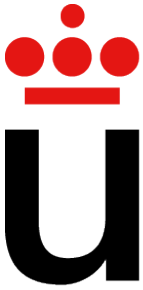


Universidad Rey Juan Carlos

ESCUELA DE INGENIERÍA DE FUENLABRADA

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA TELECOMUNICACIÓN

Trabajo Fin de Grado

APLICACIÓN DE VISUALIZACIÓN Y ANÁLISIS DE DATOS PARA GESTIÓN DE SERVICIOS DE TRANSPORTE

Autor : Miguel Antonio Pereyra Martínez

Tutor : José Felipe Ortega Soto

Curso Académico 2023/2024

Dedicado a

mi madre, mi hermana y mis abuelos

II

Agradecimientos

Tras 6 años de clase, de exámenes, dos grados y por supuesto, de muchas experiencias que han significado un gran crecimiento a nivel personal y profesional, cierro definitivamente esta gran etapa de mi vida.

Quiero agradecer este gran hito personal, una vez más, a mi madre que sin ella nada de esto hubiera sido posible. Este es un logro de ambos, que por supuesto va dedicado a mi hermana que siempre ha velado por nosotros y nunca nos ha abandonado.

Agradecer una vez más a mis compañeros de carrera, que como buen doble grado se merecen doble agradecimiento por compartir con ellos muchos momentos juntos. Paula, Ángela, Ana, Jaime y Polo, gracias por haber sido los mejores compañeros que he podido tener.

A Felipe, tutor de este Trabajo de Fin de Grado, por ayudarme a llevarlo a cabo y por estar dispuesto a ayudarme en todo momento en esta última etapa universitaria.

Y una vez más, a mis compañeros de Iberