inicio del código se puede ver como se importan dichos archivos ubicados en la carpeta apps.

```
from dash.dependencies import Input, Output
import dash_core_components as dcc
import dash_html_components as html

from app import app
from apps import data, experiment


app.layout = html.Div(
        [
                # header
                html.Div([
                        html.Div(
                                id="banner-text",
                                        children=[
                                                html.H5("ARBOLES
                                                   REGRESIVOS"),
                                        ],
                                ),
                        ],
                        className="banner"
                ),

                # tabs
                html.Div([

                        dcc.Tabs(
                        id="tabs",
                        children=[
                                dcc.Tab(label="Datos", value="
                                   data_tab", style=tab_style,
                                   selected_style=tab_selected_style)
                                   ,
                                dcc.Tab(label="ML", value="ml_tab",
                                   style=tab_style, selected_style=
                                   tab_selected_style),
                                dcc.Tab(label="Experimentos", value="
                                   experiments_tab", style=tab_style,
                                    selected_style=tab_selected_style
                                   ),
                        ],
                        value="data_tab",
                        )
                ],style={'marginRight': '3.5%', 'marginLeft': '3.5%'}
                ),

```

```
                # Tab content
                html.Div(id="tab_content",  style={"margin": "2% 3%",
                    'backgroundColor': 'transparent'}),
        ],
)

@app.callback(
        Output("tab_content", "children"),
        [Input("tabs", "value")])
def render_content(tab):

        if tab == "data_tab":
                return data.layout_datos
        elif tab == "ml_tab":
                return data.layout_ml
        elif tab == "experiments_tab":
                return experiment.layout
        else:
                return data.layout_datos


if __name__ == "__main__":

        app.run_server(debug=False)
```

Como hemos mencionado anteriormente, en el archivo data.py se encuentra el desarrollo de todos los elementos presentes en las interfaces de las pestañas Datos y ML. El archivo tiene dos partes diferenciadas, en la primera parte encontraremos la variable layout_datos. En ella están los elementos presentes en la interfaz de la pestaña datos. Después tendremos los métodos que implementan las acciones presentes en la misma. En la segunda parte del archivo encontraremos la variable layout_ml en la que estarán contenidos los diferentes elementos de la pestaña ML, y como en el caso anterior justamente después encontraremos los métodos encargados de la funcionalidad de la interfaz.

En el siguiente código se mostrará el desarrollo de uno de los paneles presentes en la pestaña ML, en este caso será el panel de “Eliminar filas". En el código se ve cómo se actualizan los valores de la columna a elegir, id= 'column-row', en función de las columnas marcadas en el panel de “Selección de características", id= 'columns-select', en el método def set_columns_options(opcions). Más tarde se actualizan los valores con los elementos de una columna en el método def set_rows_options(columnrow), y finalmente se borran los elementos de la columna en el método def delete_row(n_clicks_timestamp, column, row).

```
dcc.Checklist(
        id='columns-select',
        options=[],#[{'label': i, 'value': i} for i in df_spark.
           toPandas().columns.values],
```

```
        value=[],
        labelStyle={'display': 'inline-block','margin': '1%'},
        style={'marginLeft': 20}
),
html.Div([
        html.Button(id='select',
                n_clicks_timestamp=0,
                children='Seleccionar',
                style= {'backgroundColor':'#3F455E',
                'color': 'white',
                'marginLeft': 0,
                'marginRight': 10,
                'marginTop': 10,
                'marginBottom': 10}
        ),

        ],style={'text-align': 'center'}
),

html.Div([
        html.Div(html.P("Eliminar filas"),style={'textAlign': 'center
           ','fontSize': 25,'marginTop': 0}),

        html.Div(html.Hr() ,style={'marginRight': 20,'marginLeft':
           20, 'marginTop': 10}),

        html.Div([
                dcc.Dropdown(
                        id='column-row',
                        options=[],
                        value='',
                        style={'margin': 10,
                                'marginRight': 0,
                                'marginLeft': 0}
                ),
                dcc.Dropdown(
                        id='row-row',
                        options=[],
                        value='',
                        multi=True,
                        style={'margin': 10,
                                'marginRight': 0,
                                'marginLeft': 0}
                ),
                html.Div([html.Button(id='rows',
                                        n_clicks_timestamp=0,
                                        children='Eliminar',
```

```
                                        style= {'backgroundColor':'#3F455E',
                                                'color': 'white',
                                                'marginLeft': 10,
                                                'marginRight': 10,
                                                'marginTop': 10,
                                                'marginBottom': 10}
                                        ),
                        ],style={'text-align': 'center'}
                ),

                dcc.Loading(
                        id="loading-2",
                        children=[html.Div([html.Div(id='rows-state')])],
                        type="circle",
                )
        ],
        style={'marginLeft': 20,
                'marginRight': 20,
                'marginTop': 0,
                'marginBottom': 10,
                }
        )

        ],
        style={'width': '31%',
                        'margin': '2%',
                        'display': 'inline-block',
                        'min-height': 300,
                        'vertical-align': 'middle',
                        'backgroundColor':'#FFFFFF',
                        'box-shadow': '2px 2px 2px lightgrey',
                        'border-radius': 5
                        }
),

#implementacion de los metodos de la interfaz

@app.callback(
        Output('column-row', 'options'),
        [Input('columns-select', 'value')])
def set_columns_options(opcions):

        return [{'label': i, 'value': i} for i in opcions]

@app.callback(
```

```
94          Output('row-row', 'options'),
95          [Input('column-row', 'value')])
96  def set_rows_options(columnrow):
97
98          global df_pandas_ml
99
100         timestamp_cols = ['tiempo_inicio_leg', 'tiempo_final_leg', '
               take_off', 'touchdown']
101
102         if columnrow != '':
103
104                 item = df_pandas_ml[columnrow].value_counts().
                       rename_axis('values').reset_index(name='count')
105
106                 if columnrow in timestamp_cols:
107
108                         item['values'] = pd.to_datetime(item['values'
                               ].astype(int), unit='s')
109
110                 item = item.astype(str)
111
112                 return [{'label': i, 'value': i} for i in item[item.
                       columns.values[0]]]
113
114         else:
115                 return []
116
117
118 @app.callback(
119         [Output('rows-state', 'children'),
120         Output('column-row', 'value'),
121         Output('row-row', 'value')],
122         [Input('rows', 'n_clicks_timestamp')],
123         [State('column-row', 'value'),
124         State('row-row', 'value')])
125 def delete_row(n_clicks_timestamp, column, row):
126
127         global df_pandas_ml
128
129         timestamp_cols = ['tiempo_inicio_leg', 'tiempo_final_leg', '
               take_off', 'touchdown']
130
131         #column es de tipo date y es un array
132         if column in timestamp_cols:
133                 for i in row:
134                         row[row.index(i)] = str(int(datetime.strptime
                               (i, '%Y-%m-%d %H:%M:%S').replace(tzinfo=
```

```
                                timezone.utc).timestamp()))
135
136                     rows_delete = ''
137                     for u in row:
138                             rows_delete = rows_delete + str(datetime.
                                   utcfromtimestamp(int(u))) + ', '
139
140             #eliminacion de vuelos
141             df_spark = pd_spark(df_pandas_ml)
142
143             df_spark = rowdelete(df_spark,column,row)
144
145             df_pandas_ml = df_spark.toPandas()
146
147             rows_delete = ''
148             for u in row:
149                     rows_delete = rows_delete + u + ', '
150
151             rows_delete = rows_delete[:len(rows_delete) - 2]
152
153             return u'Se han eliminado de {} los valores {}.'.format(
                   column, rows_delete),'',''
```

Por ultimo tendremos el archivo experiment.py, en este archivo se seguirá la misma estructura que se ha mencionado con anterioridad. Tenemos una variable layout con los elementos de la interfaz y posteriormente los métodos que implementan las diferentes funcionalidades. Una diferencia con respecto a lo anterior, es dos métodos que se encuentran al inicio del documento que presentan los elementos de las interfaces de dos pestañas emergentes.

En el siguiente código se muestra como se actualiza el valor de los parámetros según el algoritmo escogido.

```
1  dcc.RadioItems(
2          id='type_algorithm',
3
4          options=[
5                  {'label': 'Decision tree', 'value': 'dt'},
6                  {'label': 'Random forest', 'value': 'rf'},
7                  {'label': 'Gradient-boosted Tree', 'value': 'gbt'}
8          ],
9          value='',
10         labelStyle={
11                 'display': 'inline-block',
12                 'marginLeft': 20,
13                 'marginRight': 20,
14                 'marginTop': 10,
15                 'marginBottom': 10
```

```
        },
        style={
                'marginLeft': 20,
                'marginRight': 20,
                'marginTop': 10,
                'marginBottom': 10,
                'textAlign': 'center'
        }
),
html.Hr(),
html.Div([
        html.P("Profundidad maxima"),
        dcc.Input(id='1_param',type='text'),
],style={'display': 'inline-block','vertical-align': 'middle','
   marginLeft': 10,'marginRight': 10,'marginTop': 10,'marginBottom':
   10 }),

html.Div([
        html.P("Instancias por nodo"),
        dcc.Input(id='2_param',type='text'),
],style={'display': 'inline-block','vertical-align': 'middle','
   marginLeft': 10,'marginRight': 10,'marginTop': 10,'marginBottom':
   10 }),

html.Div([
        html.P("Numero de caracteristicas"),
        dcc.Dropdown(
        id='11_param',
        options=[
                {'label': 'all', 'value': 'all'},
                {'label': 'auto', 'value': 'auto'},
                {'label': 'onethird', 'value': 'onethird'},
                {'label': 'sqrt', 'value': 'sqrt'},
                {'label': 'log2', 'value': 'log2'}
        ],
        style={ 'width': '100%'}
        ),
],style={'display': 'inline-block','margin': 10,'width': 200, '
   vertical-align': 'middle'  }),
html.Div([
        html.P("Iteraciones maximas", ),
        dcc.Input(id='13_param',type='text'),
],style={'display': 'inline-block','vertical-align': 'middle', '
   marginLeft': 10,'marginRight': 10,'marginTop': 10,'marginBottom':
   10 }),
html.Div([
        html.P("Contribucion de estimadores"),
```

```
56          dcc.Input(id='14_param',type='text'),
57  ],style={'display': 'inline-block','vertical-align': 'middle','
       marginLeft': 10,'marginRight': 10,'marginTop': 10,'marginBottom':
       10 }),
58
59  # submit button
60  html.Button(
61          'Anadir',
62          id='submit_new_algorithm',
63          n_clicks=0,
64          n_clicks_timestamp = 0,
65          style= {'backgroundColor':'#3F455E',
66                          'color': 'white',
67                          'marginLeft': 10,
68                          'marginRight': 10,
69                          'marginTop': 10,
70                          'marginBottom': 10}
71  ),
72
73  @app.callback(
74          [Output('1_param', 'disabled'),
75          Output('1_param', 'value'),
76          Output('1_param', 'style'),
77          Output('2_param', 'disabled'),
78          Output('2_param', 'value'),
79          Output('2_param', 'style'),
80          Output('3_param', 'disabled'),
81          Output('3_param', 'value'),
82          Output('3_param', 'style'),
83          Output('13_param', 'disabled'),
84          Output('13_param', 'value'),
85          Output('13_param', 'style'),
86          Output('14_param', 'disabled'),
87          Output('14_param', 'value'),
88          Output('14_param', 'style')],
89          [Input('type_algorithm', 'value')])
90  def set_algorithm_options(algorithm):
91
92          if algorithm == "dt":
93                  output = [
94                          False,5,{'backgroundColor': '#242633', 'color
                               ': 'white'}, #maxDepth
95                          False,1,{'backgroundColor': '#242633', 'color
                               ': 'white'}, #minInstancesPerNode
96                          False,0.0,{'backgroundColor': '#242633', '
                               color': 'white'}, #minInfoGain
97                          True,'',{'backgroundColor': '#242633', 'color
```

```
                        ': 'white'}, #maxIter
98                      True,'',{'backgroundColor': '#242633', 'color
                        ': 'white'}  #stepSize
99              ]
100     elif algorithm == "rf":
101             output = [
102                     False,5,{'backgroundColor': '#242633', 'color
                        ': 'white'}, #maxDepth
103                     False,1,{'backgroundColor': '#242633', 'color
                        ': 'white'}, #minInstancesPerNode
104                     False,0.0,{'backgroundColor': '#242633', '
                        color': 'white'}, #minInfoGain
105                     True,'',{'backgroundColor': '#242633', 'color
                        ': 'white'}, #maxIter
106                     True,'',{'backgroundColor': '#242633', 'color
                        ': 'white'}  #stepSize
107             ]
108
109     elif algorithm == "gbt":
110             output = [
111                     False,5,{'backgroundColor': '#242633', 'color
                        ': 'white'}, #maxDepth
112                     False,1,{'backgroundColor': '#242633', 'color
                        ': 'white'}, #minInstancesPerNode
113                     False,0.0,{'backgroundColor': '#242633', '
                        color': 'white'}, #minInfoGain
114                     False,'20',{'backgroundColor': '#242633', '
                        color': 'white'}, #maxIter
115                     False,0.1,{'backgroundColor': '#242633', '
                        color': 'white'}  #stepSize
116             ]
117
118     else:
119             output = [
120                     True,'',{'backgroundColor': '#242633', 'color
                        ': 'white'}, #maxDepth
121                     True,'',{'backgroundColor': '#242633', 'color
                        ': 'white'}, #minInstancesPerNode
122                     True,'',{'backgroundColor': '#242633', 'color
                        ': 'white'}, #minInfoGain
123                     True,'',{'backgroundColor': '#242633', 'color
                        ': 'white'}, #maxIter
124                     True,'',{'backgroundColor': '#242633', 'color
                        ': 'white'}  #stepSize
125             ]
126
127     return output
```

## 6.3. Pruebas

En esta sección nos vamos a enfocar en reflejar las pruebas a las que hemos sometido a la aplicación, para asegurar el correcto funcionamiento de todos sus componentes. Existen numerosos tipos de pruebas software, de rendimiento, se seguridad, de usabilidad, funcionales, etc. En nuestro caso hemos decidido centrarnos en pruebas funcionales, ya que las otras no serán relevantes al tratarse de una aplicación que se ejecutara a nivel local, sin que sea necesario autenticarse y con un uso bastante definido. Las pruebas funcionales son las encargadas de asegurar que la aplicación realiza correctamente todas las funciones especificadas en las historias de usuario.

Las pruebas llevadas a cabo son denominadas como pruebas de caja negra, son un tipo de pruebas funcionales centradas en probar individualmente los distintos módulos de la aplicación. Estas pruebas se limitan a ver lo que ocurre con el componente cuando recibe una entrada de datos, sin hacer hincapié en los detalles internos de implementación, verificando que la salida de los mismos tiene el resultado esperado.

Para documentar la realización de estas pruebas se han decidido utilizar tablas, cada una representará las pruebas realizadas a un componente de la aplicación.

| **PCN-01** | **Cargar un dataset** |
|---|---|
| Propósito | Poder cargar un archivo csv. |
| Prerrequisito | Ninguno. |
| Datos de entrada | Dataset seleccionado. |
| Pasos | **1.** El usuario pincha en en un link y selecciona un archivo, o bien arrastra un archivo. |
| Resultado esperado | Se mostraran las columnas del dataset, y se seleccionará la primera mostrando una tabla y un gráfico con información de la columna. |
| Resultado obtenido | Se mostraran las columnas del dataset, y se seleccionará la primera mostrando una tabla y un gráfico con información de la columna. |
| Resultado de la prueba | Positivo |

Tabla 6.1: Prueba de caja negra 01, cargar un dataset.

| PCN-02 | Visualizar la información de una columna |
|---|---|
| Propósito | Poder ver la información de una columna. |
| Prerrequisito | Haber cargado un dataset. |
| Datos de entrada | Ninguno. |
| Pasos | **1.** Seleccionar la columna a visualizar.<br>**2.** Pulsar el botón visualizar. |
| Resultado esperado | El explorador de características muestra una tabla y debajo un gráfico con los elementos de la columna y el número de elementos. |
| Resultado obtenido | El explorador de características muestra una tabla y debajo un gráfico con los elementos de la columna y el número de elementos. |
| Resultado de la prueba | Positivo |

Tabla 6.2: Prueba de caja negra 02, visualizar la información de una columna.

| PCN-03 | Selección de características |
|---|---|
| Propósito | Poder seleccionar las columnas que van a ser procesadas por los algoritmos. |
| Prerrequisito | Haber cargado un dataset. |
| Datos de entrada | Ninguno. |
| Pasos | **1.** Seleccionar columnas.<br>**2.** Pulsar el botón seleccionar. |
| Resultado esperado | Aparecerán las columnas seleccionadas como opciones tanto en el panel de eliminar filas, como en el campo de predicción. |
| Resultado obtenido | Se actualizan las opciones tanto en el panel de eliminar filas, como en el campo de predicción. |
| Resultado de la prueba | Positivo |

Tabla 6.3: Prueba de caja negra 03, selección de características.

| PCN-04 | Eliminación de filas |
|---|---|
| Propósito | Poder eliminar las filas de un dataset. |
| Prerrequisito | Haber seleccionado al menos una columna en el panel de selección de características. |
| Datos de entrada | Columnas seleccionadas. |
| Pasos | **1.** Seleccionar una columna.<br>**2.** Seleccionar al menos un elemento de una columna.<br>**3.** Pulsar el botón eliminar. |
| Resultado esperado | Aparecerá un mensaje informativo indicando los elementos borrados. |
| Resultado obtenido | Aparece un mensaje informativo indicando los elementos borrados. |
| Resultado de la prueba | Positivo |

Tabla 6.4: Prueba de caja negra 04, eliminación de filas.

| PCN-05 | Guardar un csv |
|---|---|
| Propósito | Crear un csv con las modificaciones hechas por los paneles de selección de características y eliminar filas. |
| Prerrequisito | Haber seleccionado al menos una columna en el panel de selección de características. |
| Datos de entrada | Dataset modificado. |
| Pasos | **1.** Presionar el botón de CREAR CSV. |
| Resultado esperado | Aparecerá un link para descargar el csv generado. |
| Resultado obtenido | Aparece un link de descarga del csv generado. |
| Resultado de la prueba | Positivo. |

Tabla 6.5: Prueba de caja negra 05, crear un csv.

| PCN-06 | Visualización de los parámetros de un algoritmo |
|---|---|
| Propósito | Establecer el algoritmo a utilizar actualizándose sus parámetros. |
| Prerrequisito | Ninguno. |
| Datos de entrada | Ninguno. |
| Pasos | **1.** Seleccionar un algoritmo. |
| Resultado esperado | Se actualizarán los parámetros con los valores predefinidos según el algoritmo. |
| Resultado obtenido | Aparecen los parámetros con los valores predefinidos según el algoritmo. |
| Resultado de la prueba | Positivo. |

Tabla 6.6: Prueba de caja negra 06; visualización de los parámetros de un algoritmo.

| PCN-07 | Ejecución de un algoritmo |
|---|---|
| Propósito | Ejecutar un algoritmo según los datos introducidos. |
| Prerrequisito | Haber seleccionado al menos dos columnas en el panel de selección de características. |
| Datos de entrada | Algoritmo modificado, según las columnas seleccionadas y las posibles filas eliminadas. |
| Pasos | **1.** Seleccionar un algoritmo.<br>**2.** Establecer los parámetros del algoritmo.<br>**3.** Seleccionar el porcentaje de división del dataset.<br>**4.** Presionar el botón ejecutar. |
| Resultado esperado | En el panel de resultados aparecerán las métricas de evaluación de los resultados, justo debajo aparecerá un gráfico con las predicciones y la columna predicha. |
| Resultado obtenido | En el panel de resultados aparecen las métricas de evaluación de los resultados, justo debajo aparecerá un gráfico con las predicciones y la columna predicha. |
| Resultado de la prueba | Positivo |

Tabla 6.7: Prueba de caja negra 07, ejecución de un algoritmo.

| PCN-08 | Selección de las columnas a visualizar en el gráfico de resultados |
|---|---|
| Propósito | Visualizar un gráfico con las columnas seleccionadas. |
| Prerrequisito | Haber ejecutado correctamente un algoritmo. |
| Datos de entrada | Dataset de resultados. |
| Pasos | **1.** Seleccionar una columna. |
| Resultado esperado | Se mostrará un gráfico con las columnas seleccionadas. |
| Resultado obtenido | Gráfico de las columnas seleccionadas. |
| Resultado de la prueba | Positivo. |

Tabla 6.8: Prueba de caja negra 08, selección de las columnas a visualizar en el gráfico de resultados.

| PCN-09 | Añadir un dataset a la pestaña Experimentos |
|---|---|
| Propósito | Añadir un dataset. |
| Prerrequisito | Ninguno. |
| Datos de entrada | Dataset introducido. |
| Pasos | **1.**Introducir un dataset.<br>**2.** Seleccionar un porcentaje de división.<br>**3.** Seleccionar la columna a predecir.<br>**4.** Pulsar el botón añadir. |
| Resultado esperado | Se añadirá al panel datasets una opción con el nombre del dataset. |
| Resultado obtenido | Aparece en el panel datasets una opción con el nombre del dataset. |
| Resultado de la prueba | Positivo. |

Tabla 6.9: Prueba de caja negra 09, añadir un dataset a la pestaña Experimentos.

| PCN-10 | Eliminar un dataset en la pestaña Experimentos |
|---|---|
| Propósito | Eliminar un dataset del panel datasets. |
| Prerrequisito | Haber añadido un dataset. |
| Datos de entrada | Ninguno. |
| Pasos | **1.** Seleccionar un dataset.<br>**2.** Pulsar el botón eliminar. |
| Resultado esperado | El dataset desaparecerá del panel datasets. |
| Resultado obtenido | El dataset desaparece del panel datasets. |
| Resultado de la prueba | Positivo. |

Tabla 6.10: Prueba de caja negra 10, eliminar un dataset en la pestaña Experimentos.

| PCN-11 | Añadir un algoritmo a la pestaña Experimentos |
|---|---|
| Propósito | Añadir un algoritmo. |
| Prerrequisito | Ninguno. |
| Datos de entrada | Ninguno. |
| Pasos | **1.**Seleccionar un algoritmo.<br>**2.** Establecer los parámetros del algoritmo.<br>**3.** Pulsar el botón añadir. |
| Resultado esperado | Se añadirá al panel algoritmos una opción con el nombre del algoritmo. |
| Resultado obtenido | Aparece en el panel algoritmos una opción con el nombre del algoritmo. |
| Resultado de la prueba | Positivo. |

Tabla 6.11: Prueba de caja negra 11, añadir un algoritmo a la pestaña Experimentos.

| PCN-12 | Eliminar un algoritmo en la pestaña Experimentos |
|---|---|
| Propósito | Eliminar un algoritmo del panel algoritmos. |
| Prerrequisito | Haber añadido un algoritmo. |
| Datos de entrada | Ninguno. |
| Pasos | **1.** Seleccionar un algoritmo.<br>**2.** Pulsar el botón eliminar. |
| Resultado esperado | El algoritmo desaparecerá del panel algoritmos. |
| Resultado obtenido | El algoritmo desaparecerá del panel algoritmos. |
| Resultado de la prueba | Positivo. |

Tabla 6.12: Prueba de caja negra 12, eliminar un algoritmo en la pestaña Experimentos.

| PCN-13 | Visualizar los resultados de los experimentos |
|---|---|
| Propósito | Poder ver los diferentes resultados de los experimentos. |
| Prerrequisito | Haber ejecutado un experimento. |
| Datos de entrada | Resultados de los experimentos. |
| Pasos | **1.** Seleccionar un resultado de la lista de resultados. |
| Resultado esperado | A la derecha de visualizarán las métricas de error, a la izquierda los parámetros del algoritmo y debajo un gráfico con la columna predicha y las predicciones. |
| Resultado obtenido | A la derecha se visualizan las métricas de error, a la izquierda los parámetros del algoritmo y debajo un gráfico con la columna predicha y las predicciones. |
| Resultado de la prueba | Positivo. |

Tabla 6.13: Prueba de caja negra 13, visualizar los resultados de los experimentos.

| PCN-14 | Selección de las columnas a visualizar en el gráfico de resultados de los Experimentos |
|---|---|
| Propósito | Visualizar un gráfico con las columnas seleccionadas. |
| Prerrequisito | Haber ejecutado correctamente los experimentos. |
| Datos de entrada | Dataset de resultados. |
| Pasos | **1.** Seleccionar una columna. |
| Resultado esperado | Se mostrará un gráfico con las columnas seleccionadas. |
| Resultado obtenido | Gráfico de las columnas seleccionadas. |
| Resultado de la prueba | Positivo. |

Tabla 6.14: Prueba de caja negra 14, selección de las columnas a visualizar en el gráfico de resultados de los Experimentos.

| PCN-15 | Ver los detalles de un punto del gráfico de resultados |
|---|---|
| Propósito | Poder visualizar los detalles de un punto del gráfico de resultados. |
| Prerrequisito | visualización del gráfico de los resultados. |
| Datos de entrada | Dataset de resultados. |
| Pasos | **1.** Poner el ratón sobre un punto. |
| Resultado esperado | Se mostrará un cuadro con los detalles del punto del gráfico. |
| Resultado obtenido | Se muestra un cuadro con los detalles del punto del gráfico. |
| Resultado de la prueba | Positivo. |

Tabla 6.15: Prueba de caja negra 15, ver los detalles de un punto del gráfico de resultados.

# Capítulo 7

# Resultados

Este capítulo documentaremos los resultados de los experimentos realizados, con el fin de extraer conclusiones y ver con que precisión son capaces de predecir los modelos generados. Para la ejecución de estos experimentos utilizaremos los tres algoritmos implementados variando sus parámetros, los cueles vienen descritos en el apartado 4.4.1, para poder ver que configuración arroja mejores resultados y por lo tanto se adapta más a la predicción del tiempo de llegada de los vuelos.

La organización de este capítulo consta de cuatro secciones, las tres primeras representan los distintos datasets con los que se han hecho los experimentos. Todos los dataset han sido creados mediante la aplicación siguiendo la sección C.2 del manual de usuario, presente en el apéndice C. En todos los experimentos se toma como columna de predicción *touchdown*, que representa los tiempos reales de llegada de los vuelos y se emplea un 65 % de los datos para entrenamiento y un 35 % para probar el modelo. Por ultimo tendremos una sección donde se analizaran los resultados de los mejores modelos obtenidos por cada algoritmo.

## 7.1. Dataset de vuelos con destino a Barajas el 02-02-2018

En esta sección vamos a trabajar con los datos del archivo csv pruebas. Este dataset consta de las columnas *take_off*, *tiempo_inicio_leg*, *tiempo_final_leg*, *operator*, *aircrafttype*, *timestamp* y *airport_origin*. El resto de las columnas presentes en el dataset resultante del programa de preprocesado de datos, presente en la sección 5.2, no se han tenido en cuenta ya que no aportaban información útil para los algoritmos de predicción.

#### Experimento 1

En este experimento vamos a utilizar el algoritmo de árbol de decisión, modificando el parámetro número de contenedores o maxBins, que tomará los valores 150, 300 y 500. Este parámetro determinará el número de divisiones que se produce por cada columna o característica. Dependiendo de si la columna es de tipo continuo o categórica, se dividirán de una forma u otra. En nuestro caso las de tipo continuo serán las columnas *take_off*, *tiempo_inicio_leg* y *tiempo_final_leg*, en datasets con grandes volúmenes de información la división se realizara te-

niendo en cuenta las divisiones realizadas mediante el cálculo de cuantiles sobre una fracción muestreada de los datos. Las columnas categóricas serán *operator*, *aircrafttype* y *airport_origin*, se dividirán según sus categorías.

El número máximo de contenedores debe ser siempre como mínimo el número de máximo de categorías que pueda tener cualquiera de las columnas categóricas, por lo que en nuestros experimentos el valor mínimo es de 150 ya que la columna *aircrafttype* tiene 146 categorias. Cuanto mayor será número de contenedores el algoritmo podrá realizar mayor número de divisiones y por tanto tendrán más nivel de detalle, aunque esto aumenta la computación y la comunicación entre los nodos. De ninguna forma el número máximo de contenedores podrá sobrepasar el número de filas del dataset.

En las siguientes tablas vemos como a mayor número de contenedores las cuatro métricas mejoran sus valores, coincidiendo con la afirmación de que a mayor número de divisiones se obtendrá un análisis más preciso. El valor con el que nos quedaremos para el número máximo de contenedores será de 500.

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 13.332,010231033064 |
| Error cuadrático medio (MSE) | 177.742.496,8003703 |
| Error absoluto medio (MAE) | 6.122,1155425626675 |
| R al cuadrado (R2) | 0,4829124204891104 |

Tabla 7.1: Experimento 1.1

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 300 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 13334.601832114988 |
| Error cuadrático medio (MSE) | 177.811.606,0210444 |
| Error absoluto medio (MAE) | 6.114,120157768497 |
| R al cuadrado (R2) | 0,48271136829131533 |

Tabla 7.2: Experimento 1.2

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 13.329,178675566378 |
| Error cuadrático medio (MSE) | 177.667.004,16517353 |
| Error absoluto medio (MAE) | 6.083,386503942942 |
| R al cuadrado (R2) | 0,48313204328455317 |

Tabla 7.3: Experimento 1.3

**Experimento 2**

En este experimento vamos a utilizar el algoritmo de árbol de decisión, modificando el parámetro profundidad máxima que tomará los valores de 2 y 8, junto con la configuración que mejor resultado obtuvo en el experimento anterior con profundidad 5, correspondiente al experimento 1.3.

El parámetro de profundidad máxima o maxDepth nos indicará el número máximo de niveles que podrá llegar a tener nuestro árbol, a mayor profundidad se conseguirá una mayor precisión, pero también aumentará el costo computacional y se podrá llegar a sobreajustar los resultados.

En las siguientes tablas, junto con el experimento 1.3, vemos como resultados empeoran cuando la profundidad es mayor para todas las métricas, por lo que nos quedaremos con el valor 2 para la profundidad máxima, correspondiente al experimento 2.1.

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.109,114743356952 |
| Error cuadrático medio (MSE) | 50.539.512,43421518 |
| Error absoluto medio (MAE) | 5.131,984563906987 |
| R al cuadrado (R2) | 0,852970704110239 |

Tabla 7.4: Experimento 2.1

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 8 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 13.283,825800141465 |
| Error cuadrático medio (MSE) | 176.460.027,88850406 |
| Error absoluto medio (MAE) | 6.074,622504709804 |
| R al cuadrado (R2) | 0,4866433726102065 |

Tabla 7.5: Experimento 2.2

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 8.128,5233271321 |
| Error cuadrático medio (MSE) | 66.072.891,47973073 |
| Error absoluto medio (MAE) | 5.041,274113566188 |
| R al cuadrado (R2) | 0,8077810757609195 |

Tabla 7.6: Experimento 2.3

### Experimento 3

En este experimento vamos a utilizar el algoritmo de árbol de decisión, modificando el parámetro de instancias por nodo o minInstancesPerNode, junto con la configuración que mejor resultado obtuvo en el experimento anterior, correspondiente con el experimento 2.1.

Este parámetro afecta a la división de un nodo, ya que especifica el número mínimo de filas o instancias de entrenamiento que deberá recibir cada uno de sus hijos. Tiene como valor predeterminado el 1, será de gran importancia que el valor sea mayor que 1 ya que a la hora de evaluar el error que se obtiene en cada hoja si solo se dispone de una instancia puede llevar a que los resultados obtenidos de evaluar esa hoja no sean fiables.

En la siguiente tablas, vemos como este parámetro no modifica los resultados.

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 2 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.109,114743356952 |
| Error cuadrático medio (MSE) | 50.539.512,43421518 |
| Error absoluto medio (MAE) | 5.131,984563906987 |
| R al cuadrado (R2) | 0,8529707041100829 |

Tabla 7.7: Experimento 3.1

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 4 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.109,114743356952 |
| Error cuadrático medio (MSE) | 50.539.512,43421518 |
| Error absoluto medio (MAE) | 5.131,984563906987 |
| R al cuadrado (R2) | 0,8529707041100829 |

Tabla 7.8: Experimento 3.2

**Experimento 4**

En este experimento vamos a utilizar el algoritmo de árbol de decisión, modificando el parámetro de ganancia de información o minInfoGain. Junto con la configuración que mejor resultado obtuvo en el experimento anterior, correspondiente al experimento 2.1.

Este parámetro se calculara mediante la diferencia de la impureza del nodo padre y de la suma ponderada de la impureza de los nodos hijos, en nuestro caso la impureza viene determinada por la varianza. Esto quiere decir que cuanto mayor sea este parámetro mejor tendrán que ser los resultados obtenidos por los nodos para poder dividirse.

En la siguiente tabla, vemos como este parámetro no modifica los resultados.

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 1 |
| Ganancia de información | 0.1 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.109,114743356952 |
| Error cuadrático medio (MSE) | 50.539.512,43421518 |
| Error absoluto medio (MAE) | 5.131,984563906987 |
| R al cuadrado (R2) | 0,8529707041100829 |

Tabla 7.9: Experimento 4.1

**Experimento 5**

En este experimento vamos a utilizar el algoritmo de Random Forest, modificando el parámetro de número de profundidad máxima o maxDepth, junto con la configuración de los parámetros que viene por defecto, excepto el parámetro de numero de arboles que será 150.

En el experimento 2 describimos en que consiste este parámetro y como el aumento del mismo puede llevar a un sobreajuste. Con el algoritmo de Random Forest, al entrenar varios árboles y hacer un promedio de las predicciones de cada árbol se consigue reducir este sobreajuste y nos permite utilizar arboles más profundos.

En las siguientes tablas vemos como el valor que mejor resultados arroja es 3, correspondiente al experimento 5.4, ya que si la profundidad es mayor los resultados empeoran. En el caso de los Arboles de Decisión o Decision Tree la profundidad máxima que mejores resultados obtenía es 2, como hemos comentado anteriormente en el caso de Random Forest se pueden obtener buenos resultados con árboles más profundos.

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 20 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 8.416,34354696207 |
| Error cuadrático medio (MSE) | 70.833.422,90944076 |
| Error absoluto medio (MAE) | 5.831,7736154424265 |
| R al cuadrado (R2) | 0,7912512323609303 |

Tabla 7.10: Experimento 5.1

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 20 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 8.303,460192692988 |
| Error cuadrático medio (MSE) | 68.943.658,20206725 |
| Error absoluto medio (MAE) | 5.392,3810925106545 |
| R al cuadrado (R2) | 0,7991128195336196 |

Tabla 7.11: Experimento 5.2

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 8 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 20 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 8.769,117941689927 |
| Error cuadrático medio (MSE) | 76.898.222,18565923 |
| Error absoluto medio (MAE) | 5.855,223079933285 |
| R al cuadrado (R2) | 0,7759793919716977 |

Tabla 7.12: Experimento 5.3

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 20 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 8.072,342582542779 |
| Error cuadrático medio (MSE) | 65.163.946,5805138 |
| Error absoluto medio (MAE) | 5.381,40140243304 |
| R al cuadrado (R2) | 0,8193323740002017 |

Tabla 7.13: Experimento 5.4

**Experimento 6**

En este experimento vamos a utilizar el algoritmo de Random Forest, modificando el parámetro de número de árboles o numTrees, junto con la configuración que mejor resultado obtuvo en los experimentos anteriores, correspondiente al experimento 5.4.

Este parámetro indica el número de árboles que utilizará el algoritmo. Cuanto mayor sea este número más decrecerá la varianza en las predicciones, también hará que el tiempo de entrenamiento aumente.

En las siguientes tablas vemos como el parámetro que mejor resultados obtiene es 5, correspondiente al experimento 6.4. Normalmente los resultados mejoran a mayor número de árboles, pero en nuestro caso llegara un momento en el la creación de más arboles hará que los datos se dupliquen entre las diferentes muestras. Esto puede conllevar a que entre los datos que se duplican en los diferentes arboles haya outliers o valores atípicos, por lo que el incremento del número de árboles hace que empeoren los resultados.

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 25 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.660,650708892278 |
| Error cuadrático medio (MSE) | 58.686.936,98904576 |
| Error absoluto medio (MAE) | 4.904,81579896151 |
| R al cuadrado (R2) | 0,8297092828938361 |

Tabla 7.14: Experimento 6.1

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 15 |
| División de datos | 1 |
| División de datos | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 8.028,54501020212 |
| Error cuadrático medio (MSE) | 64.449.392,71035357 |
| Error absoluto medio (MAE) | 5.288,372901369623 |
| R al cuadrado (R2) | 0,8157224528259982 |

Tabla 7.15: Experimento 6.2

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 10 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.844,686558129345 |
| Error cuadrático medio (MSE) | 61.531.450,60890544 |
| Error absoluto medio (MAE) | 5.197,67921283959 |
| R al cuadrado (R2) | 0,8296939373974515 |

Tabla 7.16: Experimento 6.3

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 5 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 6.579,516006779781 |
| Error cuadrático medio (MSE) | 43.292.606,083837036 |
| Error absoluto medio (MAE) | 4.532,2712658179007 |
| R al cuadrado (R2) | 0,8719484395116373 |

Tabla 7.17: Experimento 6.4

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 3 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.597,716823818272 |
| Error cuadrático medio (MSE) | 57.722.151,8608932 |
| Error absoluto medio (MAE) | 5.012,048622365725 |
| R al cuadrado (R2) | 0,8365672008107406 |

Tabla 7.18: Experimento 6.5

### Experimento 7

En este experimento vamos a utilizar el algoritmo de Random Forest, modificando el parámetro de división de datos o subsamplingRate, junto con la configuración que mejor resultado obtuvo en los experimentos anteriores, correspondiente al experimento 6.4.

Este parámetro especifica la fracción del tamaño original del dataset que se utilizará para entrenar cada árbol. Cuanto menor sea esa fracción más se puede reducir el tiempo necesario para el entrenamiento. En ambas tablas utilizamos el valor de 0,5 o lo que es lo mismo utilizaremos divisiones de la mitad del tamaño del dataset original para entrenar cada uno de los árboles.

En las siguientes tablas, vemos como el parámetro que mejor resultados obtiene es el que viene por defecto 0,5. En la segunda parte del experimento intentamos aumentar el número de árboles, teniendo en cuenta que si se cogen solo la mitad de los datos habrá menos riesgo de que estén presentes valores atípicos, pero los resultados empeoran.

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 5 |
| División de datos | 0,5 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.008,870950946008 |
| Error cuadrático medio (MSE) | 49.117.277,97670505 |
| Error absoluto medio (MAE) | 4.972,651112809693 |
| R al cuadrado (R2) | 0,8594433878991312 |

Tabla 7.19: Experimento 7.1

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 15 |
| División de datos | 0,5 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 9.240,050392627123 |
| Error cuadrático medio (MSE) | 85.394.055,66947948 |
| Error absoluto medio (MAE) | 6.177,76534339533 |
| R al cuadrado (R2) | 0,7589908955148872 |

Tabla 7.20: Experimento 7.2

### Experimento 8

En este experimento vamos a utilizar el algoritmo de Gradient-boosted Tree, modificando el parámetro de iteraciones máximas o numIterations, junto con la configuración que viene por defecto, excepto el parámetro de número de árboles que será 150.

Este parámetro determinara el número de árboles que utilizara el algoritmo, ya que cada iteración corresponde a un árbol. El aumento de este número hará que el modelo sea más expresivo mejorando los datos de entrenamiento, pero también hará que aumente el tiempo de entrenamiento.

En las siguientes tablas, vemos como el parámetro que mejor resultados obtiene es el valor 5, correspondiente al experimento 8.4. Normalmente el aumento del número de iteraciones mejora los resultados, en nuestro caso al tener datos ruidosos el método para actualizar las probabilidades puede estar enfatizando demasiado los valores atípicos u outliers.

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 20 |
| Función de pérdida | squared |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 15.231,293272165112 |
| Error cuadrático medio (MSE) | 231.992.294,74270222 |
| Error absoluto medio (MAE) | 7.357,226485463859 |
| R al cuadrado (R2) | 0,3250891806229209 |

Tabla 7.21: Experimento 8.1

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 15 |
| Función de pérdida | squared |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 15.046,235838942986 |
| Error cuadrático medio (MSE) | 226.389.212,92109236 |
| Error absoluto medio (MAE) | 7.260,578376772713 |
| R al cuadrado (R2) | 0,34138963813408474 |

Tabla 7.22: Experimento 8.2

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 10 |
| Función de pérdida | squared |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 14.697,893558306097 |
| Error cuadrático medio (MSE) | 216.028.075,05129588 |
| Error absoluto medio (MAE) | 7.070,503533333708 |
| R al cuadrado (R2) | 0,3715322084163858 |

Tabla 7.23: Experimento 8.3

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 5 |
| Función de pérdida | squared |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 14.081,684861818729 |
| Error cuadrático medio (MSE) | 198.293.848,54757476 |
| Error absoluto medio (MAE) | 6.626,100112159374 |
| R al cuadrado (R2) | 0,42312453114391735 |

Tabla 7.24: Experimento 8.4

**Experimento 9**

En este experimento vamos a utilizar el algoritmo de Gradient-boosted Tree, modificando el parámetro de función de pérdida o loss, junto con la configuración que mejor resultado obtuvo en los experimentos anteriores, correspondiente al experimento 8.4.

Este parámetro es de suma importancia ya que determina si las instancias obtienen un peor o mejor resultado, haciendo que se ponga un mayor énfasis en cada iteración en las instancias que peor resultado obtuvieron. En nuestro caso al ser aplicable a la regresión se tendrán dos funciones de perdida, el error al cuadrado o L2 (MSE) y el error absoluto o L1 (MAE).

En las siguientes tablas, vemos como la modificación parámetro nos hace obtener mejores resultados. Esto se debe a que el error cuadrado o L2 aumenta mucho si el error es mayor a 1 haciendo que tomen mucha importancia los valores atípicos, por lo que en nuestro caso será más apropiado utilizar el error absoluto o L1.

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 5 |
| Función de pérdida | absolute |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 13.332,085433401946 |
| Error cuadrático medio (MSE) | 177.744.502,00352836 |
| Error absoluto medio (MAE) | 6.122,165845452056 |
| R al cuadrado (R2) | 0,48290658696258937 |

Tabla 7.25: Experimento 9.1

**Experimento 10**

En este experimento vamos a utilizar el algoritmo de Gradient-boosted Tree, modificando el parámetro de número de características o featureSubsetStrategy, junto con la configuración que mejor resultado obtuvo en los experimentos anteriores, correspondiente al experimento 9.1.

Este parámetro selecciona el número de columnas o características que se van a considerar para la división de cada nodo. El valor “auto.[es]cogerá la opción “all"(utiliza todas las características) si el número de árboles es 1, si el número de árboles es mayor a uno “onethird"(utiliza 1/3 de las características).

En las siguientes tablas, vemos como la modificación parámetro al valor “auto", nos da los mismos resultados.

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 5 |
| Función de pérdida | absolute |
| Número de características | auto |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 13.332,085433401946 |
| Error cuadrático medio (MSE) | 177.744.502,00352836 |
| Error absoluto medio (MAE) | 6.122,165845452056 |
| R al cuadrado (R2) | 0,48290658696258937 |

Tabla 7.26: Experimento 10.1

### Experimento 11

En este experimento vamos a utilizar el algoritmo de Gradient-boosted Tree, modificando el parámetro de profundidad máxima o maxDepth, junto con la configuración que mejor resultado obtuvo en los experimentos anteriores, correspondiente al experimento 9.1. Este parámetro fue explicado en anteriores experimentos, vemos como mejoran los resultados notablemente.

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 5 |
| Función de pérdida | absolute |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.072,495613137649 |
| Error cuadrático medio (MSE) | 50.020.194,197851285 |
| Error absoluto medio (MAE) | 5.103,511746771592 |
| R al cuadrado (R2) | 0,8544815021167064 |

Tabla 7.27: Experimento 11.1

## 7.2. Datasets de vuelos filtrados con destino a Barajas el 02-02-2018

El conjunto de datos anterior presenta outliers o valores atípicos, esto es debido a que en algunos vuelos pasaron por localizaciones donde los sistemas ADS-B carecían de estaciones receptoras, lo que provoco la pérdida de mensajes. Estos datos pueden generar confusión en los algoritmos a la hora de establecer un modelo de predicción por lo que se ha decidido eliminar algunos vuelos. A la hora de establecer los parámetros de los algoritmos nos fijaremos en las configuraciones que mejor resultados obtuvieron en los experimentos de la sección anterior. Primeramente haremos tres experimentos (12, 13 y 14) con un dataset al que se le han eliminado los siguientes vuelos:

- tiempo_inicio_leg: 2018-02-02 22:50:00 – airport_origin: GOBD
- tiempo_inicio_leg: 2018-02-02 01:15:00 – airport_origin: SBGR
- tiempo_inicio_leg: 2018-02-02 17:40:00 – airport_origin: SUMU
- tiempo_inicio_leg: 2018-02-02 10:50:00 – airport_origin: OMDB
- tiempo_inicio_leg: 2018-02-02 08:53:00 – airport_origin: EBBR
- tiempo_inicio_leg: 2018-02-02 04:40:00 – airport_origin: MUHA
- tiempo_inicio_leg: 2018-02-02 08:51:00 – airport_origin: LFPG

Más tarde haremos otros tres experimentos (15, 16 y 17) utilizando otro dataset al que se le han eliminado 5 vuelos más respecto al anterior. Los vuelos eliminados son:

- tiempo_inicio_leg: 2018-02-02 00:15:00 – airport_origin: KPHL
- tiempo_inicio_leg: 2018-02-02 14:16:00 – airport_origin: LTBA
- tiempo_inicio_leg: 2018-02-02 10:54:00 – airport_origin: KTEB
- tiempo_inicio_leg: 2018-02-02 12:05:00 – airport_origin: EFHK
- tiempo_inicio_leg: 2018-02-02 12:40:00 – airport_origin: LROP

**Experimento 12**

En este experimento utilizaremos la configuración del algoritmo de Decision Tree o Árbol de Decisión correspondiente al experimento 2.1, el cual obtuvo mejores resultados. Vemos que lejos de mejorar los resultados, la eliminación de estos vuelos hace que se obtengan peores resultados.

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.635,01034523824 |
| Error cuadrático medio (MSE) | 58.293.382,97189494 |
| Error absoluto medio (MAE) | 5.447,3203076916625 |
| R al cuadrado (R2) | 0,8394589179554953 |

Tabla 7.28: Experimento 12.1

**Experimento 13**

En este experimento en una primera parte utilizaremos la configuración del algoritmo de Random Forest correspondiente al experimento 6.4, el cual obtuvo mejores resultados. En la segunda parte del experimento, probaremos a aumentar el número de árboles. Podemos ver como ambos experimentos han obtenido peores resultados que con el dataset original.

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 5 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 6.816,991064024434 |
| Error cuadrático medio (MSE) | 46.463.058,00033659 |
| Error absoluto medio (MAE) | 4.478,978944451823 |
| R al cuadrado (R2) | 0,8725172782775201 |

Tabla 7.29: Experimento 13.1

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 10 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 8.410,371908198926 |
| Error cuadrático medio (MSE) | 70.737.327,93257513 |
| Error absoluto medio (MAE) | 5.111,5275727448434 |
| R al cuadrado (R2) | 0,8032014640782677 |

Tabla 7.30: Experimento 13.2

**Experimento 14**

En este experimento en una primera parte utilizaremos la configuración del algoritmo de Gradient-boosted Tree correspondiente al experimento 11.1, el cual obtuvo mejores resultados. En la segunda parte del experimento, probaremos a aumentar el número de iteraciones y la profundidad.

Podemos ver como el experimento 14.1 han obtenido peores resultados que el 11.1. Lo mismo ha ocurrido con la segunda parte del experimento la 14.2, aunque se han conseguido mejorar los resultados de la primera parte como era de esperar, ya que reducir los valores atípicos nos permitiría mejorar los resultados obtenidos añadiendo más iteraciones al algoritmo y un nivel más de profundidad.

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 5 |
| Función de pérdida | absolute |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.668,190496062982 |
| Error cuadrático medio (MSE) | 58.801.145,48391064 |
| Error absoluto medio (MAE) | 5.432,60191368781 |
| R al cuadrado (R2) | 0,8380605303693793 |

Tabla 7.31: Experimento 14.1

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 10 |
| Función de pérdida | absolute |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.498,768629943387 |
| Error cuadrático medio (MSE) | 56.231.530,96542304 |
| Error absoluto medio (MAE) | 4.555,893977799167 |
| R al cuadrado (R2) | 0,8451372974774685 |

Tabla 7.32: Experimento 14.2

**Experimento 15**

En este experimento se evaluaran el algoritmo Decision Tree con las mismas configuraciones del experimento 12. Los resultados son prácticamente igual a los obtenidos al segundo dataset, siendo por lo tanto también peores que los obtenidos con el dataset original.

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 6.899,448496411935 |
| Error cuadrático medio (MSE) | 47.602.389,55464091 |
| Error absoluto medio (MAE) | 4.772,63489447762 |
| R al cuadrado (R2) | 0,8618042284368501 |

Tabla 7.33: Experimento 15.1

### Experimento 16

En este experimento se evaluaran el algoritmo Random Forest con las mismas configuraciones del experimento 6.4. La primera parte del experimento muestra peores resultados en ambos casos.

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 5 |
| División de datos | 1 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.592,991064024434 |
| Error cuadrático medio (MSE) | 57.648.988,00033659 |
| Error absoluto medio (MAE) | 4.847,978944451823 |
| R al cuadrado (R2) | 0,8365172782775201 |

Tabla 7.34: Experimento 16.1

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 10 |
| División de datos | 0,5 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 8.163,371908198926 |
| Error cuadrático medio (MSE) | 66.636.925,93257513 |
| Error absoluto medio (MAE) | 5.454,5275727448434 |
| R al cuadrado (R2) | 0,8082014640782677 |

Tabla 7.35: Experimento 16.2

### Experimento 17

En este experimento se evaluaran el algoritmo Gradient-boosted Tree con las mismas configuraciones del experimento 14. Los resultados obtenidos han mejorado a los experimentos hechos con los dos datasets anteriores, destacando el experimento 17.1.

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 2 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 5 |
| Función de pérdida | absolute |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 6.885.354221896142 |
| Error cuadrático medio (MSE) | 47.408.102,76098303 |
| Error absoluto medio (MAE) | 4.771,904793686025 |
| R al cuadrado (R2) | 0,8623682676291305 |

Tabla 7.36: Experimento 17.1

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 3 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 10 |
| Función de pérdida | absolute |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 7.564,2167989805785 |
| Error cuadrático medio (MSE) | 57.217.375,781979986 |
| Error absoluto medio (MAE) | 4.673,045524851013 |
| R al cuadrado (R2) | 0,8338907045005998 |

Tabla 7.37: Experimento 17.2

## 7.3. Dataset de vuelos filtrados con destino a Barajas el 02-02-2018 (eliminación de características poco significativas)

Por ultimo vamos a utilizar el segundo dataset generado en la seccion anterior, al que le hemos eliminado las columnas de las características que que menos información proporcionan a los algoritmos, estas columnas son: *operator*, *aircrafttype* y *airport_origin*.

### Experimento 18

En este experimento utilizaremos el algoritmo de Decision Tree variando el parámetro de profundidad máxima. Vemos como los resultados mejoran notablemente respecto a los anteriores experimentos. Cabe destacar que los resultados mejoran en la segunda tabla respecto a la primera al aumentar la profundidad máxima.

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 5 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 2.165,2381033142224 |
| Error cuadrático medio (MSE) | 4.688.256,044043771 |
| Error absoluto medio (MAE) | 1.489,9928323128645 |
| R al cuadrado (R2) | 0,9860527685135974 |

Tabla 7.38: Experimento 18.1

| Algoritmo | Decision tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 15 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 1.594,394403287748 |
| Error cuadrático medio (MSE) | 2.542.093,513235294 |
| Error absoluto medio (MAE) | 1.041,2735294117647 |
| R al cuadrado (R2) | 0,9924374508652924 |

Tabla 7.39: Experimento 18.2

**Experimento 19**

En este experimento utilizaremos el algoritmo de Random Forest variando los dos parámetros que hemos visto anteriormente más relevantes y aumentando la profundidad a 20. Estos parámetros son el número de características y el número de árboles. Vemos como los resultados mejoran notablemente respecto a los anteriores experimentos. Cabe destacar que los resultados mejoran a mayor número de árboles y si se emplean todas las características en cada árbol.

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 20 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 20 |
| División de datos | 0,5 |
| Número de características | auto |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 2.363,5669368399417 |
| Error cuadrático medio (MSE) | 5.586.448,664922945 |
| Error absoluto medio (MAE) | 1.428,708640053693 |
| R al cuadrado (R2) | 0,983380708735914 |

Tabla 7.40: Experimento 19.1

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 20 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 20 |
| División de datos | 0,5 |
| Número de características | all |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 2.053,794173607887 |
| Error cuadrático medio (MSE) | 4.218.070,507545703 |
| Error absoluto medio (MAE) | 1.135,9331344492296 |
| R al cuadrado (R2) | 0,987451537364423 |

Tabla 7.41: Experimento 19.2

| Algoritmo | Random forest |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 20 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 150 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Número de arboles | 30 |
| División de datos | 0,5 |
| Número de características | all |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 1.821,8562148119406 |
| Error cuadrático medio (MSE) | 3.319.160,0674488917 |
| Error absoluto medio (MAE) | 1.045,6374808591954 |
| R al cuadrado (R2) | 0,9901257326037158 |

Tabla 7.42: Experimento 19.3

**Experimento 20**

En este experimento utilizaremos el algoritmo de Gradient-boosted Tree variando los dos parámetros que hemos visto anteriormente más relevantes y aumentando la profundidad a 20. Estos parámetros son el número de iteraciones y la función de perdida. Vemos como los resultados mejoran notablemente respecto a los anteriores experimentos, siendo prácticamente los mismos para todas las tablas del experimento y que los obtenidos por el experimento 18.2.

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 20 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 20 |
| Función de pérdida | squared |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 1.594,3932185874505 |
| Error cuadrático medio (MSE) | 2.542.089,7354776496 |
| Error absoluto medio (MAE) | 1.041,270630624715 |
| R al cuadrado (R2) | 0,9924374621038555 |

Tabla 7.43: Experimento 20.1

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 20 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 20 |
| Función de pérdida | absolute |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 1.594,4036662194296 |
| Error cuadrático medio (MSE) | 2.542.123,0508539584 |
| Error absoluto medio (MAE) | 1.041,2500000224395 |
| R al cuadrado (R2) | 0,9924373629929563 |

Tabla 7.44: Experimento 20.2

| Algoritmo | Gradient-boosted Tree |
|---|---|
| Parámetros | |
| Impuereza | variance |
| Nodos en caché | False |
| Profundidad máxima | 20 |
| Instancias por nodo | 1 |
| Ganancia de información | 0,0 |
| Número de contenedores | 500 |
| Memoria máxima | 256 |
| Frecuencia de almacenamiento | 10 |
| Iteraciones máximas | 25 |
| Función de pérdida | absolute |
| Número de características | all |
| Contribución de estimadores | 0,1 |
| Resultados | |
| Raíz del error cuadrático medio (RMSE) | 1.594,4073411980398 |
| Error cuadrático medio (MSE) | 2.542.134,769666203 |
| Error absoluto medio (MAE) | 1.041,2288235720466 |
| R al cuadrado (R2) | 0,9924373281303153 |

Tabla 7.45: Experimento 20.3

## 7.4. Análisis de los resultados

Tras la realización de todos los experimentos hemos visto como para los datasets de las secciones 7.1 y 7.2, todos los algoritmos proporcionaban mejores resultados con árboles con niveles de profundidad muy bajos. Esto es debido a que las características categóricas *operator*, *aircrafttype* y *airport_origin*, aportaban información negativa al algoritmo, de tal manera que cuando la profundidad de los árboles crecía e intervenían estas características en las decisiones los resultados el modelo generado era mucho más impreciso. Una vez aisladas las características que aportaban información predictiva útil, junto con el borrado de algunos vuelos que obtenían peores resultados de predicción se generó un nuevo dataset. Al ejecutar los algoritmos con este nuevo dataset se han podido generar arboles mucho más expresivos, mejorando notablemente los resultados.

A continuación se muestran tres gráficos, correspondientes a los datos de predicción generados por el mejor modelo creado a partir de nuestros tres algoritmos utilizados. En el eje X del gráfico tendremos los valores de aterrizaje reales de los vuelos obtenidos de los mensajes ADS-B, mientras que el eje Y representará las predicciones sobre el tiempo de aterrizaje de los vuelos obtenidas por los modelos generados. El gráfico de la Figura 7.1 corresponde al modelo generado por el experimento 18.2, en él se utilizaba el algoritmo de Decision Tree. El gráfico de la Figura 7.2 corresponde al modelo generado por el experimento 19.2, en él se utilizaba el algoritmo de Random Forest. Por último, el gráfico de la Figura 7.3 corresponde al modelo generado por el experimento 20.1, en él se utilizaba el algoritmo de Gradient-boosted Tree. Todas las gráficas tienen una tendencia lineal, es la situación ideal esperada, ya que denota que las predicciones son muy similares a los valores reales. Los tres gráficos son muy similares, se nota una leve diferencia en uno de los puntos de la esquila superior derecha, habiendo una predicción que se aleja más de los resultados reales, la cual es mejor predicha por el algoritmo de Radom Forest. Cabe destacar la similitud entre el primer y tercer gráfico, esto sucede porque el algoritmo de Gradient-boosted Tree no es capaz de optimizar más los arboles de decisión en cada iteración, teniendo los mismos resultados que el modelo generado por Decison Tree.

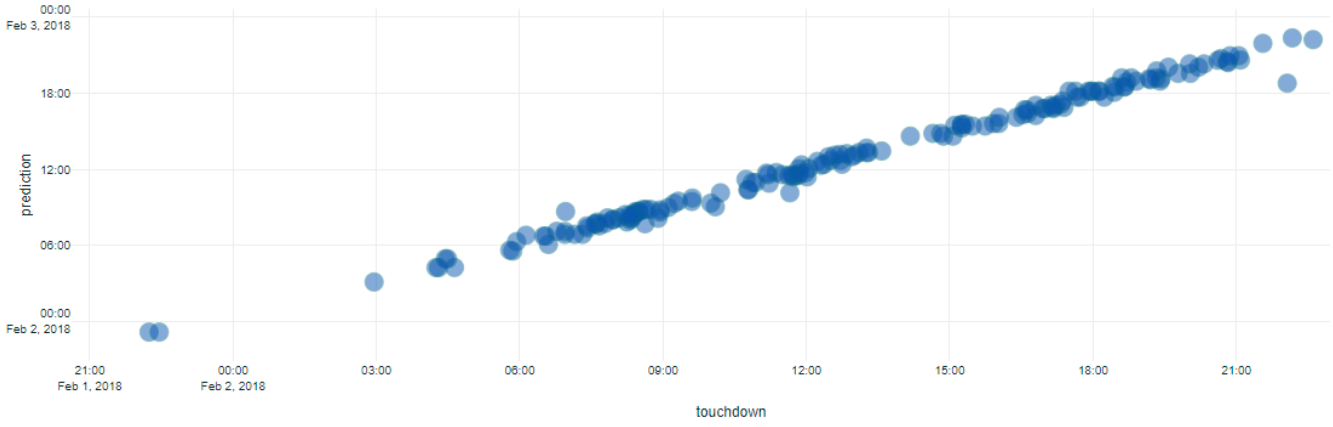



Figura 7.1: Resultados Decission tree, experimento 18.2.

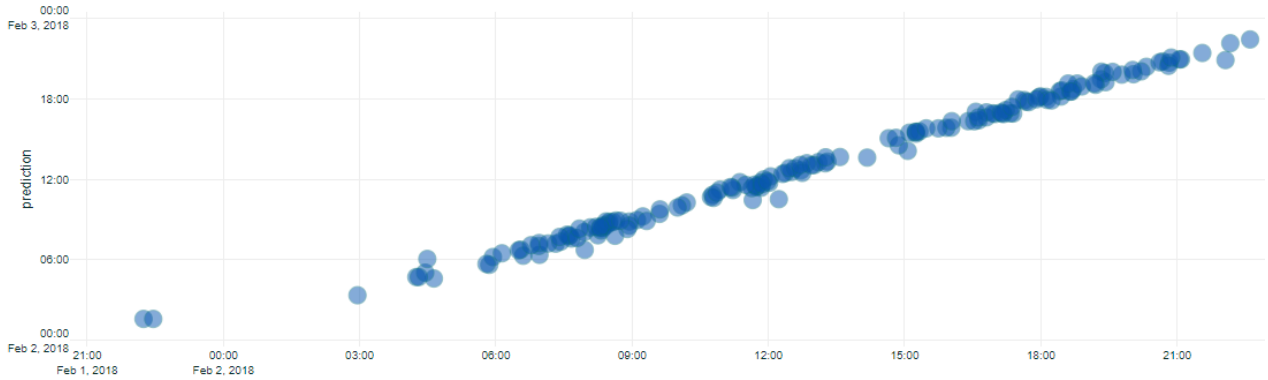



Figura 7.2: Resultados Random forest, experimento 19.3.

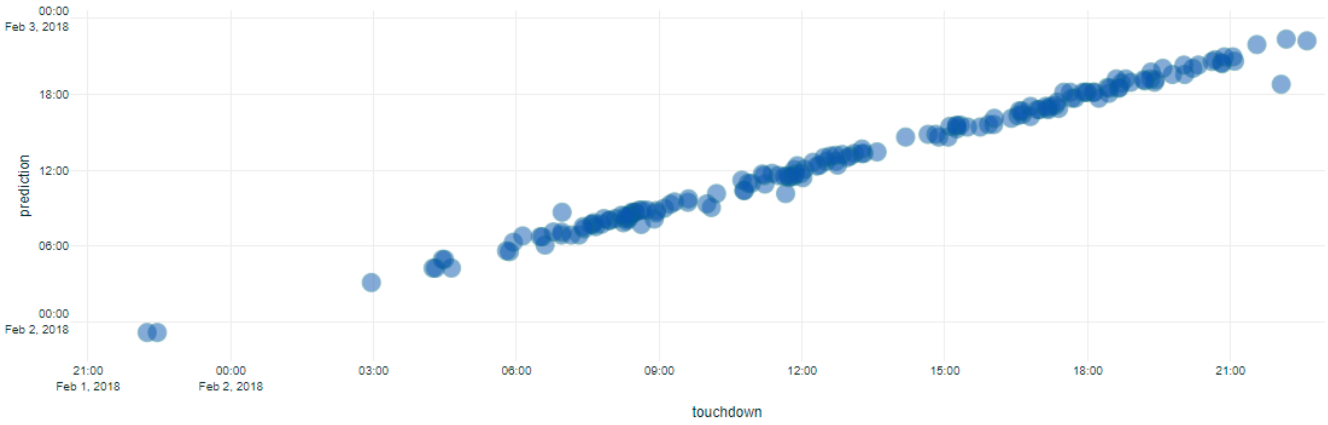



Figura 7.3: Resultados Gradient-boosted Tree, experimento 20.1.

Los resultados obtenidos por los dos modelos anteriores, generados por los algoritmos Decision Tree y Gradient-boosted Tree, han sido de 1594 segundos para la Raíz del error cuadrático medio (RMSE), 2542093 y 2542089 segundos para el Error cuadrático medio (MSE), 1041 segundos para Error absoluto medio (MAE) y 99 % para R al cuadrado (R2). Mientras que para el modelo generado por Radom Forest han sido de 1821 segundos para la Raíz del error cuadrático medio (RMSE), 3319160 segundos para el Error cuadrático medio (MSE), 1045 segundos para Error absoluto medio (MAE) y 99 % para R al cuadrado (R2).

# Capítulo 8

# Conclusiones y trabajo futuro

En este capítulo se expondrán las conclusiones a las que nos ha llevado la realización de nuestro proyecto. Adicionalmente se mencionaran las distintas mejoras que se pueden llevar a cabo, con el fin de que puedan ser realizadas en un futuro.

La realización de este proyecto ha supuesto el aprendizaje de multitud de conocimientos nuevos, ya que se han utilizado tecnologías que no habían sido tratadas durante el Grado. Este hecho también ha generado dificultades, ya que se ha tenido que emplear parte del tiempo de la realización de este proyecto al aprendizaje tanto de las tecnologías, como de la interpretación y entendimiento de los datos con los que se ha trabajado.

El hecho de trabajar con datos relacionados con vuelos nos ha hecho tener la necesidad de estudiar el funcionamiento del tráfico aéreo. Siendo determinantes los protocolos y tecnologías utilizadas para controlar la situación de los aviones, mediante las cuales se han generado los datos con los que posteriormente hemos trabajado en el proyecto. En cuanto a las tecnologías utilizadas el uso de Spark supuso el aprendizaje de este nuevo framework que nos permite el tratamiento de grandes cantidades de datos, además de poseer una librería con la que implementar multitud de algoritmos de Machine Learning.

Respecto al tratamiento de los datos, nos fue útil los conocimientos adquiridos en Sistemas de Bases de Datos, ya que hacíamos uso de algunas de las consultas aprendidas. Una ventaja de Spark era que nos permitía programar en Python, lenguaje que ya había utilizado previamente en la asignatura de Protocolos y que por lo tanto ya me era familiar. Acerca del uso de algoritmos de Machine Learning me fueron de gran ayuda todos los conocimientos adquiridos en la asignatura de Sistemas Inteligentes, en la cual vimos conceptos teóricos sobre estos algoritmos y también hicimos uso de herramientas encargadas de implementarlos y visualizar los resultados. A la hora de desarrollar la aplicación web me han sido de gran utilidad las asignaturas de Tecnologías Web, Plataformas Software Empresariales y Fundamentos de las Tecnologías de la Información.

Por último para la gestión y documentación del proyecto me han sido de gran ayuda los conceptos adquiridos en las asignaturas Plataformas Software Empresariales, Modelado Software,

Proceso de Desarrollo del Software y Gestión de Proyectos Basados en las Tecnologías de la Información. Ya que aprendí las diferentes metodologías con las que llevar a cabo la gestión de un proyecto, junto con la realización de diferentes prácticas en las que se llevaba a cabo la gestión de un proyecto. Por último apuntar que el hecho de trabajar con un equipo sin demasiadas prestaciones nos hizo tener limitaciones en el procesamiento de los algoritmos, tardando más tiempo en ser ejecutados.

## 8.1. Conclusiones

Las conclusiones fundamentales a las que nos ha llevado el desarrollo de este proyecto son fundamentalmente dos, por un lado la creciente introducción de técnicas de Big Data e IA en el mundo de la informática y por otro lado, la necesidad de conocer la hora de los aterrizajes ante el creciente incremento del tráfico aéreo.

Respecto al primer punto comentado en el párrafo anterior, en la actualidad se generan grandes cantidades de datos, a los que en muchas ocasiones se les aplica métodos de inteligencia artificial para darles un sentido determinado. El hecho de haber aprendido a manejar una tecnología como es Spark, que además de procesarlos con cierta rapidez nos permite acceder a una librería que implementa múltiples algoritmos de Machine Learning, me ha parecido muy interesante de cara a un futuro laboral. En este proyecto lo que ha sido más relevante han sido los conocimientos adquiridos en Machine Learning, concretamente en los algoritmos de regresión basados en árboles, que nos han proporcionado modelos de predicción, en nuestro caso concreto en el tiempo de aterrizaje de una serie de vuelos.

En relación al segundo punto comentado en el párrafo inicial, la predicción de los tiempos de llegada de los vuelos puede ser de gran utilidad. Primeramente en un aspecto económico, ya que las compañías aéreas se podrían evitar numerosos gastos en indemnizaciones. También sería de gran utilidad para la gestión de las pistas de aterrizaje de los aeropuertos, que pueden llegar a verse desbordados ante el creciente aumento del tráfico aéreo, tanto en la actualidad como en los próximos años.

Para finalizar, respecto a los resultados obtenidos por los distintos modelos generados, cabe destacar como a medida que hemos ido ejecutando los experimentos nos hemos dado cuenta que las variables o características que realmente son útiles a la hora de predecir han sido el tiempo real de despegue obtenido de los mensajes de ADS-B (take_off) y los tiempos de aterrizaje y despegue presentes en el Plan de Vuelo (tiempo_inicio_leg y tiempo_final_leg). Los tres algoritmos han conseguido resultados muy similares consiguiéndose un Error absoluto medio (MAE) de 17 minutos.

## 8.2. Trabajo futuro

Por último en este apartado se propondrán mejoras que podrán ser implementadas en un futuro, teniendo dos aspectos destacables, por una parte la aplicación y por otra el conjunto de

datos.

La aplicación actualmente consta únicamente de tres algoritmos regresivos basados en árboles. Sería interesante incluir otros algoritmos regresivos, como la Regresión Linear. Además se podrían añadir también algoritmos de clasificación para poder hacer predicciones sobre variables categóricas.

Los datos con los que se trabaja en este proyecto corresponden a las llegadas de aviones a barajas durante un día, tomando como fuente de datos los sistemas ADSB y los planes de vuelo. Estos datos podrían ampliarse tomando los datos de más días y considerando las llegadas de aviones a más aeropuertos. También sería interesante ampliar las fuentes de datos, teniendo en cuenta los datos meteorológicos. Ya que la meteorología influye notablemente en el aplazamiento de los vuelos y podría ayudarnos a elaborar mejores predicciones.

# Apéndice A

# Glosario

A continuación de describirán los diferentes términos utilizados a lo largo de la memoria.

**Capítulo 1 :**

- **Organización de Aviación Civil Internacional (OACI):** Agencia de la Organización de las Naciones Unidas creada en 1944 por el Convenio sobre Aviación Civil Internacional para estudiar los problemas de la aviación civil internacional y promover los reglamentos y normas únicos en la aeronáutica mundial.

- **Big Data:** Conjunto de herramientas utilizadas para gestionar, manipular y analizar grandes volúmenes de datos que no pueden ser gestionados por herramientas informáticas tradicionales como una base de datos.

- **IA:** La inteligencia artificial (IA) es la base a partir de la cual se imitan los procesos de inteligencia humana mediante la creación y la aplicación de algoritmos creados en un entorno dinámico de computación.

**Capítulo 2 :**

- **International Air Transport Association (IATA):** Instrumento para la cooperación entre aerolíneas, promoviendo la seguridad, fiabilidad, confianza y economía en el transporte aéreo en beneficio económico de sus accionistas privados.

- **Control de Tráfico Aéreo (ATC):** Servicio proporcionado por controladores situados en tierra, que guían a las aeronaves en los espacios aéreos controlados y ofrecen información y apoyo a los pilotos en los espacios aéreos no controlados. Su objetivo es proporcionar seguridad, orden y eficiencia al tráfico aéreo.

- **Gestión del espacio aéreo (ASM):** Estructura del espacio aéreo, encargada del diseño de procedimientos de vuelo, optimización y planificación de rutas, diseño y planificación de nuevos aeropuertos, y estudios de capacidad de pista.

- **Servicio de tránsito aéreo (ATS):** Servicio que regula y asiste a los aviones en tiempo real para asegurar la seguridad en sus operaciones. En particular ATS sirve para mantener un cierto orden en el tráfico aéreo, y prevenir colisiones entre aviones, dando avisos que aporten seguridad y eficiencia en los vuelos.
- **Gestión de afluencia del tránsito aéreo (ATFM):** Ofrece los medios para alcanzar la eficiencia y efectividad en la gestión del tránsito aéreo (ATM). Contribuye a la seguridad operacional, la eficiencia, la rentabilidad y la sostenibilidad ambiental de un sistema ATM. También es un importante facilitador de la interoperabilidad global de la industria del transporte aéreo.
- **Notification to Airmen (Notam):** Información relativa al establecimiento, situación o modificación de cualquier instalación, servicio, procedimiento o riesgo aeronáutico cuyo conocimiento oportuno sea indispensable para el personal afectado por las operaciones de vuelo.
- **Weather document(WX):** Documento que proporciona conjuntos de datos meteorológicos en formatos geográficos estandarizados.
- **Single European Sky ATM Research (SESAR):** Proyecto de la comunidad europea de transporte aéreo que se encarga del desarrollo e implantación del futuro sistema común de gestión del tráfico aéreo.
- **Instrument flight rules (IFR):** Conjunto de normas y procedimientos contemplados en el Reglamento de Circulación Aérea que regulan el vuelo de aeronaves con base en el uso de instrumentos para la navegación, lo cual implica que no es necesario tener contacto visual con el terreno.
- **Visual flight rules (VFR):** Conjunto de normas contenidas en el Reglamento de Circulación Aérea que establecen las condiciones suficientes para que el piloto pueda dirigir su aeronave, navegar y mantener la separación de seguridad con cualquier obstáculo con la única ayuda de la observación visual.
- **EUROCONTROL:** Organización Europea para la Seguridad de la Navegación Aérea, su objetivo fundamental es la armonización e integración de los servicios de navegación aérea en Europa para lograr una mayor seguridad y eficiencia en las operaciones de tránsito aéreo.
- **NextGen:** Proyecto cuyo objetivo es la modernización del actual sistema nacional de aviación estadounidense, aumentando la seguridad, la eficiencia, la capacidad y la previsibilidad.
- **Modo S:** Este modo permite que el radar interrogue aeronave a aeronave, de modo que el radar puede manejar un escenario de tráfico mucho mayor sin complicar dicho escenario.

**Capítulo 3 :**

- **Guide to the Project Management Body of Knowledge (PMBOK):** Libro en el que se presentan estándares, pautas y normas para la gestión de proyectos.

**Capítulo 4 :**

- **RDD(Resilient,Distributed,Datasets):** Conjunto de datos distribuidos, los cuales son la abtracción de programación central en todo el modelo que compone Spark.
- **Framework:** Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.
- **MapReduce:** Modelo de programación, utilizado en el proyecto Apache Hadoop, para computación distribuida basado en Java, pero que también se puede desarrollar en otros lenguajes de programación.
- **Apache Hadoop:** Framework de código abierto que permite el procesamiento distribuido de grandes conjuntos de datos en varios clústeres de ordenadores, pero que a ojos del usuario parece un único ordenador.
- **Scala:** Lenguaje de programación orientado a objetos, moderno y multi-paradigma, diseñado para expresar patrones de programación comunes de una forma concisa, elegante, y de tipado seguro.
- **Python:** Lenguaje de programación interpretado cuya sintaxis favorece un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.
- **SQL:** Lenguaje de programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.
- **JSON:** Formato de texto ligero para la representación de objetos JavaScript.
- **CSV:** Tipo de documento que permite la representación de datos en forma de tabla.
- **JDBC:** API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

**Capítulo 6 :**

- **HTML:** lenguaje utilizado para el desarrollo de páginas de web, se caracteriza por el uso de etiquetas.
- **Javascript:** lenguaje de programación orientado a objetos y débilmente tipado, que se utiliza para la creación de paginas web dinámicas.
- **CSS:** lenguaje utilizado para crear el diseño gráfico de paginas web escritas en un lenguaje marcado como es el caso de HTML.
- **HTTP:** protocolo de transferencia de hipertexto que se usa para todo tipo de transacciones a través de Internet. HTTP es una sigla que significa HyperText Transfer Protocol, o Protocolo de Transferencia de Hipertexto.

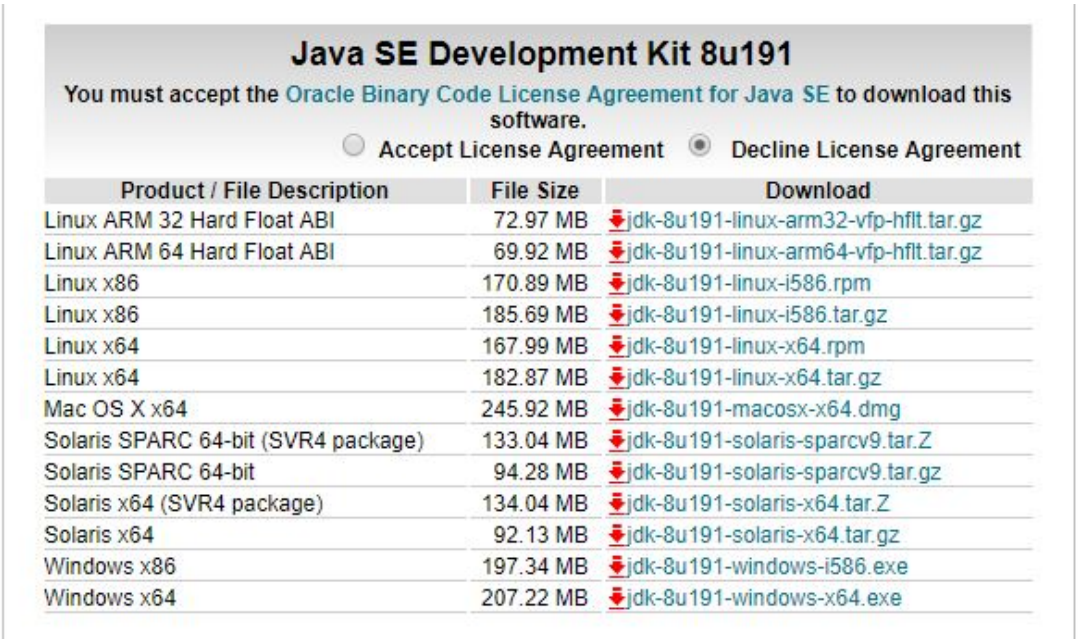
# Apéndice B

# Instalación de Spark en Windows 10

Para proceder a la instalación de Spark será necesario seguir una serie de pasos.

## B.1. Instalación o actualización de Java

Instalamos Java a través de su página oficial https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html.



**Java SE Development Kit 8u191**

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ◉ Decline License Agreement

| Product / File Description | File Size | Download |
|---|---|---|
| Linux ARM 32 Hard Float ABI | 72.97 MB | jdk-8u191-linux-arm32-vfp-hflt.tar.gz |
| Linux ARM 64 Hard Float ABI | 69.92 MB | jdk-8u191-linux-arm64-vfp-hflt.tar.gz |
| Linux x86 | 170.89 MB | jdk-8u191-linux-i586.rpm |
| Linux x86 | 185.69 MB | jdk-8u191-linux-i586.tar.gz |
| Linux x64 | 167.99 MB | jdk-8u191-linux-x64.rpm |
| Linux x64 | 182.87 MB | jdk-8u191-linux-x64.tar.gz |
| Mac OS X x64 | 245.92 MB | jdk-8u191-macosx-x64.dmg |
| Solaris SPARC 64-bit (SVR4 package) | 133.04 MB | jdk-8u191-solaris-sparcv9.tar.Z |
| Solaris SPARC 64-bit | 94.28 MB | jdk-8u191-solaris-sparcv9.tar.gz |
| Solaris x64 (SVR4 package) | 134.04 MB | jdk-8u191-solaris-x64.tar.Z |
| Solaris x64 | 92.13 MB | jdk-8u191-solaris-x64.tar.gz |
| Windows x86 | 197.34 MB | jdk-8u191-windows-i586.exe |
| Windows x64 | 207.22 MB | jdk-8u191-windows-x64.exe |

Figura B.1: Página de descargas de Java.

En la siguiente ventana pinchamos la opción “Development Tools".

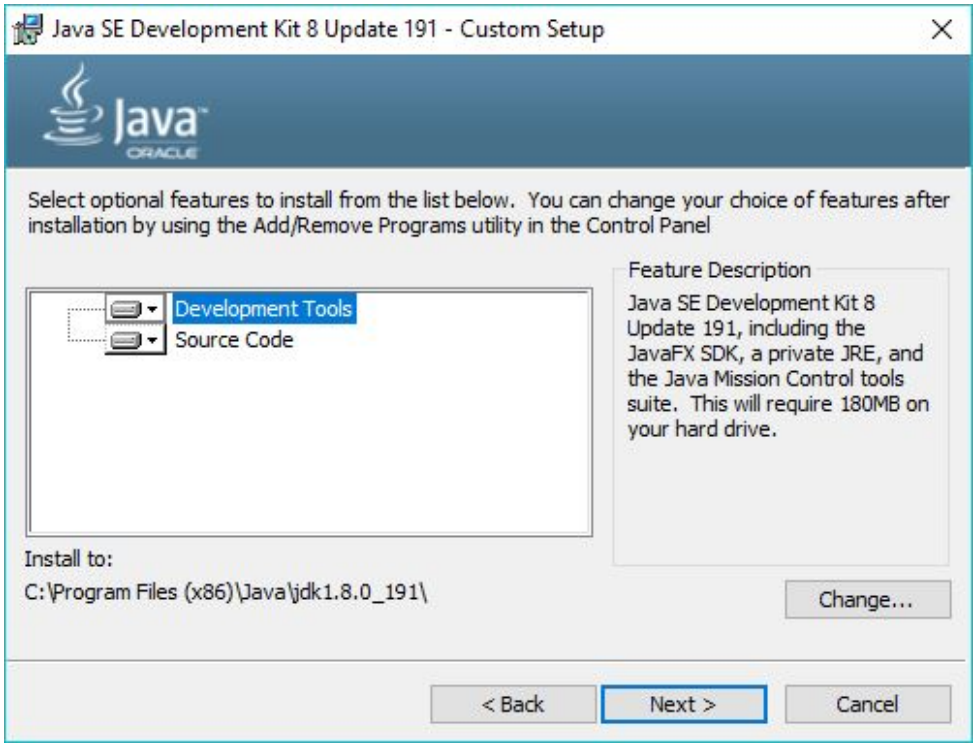



Figura B.2: Opciones de instalación de Java.

En caso de tener Java instalado, habrá que proceder a su actualización.

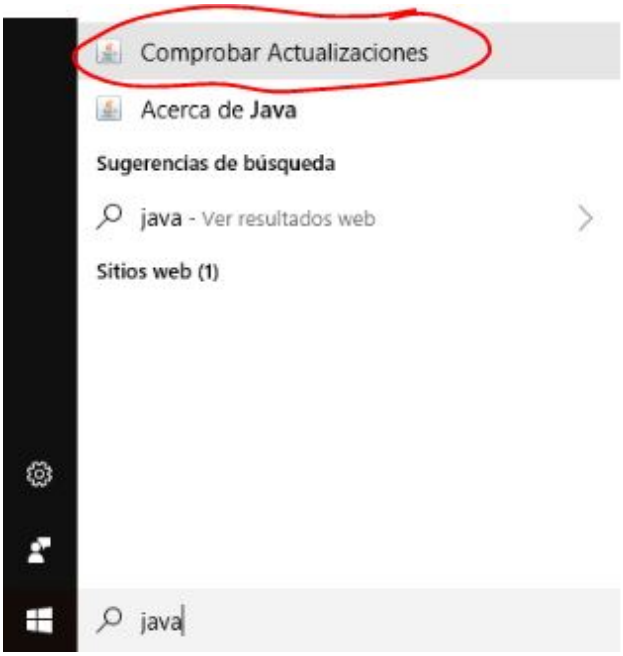



Figura B.3: Búsqueda actualizaciones de Java.

Pinchamos en “Comprobar Actualizaciones”, y nos saldrá el Panel de control de Java donde pincharemos en “Actualizar Ahora”.

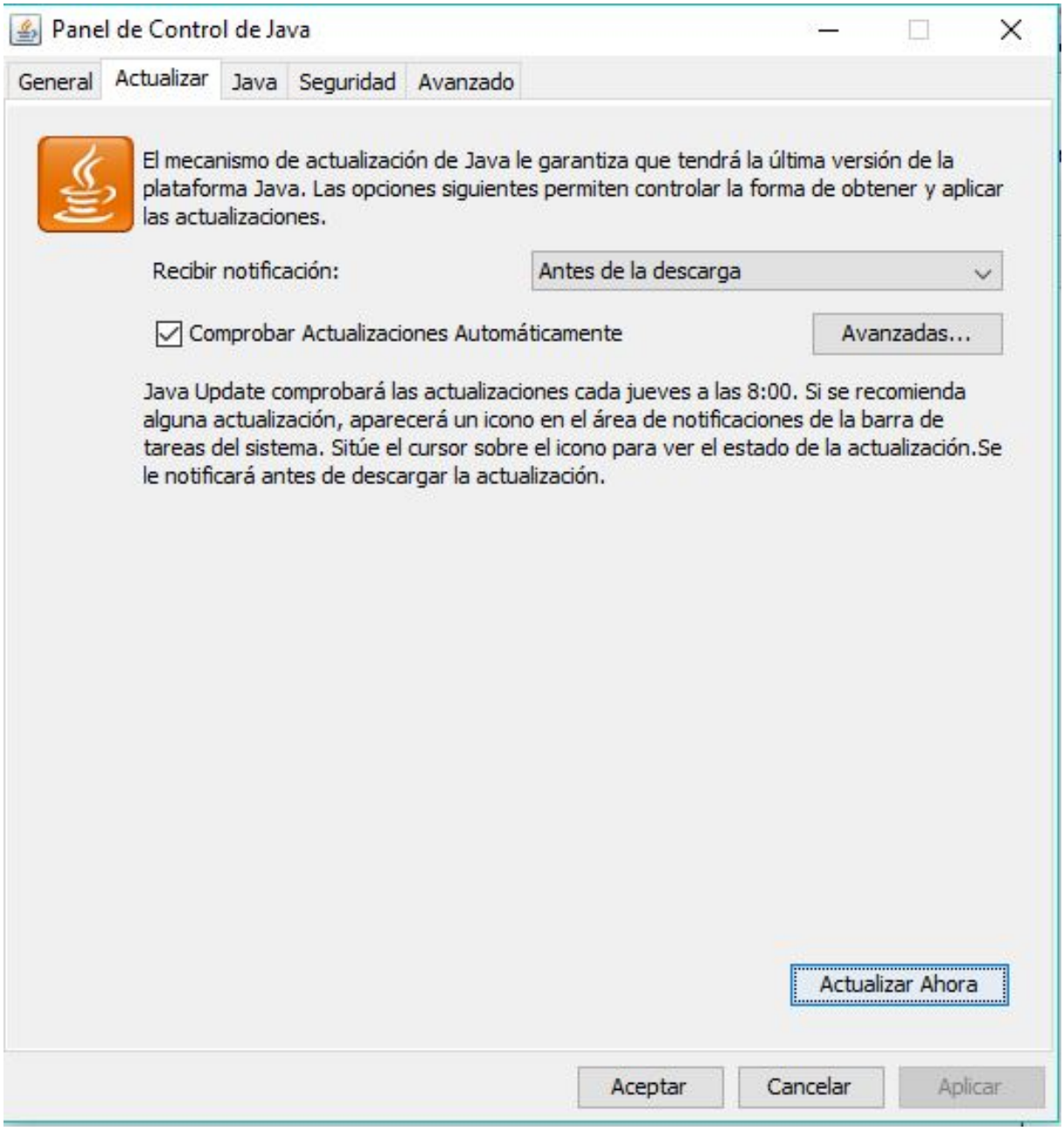



Figura B.4: Panel de control de Java.

## B.2. Descarga de Spark

Descargarnos Spark desde su página oficial http://spark.apache.org/downloads.html. Una vez descargado, creamos una carpeta en el directorio raíz llamada “Spark” y descomprimimos ahí el contenido descargado anteriormente.

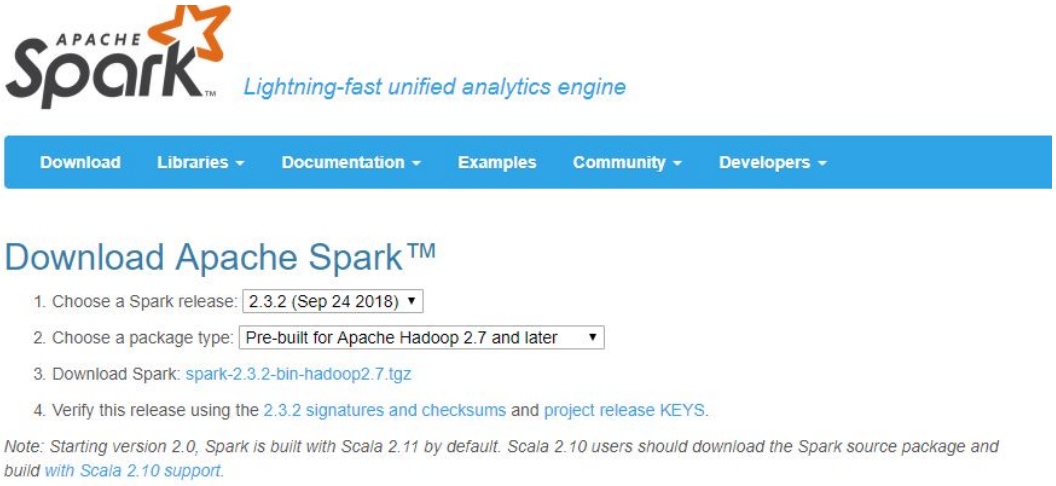



Figura B.5: Pagina de descargas de Spark.

## B.3. Descarga de Winutils

Descargamos winutils del repositorio de Github https://github.com/steveloughran/winutils. Creamos una carpeta llamada “Winutils” en el directorio raíz y descomprimimos ahí el contenido descargado anteriormente.

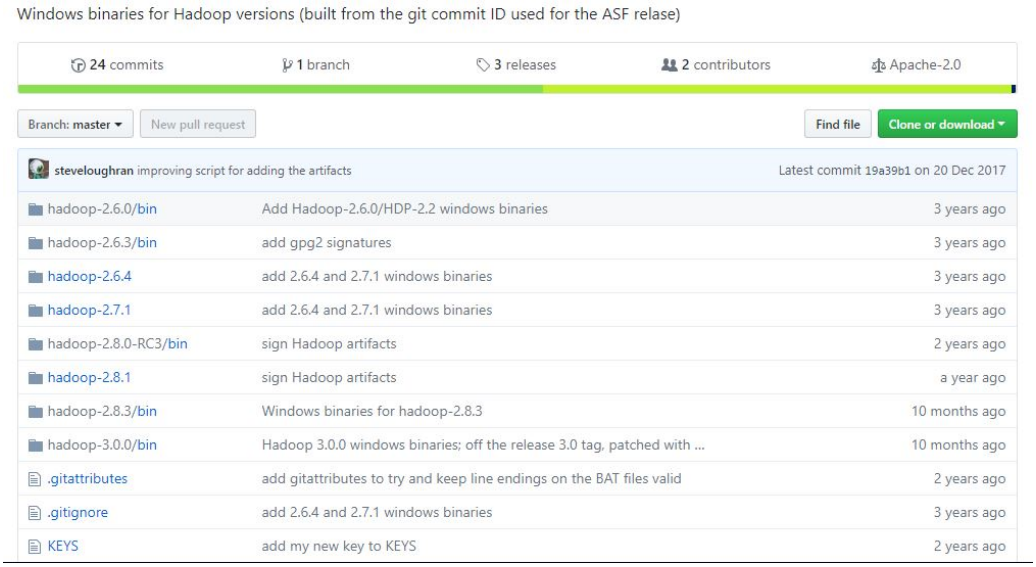



Figura B.6: Repositorio de winutils en Github.

## B.4. Configuración de las variables de entorno.

Por último necesitamos acceder a las variables de entorno del sistema. Escribimos “variables del sistema” en el buscador seleccionamos “Editar las variables de entorno del sistema”, en la siguiente ventana seleccionamos “Variables de entorno...”.

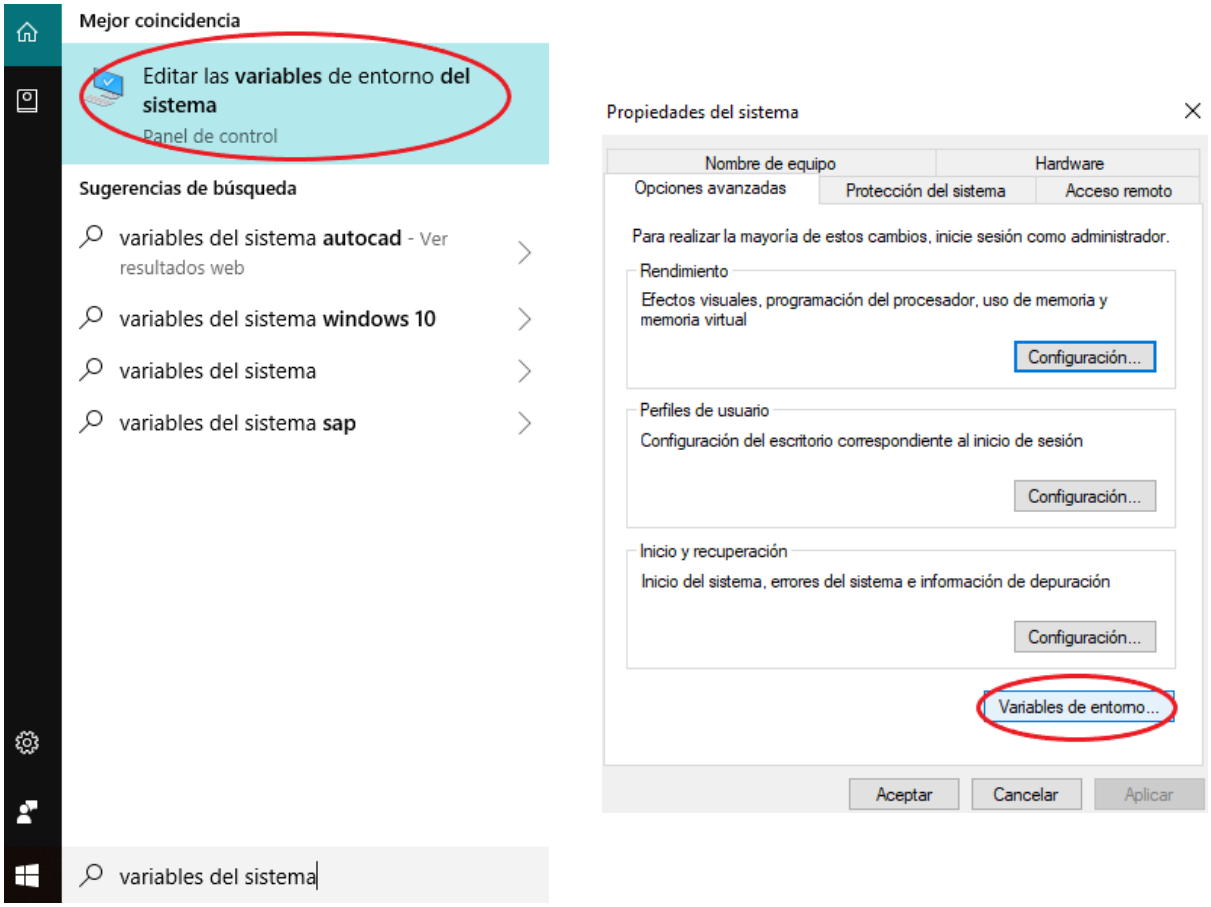



Figura B.7: Acceso a variables de entorno.

Añadimos las variables del sistema recuadradas en rojo seleccionando “Nueva...”.

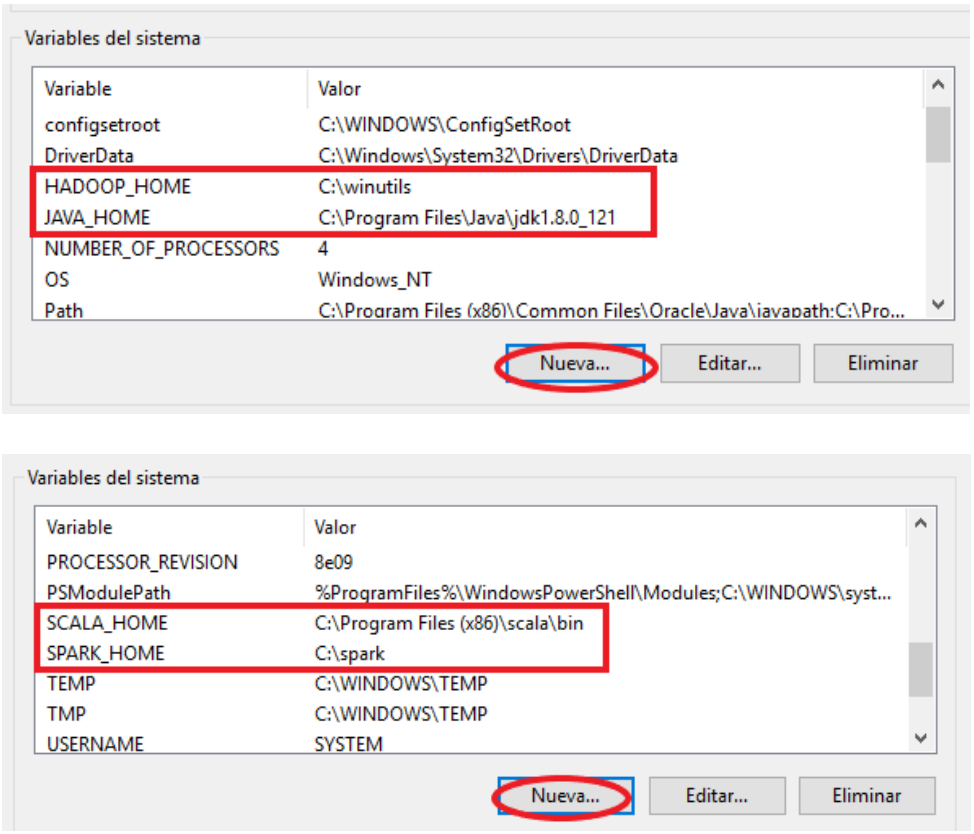



Figura B.8: Variables de entorno añadidas.

Por último, nos falta modificar la variable de entorno Path seleccionando “Editar...”.

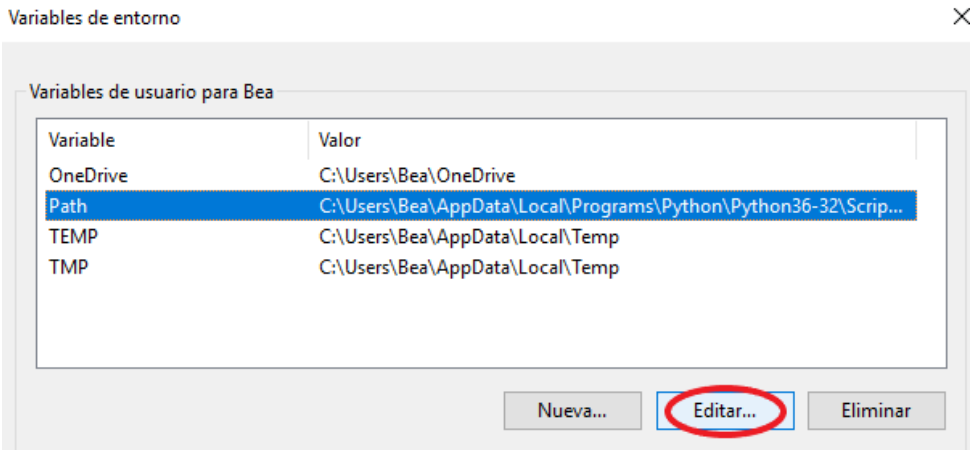



Figura B.9: Editar variable Path.

Añadimos los valores recuadrados en la imagen.

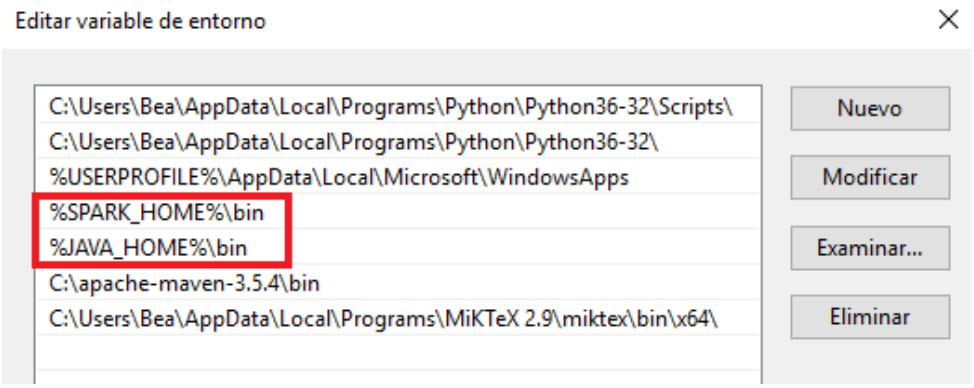



Figura B.10: Líneas añadidas a la variable Path.

Para comprobar que Spark se ha instalado correctamente, abriremos la consola como administrador e introduciremos el comando “spark-shell”.

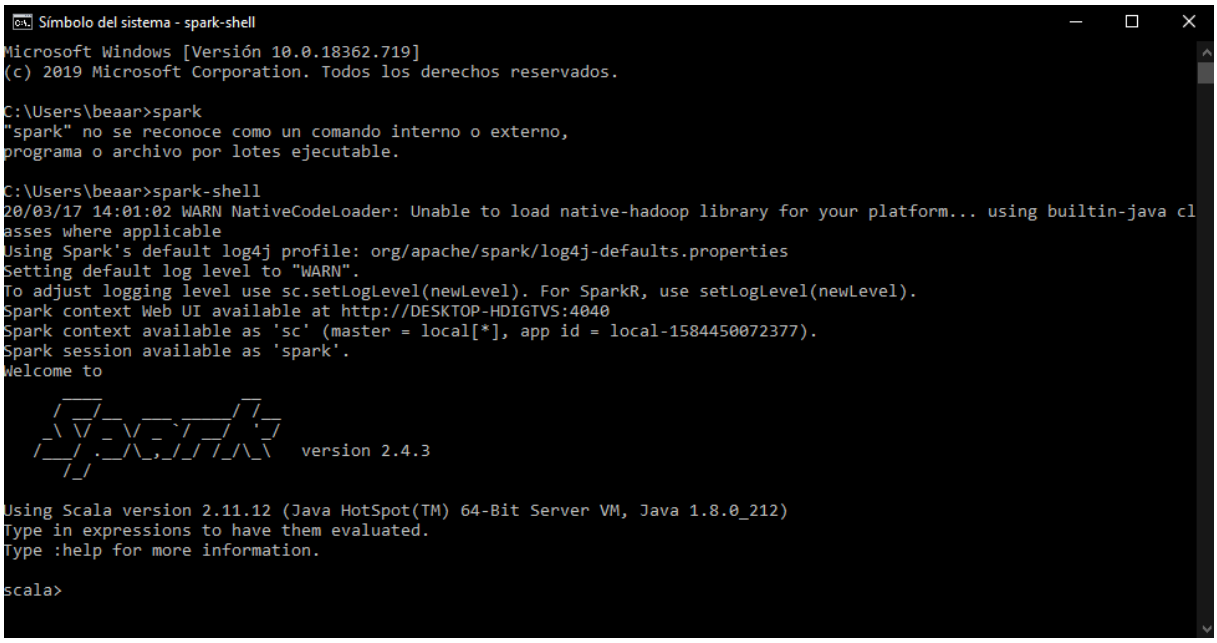



Figura B.11: Ejecución del comando Spark-shell.

# Apéndice C

# Manual de usuario

En este apartado trataremos de explicar el funcionamiento de la aplicación web desarrollada, con el fin de proporcionar los conocimientos necesarios para su correcta utilización. Este manual estará dividido en tres secciones, correspondientes a las tres pestañas de las que se compone nuestra aplicación. En ellas se ira explicando, con ayuda de imágenes, todos los elementos que las componen.

## C.1. Pestaña datos

La primera pestaña que vemos al iniciar la aplicación es la de datos, correspondiente a la Figura C.1.


Figura C.1: Vista general de la pestaña datos.

En la sección de arriba vemos un rectángulo con el texto “Arrastra o selecciona un archivo”. Si clicamos en el link veremos cómo se nos abre el explorador de archivos, correspondiente a la Figura C.2, donde podremos buscar y seleccionar el archivo csv que queramos cargar. También podremos cargar un archivo arrastrándole hasta dentro del rectángulo.

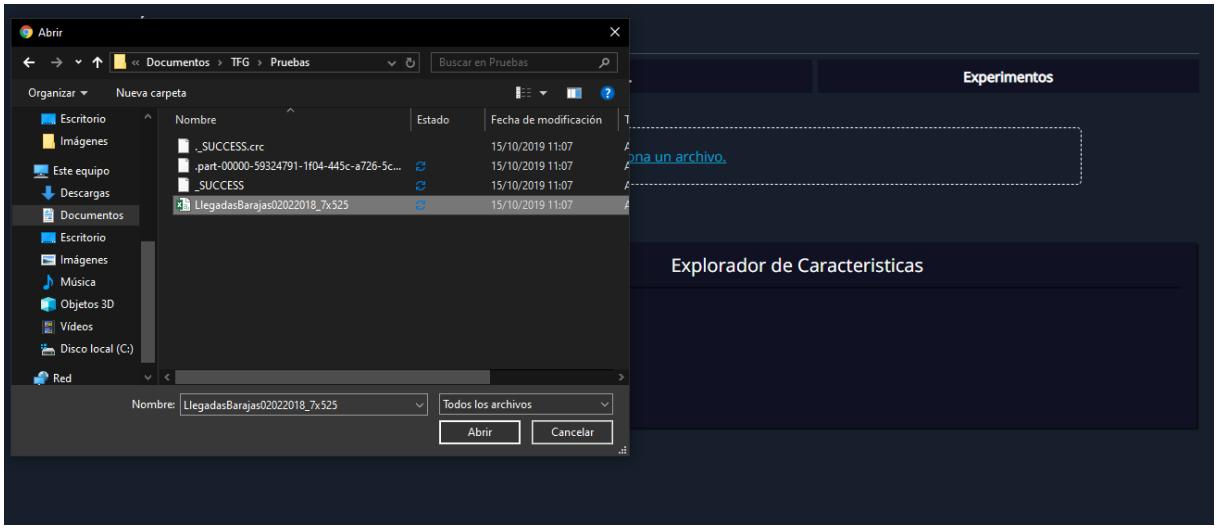



Figura C.2: Vista de la carga de un dataset.

Una vez cargados los datos se mostrara un texto en la parte superior indicando el nombre del archivo, y se rellenaran los paneles de “Características” y “Explorador de características”, como se muestra en la Figura C.3.


Figura C.3: Vista de la pestaña datos al cargar un dataset.

En el panel de características aparecerá un listado con todas las columnas del dataset junto con el botón de visualizar. Por defecto aparecerá seleccionada la primera columna.


Figura C.4: Vista del panel de características.

En el panel de explorador de características se nos permitirá visualizar una tabla con los diferentes elementos que componen la columna y el número de elementos existentes.


Figura C.5: Vista de la tabla del panel explorador de características.

Justo debajo de la tabla tendremos la misma información de la columna seleccionada expuesta en un gráfico.

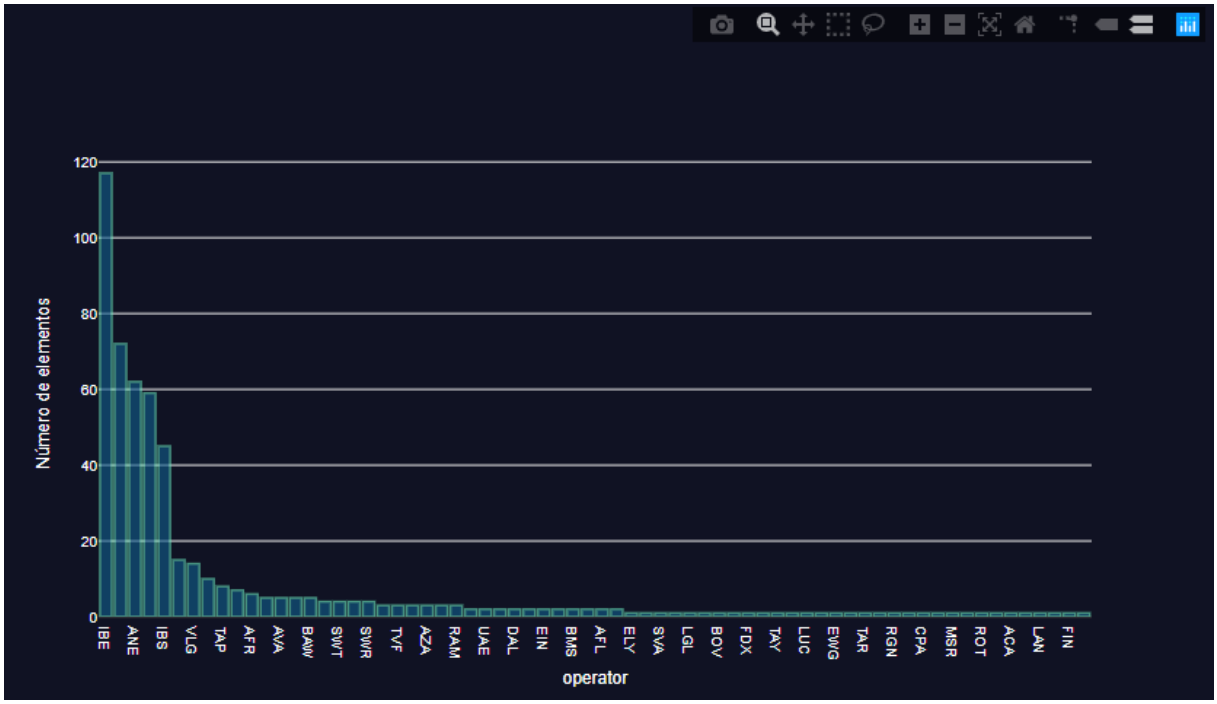



Figura C.6: Vista del gráfico del panel explorador de características.

## C.2. Pestaña ML

La pestaña ML esta compuesta por cinco elementos que explicaremos a continuación. La Figura C.7 muestra la vista de la pestaña cuándo se ha cargado un dataset anteriormente en la pestaña de datos.

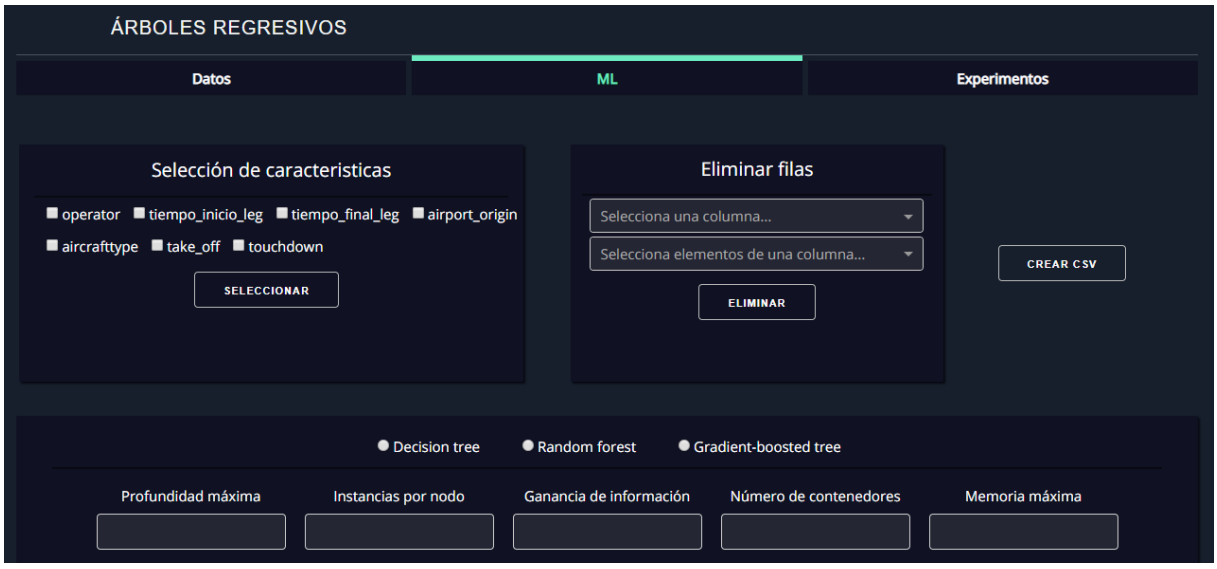



Figura C.7: Vista del gráfico del panel explorador de características.

El panel de selección de características nos muestra todas las columnas de nuestro dataset, entre las que seleccionaremos las columnas que queremos que sean procesadas por los algoritmos.

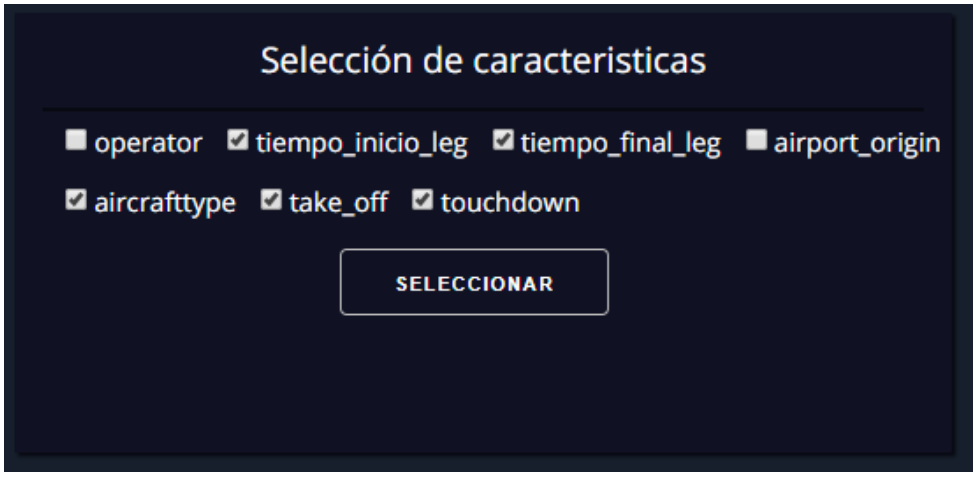



Figura C.8: Vista del panel de selección de características.

El panel de eliminar filas estará compuesto por dos campos, en el primero debemos seleccionar una columna de las elegidas anteriormente en el panel de selección de características y en el segundo deberemos escoger los valores que deseamos borrar de dicha columna. Una vez borrados los datos aparecerá un mensaje informando de los datos que han sido borrados.


Figura C.9: Vista del panel de de eliminar filas.

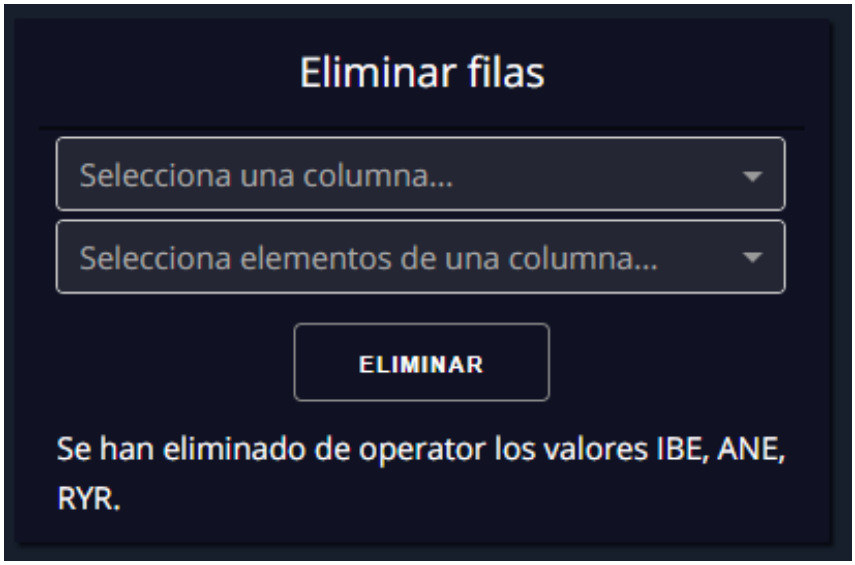



Figura C.10: Vista del panel de de eliminar filas.

El botón CREAR CSV, se encargara de generar un csv con las modificaciones hechas anteriormente por la selección de características y el eliminar filas. Al finalizar la acción aparecerá un link mediante el que descargar el archivo. El nombre del archivo se corresponde con la fecha y hora en que se generó.


Figura C.11: Vista del botón de CREAR CSV.

Seguidamente tenemos un panel con los algoritmos, sus parámetros, el porcentaje de división del dataset y la columna sobre la que se harán las predicciones.

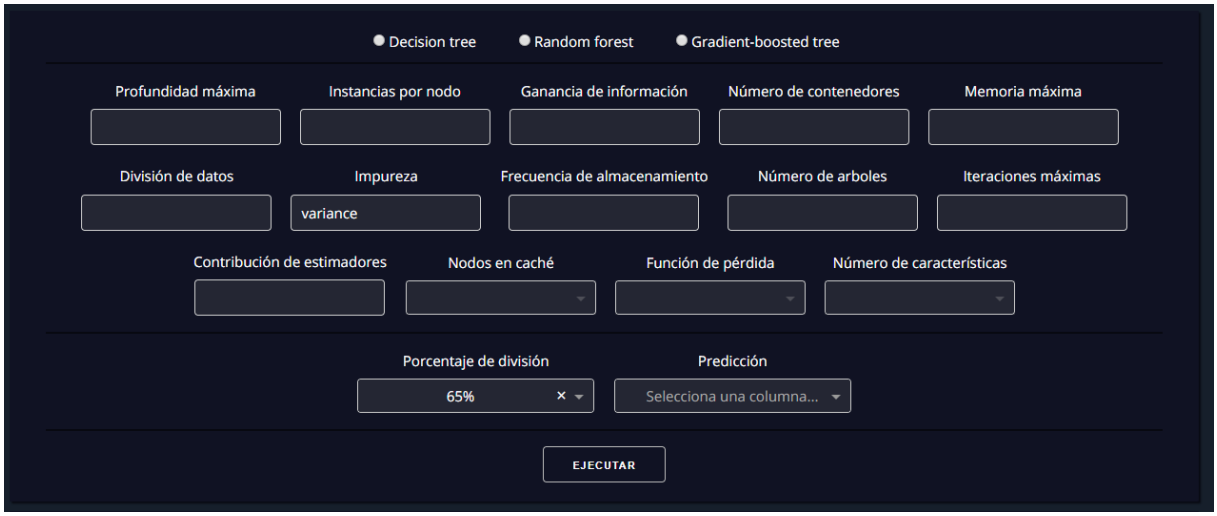



Figura C.12: Vista del panel de configuración de los algoritmos.

Al seleccionar uno de los algoritmos se establecen los valores por defecto de los parámetros pudiéndose modificar. Los parámetros que se encuentran vacíos están deshabilitados para ese algoritmo.

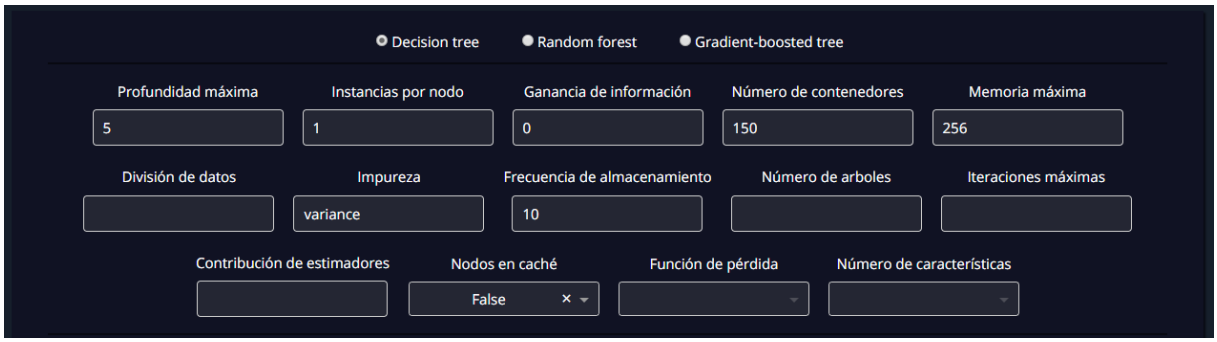



Figura C.13: Vista del panel de configuración de los algoritmos, con los parámetros predefinidos.

Para ayudar a establecer la configuración de los algoritmos se ofrece una breve descripción de cada parámetro poniendo el ratón sobre su nombre, como se muestra en la Figura C.14.

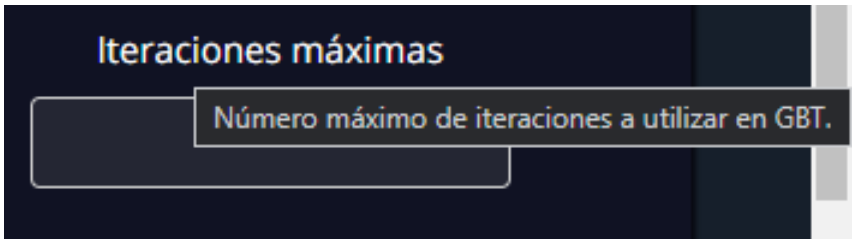



Figura C.14: Vista de la información de adicional de los parámetros.

En la selección de la columna a predecir solo nos mostrará las columnas de tipo numérico, ya que los algoritmos implementados son regresivos.

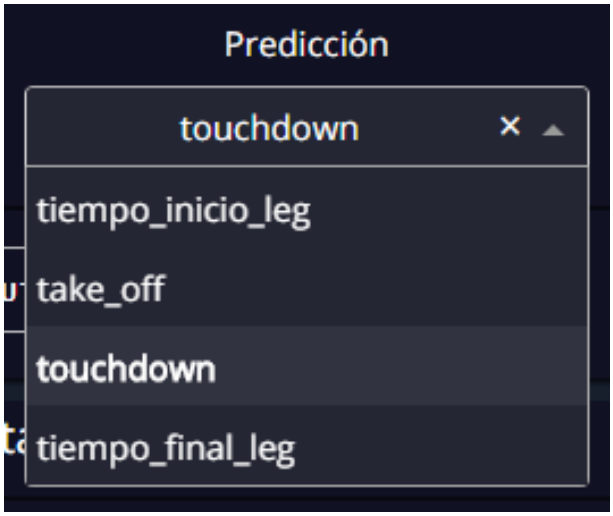



Figura C.15: Vista de la selección de la columna a predecir.

Una vez que han terminado los algoritmos de ejecutarse, en el panel de resultados aparecerán en la parte superior aparecerán los resultados obtenidos por las cuatro métricas, descritas en la sección 4.3.

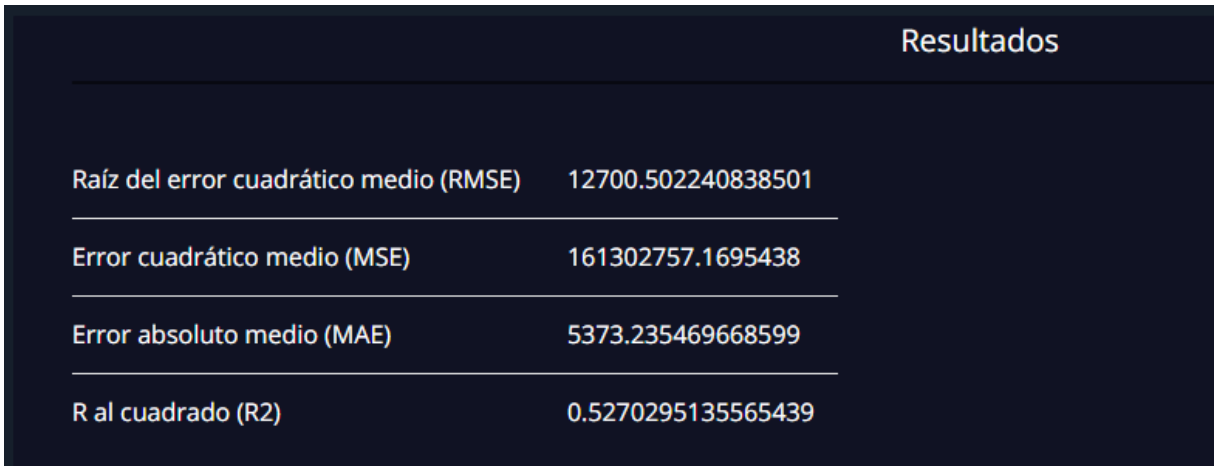



Figura C.16: Vista superior del panel de resultados.

En la parte inferior tendremos un gráfico donde se podrán visualizar los resultados. Se podrán elegir los valores de los ejes X e Y, teniendo como opciones todas las columnas del dataset además de la columna con los valores predichos. De forma predeterminada se visualizará la columna predicha y la columna de predicciones. Además se podrán ver los detalles de cada punto de la gráfica poniendo el ratón encima.

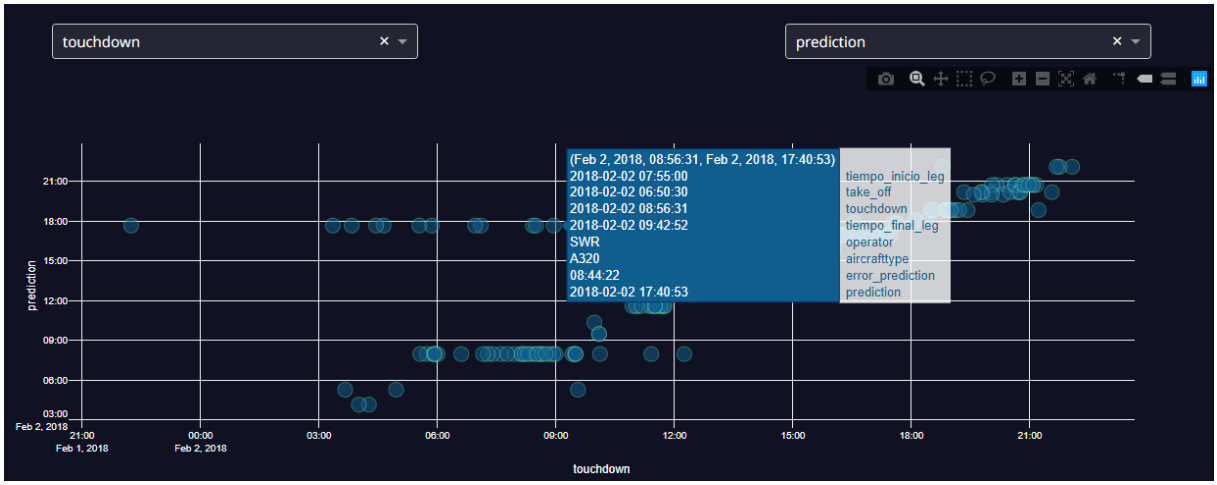



Figura C.17: Vista inferior del panel de resultados.

## C.3. Pestaña Experimentos

Por último tendremos la pestaña de experimentos, la cual nos permitirá añadir varios datasets y algoritmos pudiendo realizar pilas de experimentos. Tendrá tres paneles, uno para los dataset, otro para los algoritmos y por último el de resultados.

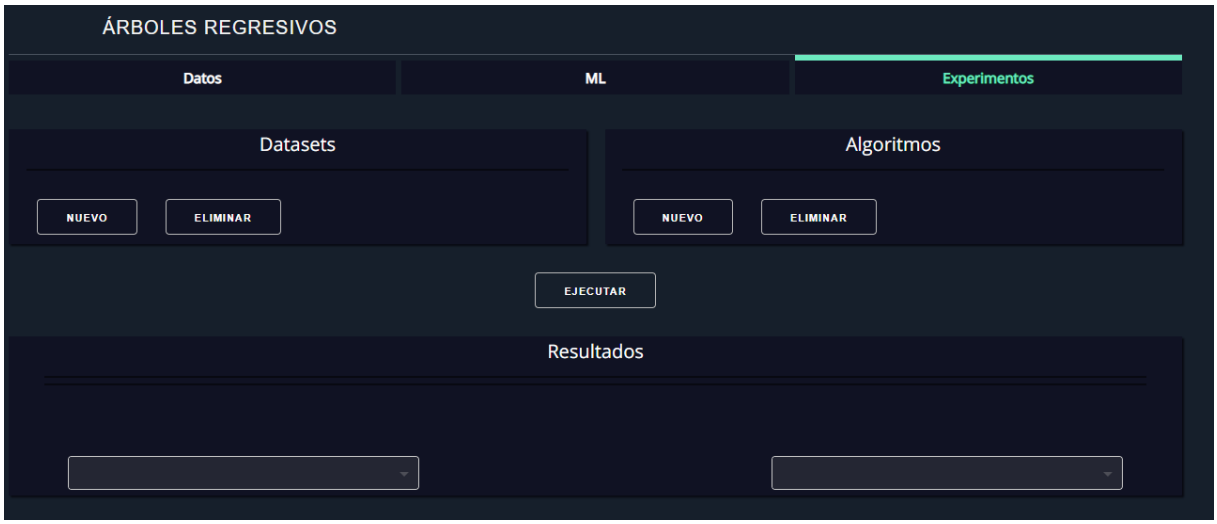



Figura C.18: Vista general de la pestaña de experimentos.

En el panel datasets al clikar en el botón nuevo nos aparecerá una ventana emergente que nos permitirá elegir un dataset, la columna que queremos predecir y el porcentaje de división.


Figura C.19: Vista de la ventana emergente nuevo dataframe.

Al igual que en el panel de datasets en la parte de algoritmos se desplegará una ventana emergente donde configurar el algoritmo a utilizar y sus parámetros. Como hemos explicado en la pestaña "ML", aquí también aparecerá una descripción de los parámetros al pasar el ratón por encima de su nombre.


Figura C.20: Vista de la ventana emergente nuevo algoritmo.

Una vez añadidos los algoritmos y datasets aparecerán en una lista, donde se podrá seleccionar uno para que sea borrado. Cuando tengamos los algoritmos que deseamos se pulsará el botón ejecutar.

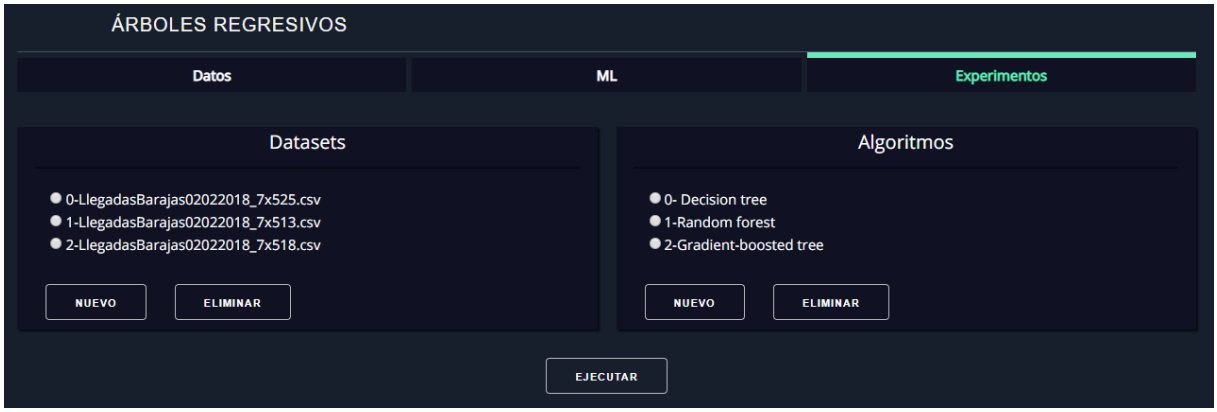



Figura C.21: Vista de la pestaña de experimentos con algoritmos y datasets añadidos.

Posicionando el ratón sobre las listas de los datasets y algoritmos, nos aparecerá información sobre las configuraciones elegidas.

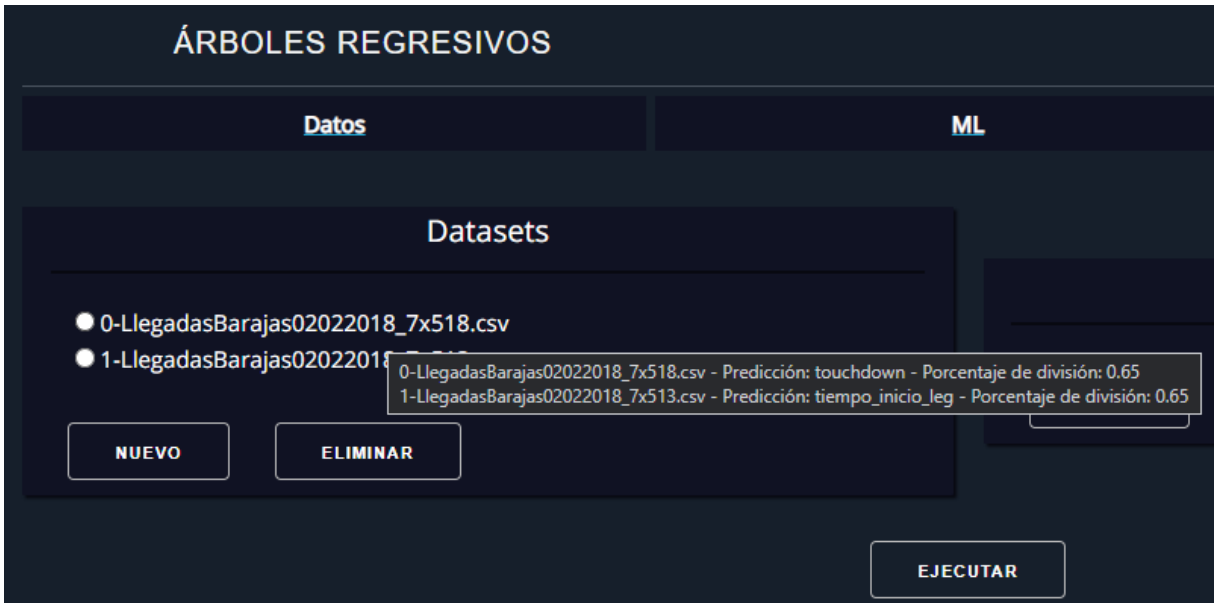



Figura C.22: Información de la configuración del listado de datasets.


Figura C.23: Información de la configuración del listado de algoritmos.

Una vez ejecutados todos los datasets con cada algoritmo configurado, aparecerá una lista con todos los resultados. El nombre de la lista estará compuesto por el dataset, el algoritmo, la variable a predecir y el porcentaje de división. Se mostrará por defecto el primero de la lista. Al igual que en la pestaña anterior se mostrarán los resultados de las métricas de evaluación y un gráfico donde elegir las columnas X e Y. Además a la derecha aparecerá la configuración de los parámetros de cada algoritmo.


Figura C.24: Vista superior del panel de resultados.

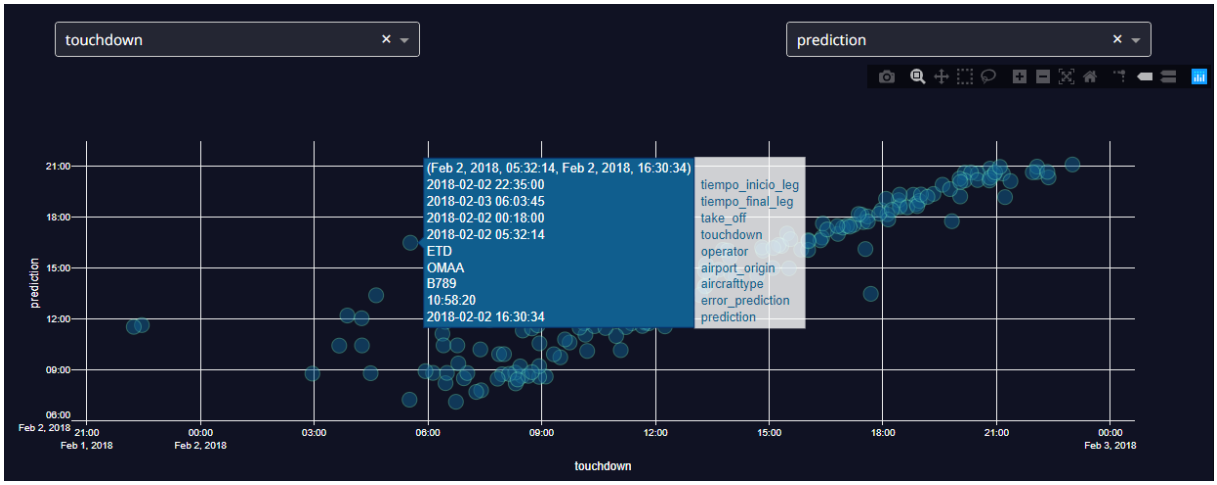



Figura C.25: Vista inferior del panel de resultados.

# Apéndice D

# Contenido del Repositorio

Se subirá a un repositorio información que contendrá los archivos con el código de la aplicación desarrollada, el dataset con los datos de los vuelos, los datasets con los que se han realizado los experimentos y la memoria.
Los archivos subidos al repositorio tendrán la siguiente estructura:

- Dump: Directorio que contiene los archivos leg.csv y message.csv, en los cueles se encuentran los datasets con la información recogida en los planes de vuelos y los sistemas ADS-B.
- Dashboard: Directorio con el código de la aplicación web. Contiene los siguientes archivos y directorios:
  - app.py: Archivo que contiene los métodos desarrollados en Spark, encargados de procesar los datos e implementar los algoritmos.
  - index.py: Archivo encargado de implementar el menú superior de pestañas.
  - apps: Directorio que contiene los archivos: data.py, con la implementación de las pestañas datos y ML, y experiment.py, con la implementación de la pestaña experimentos.
  - assets: Directorio que contiene los archivos 1.base-styles.css, 2.fonts.css y 3.custom-styles.css.
  - Documents: Directorio que contendrá los csvs generados.
- Programa de preprocesado: Directorio que contiene el archivo preprocesado.py, con el programa encargado de procesar los datos iniciales.
- Memoria TFG: Archivo pdf con la memoria.

# Bibliografía


[1] Fundación Centro Tecnológico de la Información y la Comunicación (CTIC). *Inteligencia Artificial y Big Data*. Consultado el 1 de Junio, 2019. url: https://www.fundacionctic.org/es/tecnologias/inteligencia-artificial-y-big-data

[2] Wikipedia. *Aplicaciones de la inteligencia artificial*. Consultado el 1 de Junio, 2019. url: https://es.wikipedia.org/wiki/Aplicaciones_de_la_inteligencia_artificial

[3] preferente.com. *El tráfico aéreo creció un 6,1 % en 2018 con alta ocupación*. 8 Enero, 2019. Consultado el 3 de Junio, 2019. url: https://www.preferente.com/noticias-de-transportes/noticias-de-aerolineas/el-trafico-aereo-crecio-un-61-en-2018-con-alta-ocupacion-283977.html

[4] Europa Press. *La demanda mundial de carga aérea aumentó un 3,5 % en 2018*. 6 Febrero, 2019. Consultado el 5 de Junio, 2019. url: https://www.europapress.es/turismo/transportes/aerolineas/noticia-demanda-mundial-carga-aerea-aumento-35-2018-20190206190354.html

[5] Xataka. *202.157 vuelos registrados en un sólo día: no es ciencia ficción, es el récord del viernes pasado*. 2 Julio, 2018. Consultado el 10 de Junio, 2019. url: https://www.xataka.com/otros/202-157-vuelos-registrados-solo-dia-no-ciencia-ficcion-record-viernes-pasado

[6] Banco Mundial. *Transporte aéreo, carga (millones de toneladas-kilómetros)*. Consultado el 15 de Junio, 2019. url: https://datos.bancomundial.org/indicator/IS.AIR.GOOD.MT.K1?end=2018&name_desc=false&start=1970&view=chart&year=2017

[7] África Semprún, eleconomista.es. *El 41 % de los vuelos se retrasa por el caos en el tráfico aéreo y las huelgas* . 10 Agosto, 2018. Consultado el 16 de Junio, 2019. url: https://www.eleconomista.es/transportes/noticias/9324940/08/18/El-41-de-los-vuelos-se-retrasa-por-el-caos-en-el-trafico-aereo-y-las-huelgas.html

[8] Daniel, Aviación Global. *Cómo funciona el ADS – B. La tecnología que viene ya esta aquí*. 14 Septiembre, 2018. Consultado el 19 de Junio, 2019.

url:http://www.aviacionglobal.com/articulos-tecnicos-de-aviacion/como-funciona-el-ads-b-la-tecnologia-que-viene-ya-esta-aqui/

[9] GENERAL AVIATION NEWS STAFF. *Last ADS-B satellites deployed* . 15 Enero, 2019. Consultado el 19 de Junio, 2019. url:https://generalaviationnews.com/2019/01/15/last-ads-b-satellites-deployed/

[10] OPENSKY NETWORK. *Open Air Traffic Data for Research*. Consultado el 19 de Junio, 2020. url: https://opensky-network.org

[11] MIGUEL A. MARTÍNEZ-PRIETO, ANIBAL BREGON, IVÁN GARCÍA-MIRANDA, PEDRO C. ÁLVAREZ-ESTEBAN, FERNANDO DÍAZ AND DAVID SCARLATTI. "Integrating Flight-related Information into a (Big) Data Lake". En: *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*. 2017, págs. 1-10

[12] ANA M. LÓPEZ, MARIO A. FLORES, JUAN I. SÁNCHEZ. En:*Modelos de series temporales aplicados a la predicción del tráfico aeroportuario español de pasajeros: Un enfoque agregado y desagregado*. Consultado el 26 de Junio, 2019. url: https://www.redalyc.org/html/301/30153163009/

[13] JESÚS MAILLO, ISAAC TRIGUERO, AND FRANCISCO HERRERA. "Un enfoque MapReduce del algoritmo k-vecinos más cercanos para Big Data". En: *Actas de la XVI Conferencia CAEPIA, Albacete Nov 2015*. Consultado el 26 de Junio, 2019. url: https://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/2003_00969.pdf

[14] ANTANAS VERIKAS, EVALDAS VAICIUKYNAS, ADAS GELZINIS, JAMES PARKER AND M. CHARLOTTE OLSSON. *Electromyographic Patterns during Golf Swing: Activation Sequence Profiling and Prediction of Shot Effectiveness*. Abril, 2016.

[15] STATISTA. *Global air traffic - scheduled passengers 2004-2020*. Consultado el 26 de Junio, 2019. url: https://www.statista.com/statistics/564717/airline-industry-passenger-traffic-globally/

[16] ERIC ROSEN, NATIONAL GEOGRAPHIC. *As Billions More Fly, Here's How Aviation Could Evolve*. 20 de junio, 2017. Consultado el 19 Abril, 2019. url: https://www.nationalgeographic.com/environment/urban-expeditions/transportation/air-travel-fuel-emissions-environment/

[17] AENA. *Tráfico de pasajeros,operaciones y carga en los aeropuertos españoles*. Consultado el 21 de Junio, 2019. url: http://www.aena.es/csee/ccurl/792/416/Informe2018_provisionales.pdf

[18] AIRLINES FOR AMERICA. *Last U.S. Passenger Carrier Delay Costs*. Consultado el 25 de Junio, 2019. url: http://airlines.org/dataset/per-minute-cost-of-delays-to-u-s-airlines/

[19] JOHN MULLIGAN, INDEPENDENT.IE. *Europe's flight delays cost €100 a minute*. 21 de Junio, 2019. Consultado el 4 Mayo, 2019. url: https://www.independent.ie/business/world/europes-flight-delays-cost-100-a-minute-37934906.html

[20] CONTROLADORESAÉREOS.ORG. *¿Qué es el Control de Tráfico Aéreo?*. Consultado el 20 de Junio, 2019. url: http://www.controladoresaereos.org/%C2%BFque-es-el-control-de-trafico-aereo/

[21] GREAT BUSTARD'S FLIGHT. *Gestión del tráfico aéreo (ATM) de forma muy simplificada*. 18 de Junio, 2018. Consultado el 23 de Junio, 2019. url: https://greatbustardsflight.blogspot.com/2018/06/gestion-del-trafico-aereo-atm-de-forma.html

[22] WIKIPEDIA. *Control del tráfico aéreo*. Consultado el 5 de Junio, 2019. url: https://es.wikipedia.org/wiki/Control_del_trafico_aereo

[23] SAMET AYHAN, PABLO COSTAS, AND HANAN SAMET. "Predicting Estimated Time of Arrival for Commercial Flights". En: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, págs. 33-42

[24] YAN GLINA, RICHARD JORDAN, AND MARIYA ISHUTKINA. "A Tree-based Ensemble Method for Prediction and Uncertainty Quantification of Aircraft Landing Times". En:*American Meteorological Society–10th Conference on Aritificial Intelligence Applications to Environmental Science*. New Orleans, LA. 2012

[25] CHRISTIAN STROTTMANN KERN, IVO PAIXÃO DE MEDEIROS, AND TAKASHI YONEYAMA. "Data-driven aircraft estimated time of arrival prediction". En:*Proceedings of the 9th IEEE Int'l Systems Conference (SysCon)*. Vancouver, BC. 2015

[26] GABOR TAKACS. "Predicting flight arrival times with a multistage model". En:*IEEE Int'l Conference on Big Data*. Keystone, CO. Octubre, 2014

[27] THEODORE VASILOUDIS. *Block-distributed Gradient Boosted Trees*. 26 Agosto, 2019. Consultado el 1 de Noviembre, 2019. url: http://tvas.me/articles/2019/08/26/Block-Distributed-Gradient-Boosted-Trees.html

[28] WIKIPEDIA. *Validación cruzada*. Consultado el 3 de Noviembre, 2019. url: https://es.wikipedia.org/wiki/Validacion_cruzada

[29] RUBÈN TOUS LIESA, MAURO GÓMEZ PARADA, MARIO MACÍAS LLORET, JORDI TORRES VIÑALS. Consultado el 6 de Octubre, 2019. *Introducción a Apache Spark*. Julio, 2016. url: http://reader.digitalbooks.pro/book/preview/41061/

[30] DIEGO CALVO. *Spark Streaming (procesamiento por lotes y tiempo real)*. 5 Julio, 2018. Consultado el 7 de Octubre, 2019. url: http://www.diegocalvo.es/spark-streaming/

[31] DANIEL GRAÑA, FUTURE BITES. *Apache Spark: Introducción a Spark Sql*. Consultado el 7 de Octubre, 2019. url: https://bites.futurespace.es/2017/04/28/apache-spark-introduccion-a-spark-sql/

[32] DAN LYNN, AGILDATA. *APACHE SPARK CLUSTER MANAGERS: YARN, MESOS, OR STANDALONE?*. 15 Marzo, 2016. Consultado el 15 de Octubre, 2019. url: http://www.agildata.com/apache-spark-cluster-managers-yarn-mesos-or-standalone/

[33] APACHE SPARK. *Machine Learning Library (MLlib) Guide*. Consultado el 2 de Noviembre, 2019. url: https://spark.apache.org/docs/2.4.1/ml-guide.html

[34] APACHE SPARK. *Decision Trees - RDD-based API*. Consultado el 13 de Noviembre, 2019. url: https://spark.apache.org/docs/latest/mllib-decision-tree.html

[35] APACHE SPARK. *Ensembles - RDD-based API*. Consultado el 13 de Noviembre, 2019. url: https://spark.apache.org/docs/latest/mllib-ensembles.html

[36] APACHE SPARK. *pyspark.ml package*. Consultado el 20 de Noviembre, 2019. url: https://spark.apache.org/docs/latest/api/python/pyspark.ml.html?highlight=model#pyspark.ml.regression.DecisionTreeRegressor

[37] MANAGEMENT PLAZA. *CÓMO FUNCIONA SCRUM*. Consultado el 17 de Diciembre, 2019. url: https://managementplaza.es/blog/como-funciona-scrum/

[38] ORGANIZACIÓN DE AVIACIÓN CIVIL INTERNACIONAL (OACI-ICAO). *Procedimientos para los servicios de navegación aérea - Gestión del transito aéreo*. 2007. Consultado el 20 de Junio, 2019. url: https://www.icao.int/SAM/Documents/2010/ASTERIX/07%20%20DOC4444.pdf

[39] PLOTLY DASH. *Dash Layout*. Consultado el 3 de Diciembre, 2020. url: https://dash.plotly.com/layout

[40] PROJECT MANAGEMENT INSTITUTE, PMI ET ALT 2017. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. PMI .6ª edición. url: https://www.pmi.org/pmbok-guide-standards/foundational/pmbok

[41] AGILEMANIFESTO.ORG. *Principios del Manifiesto Ágil*. Consultado el 7 Mayo, 2020. url: https://agilemanifesto.org/iso/es/principles.html

[42] OPENWEBINARS. *Conoce las 3 metodologías ágiles más usadas*. Consultado el 17 de Diciembre, 2019. url: https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/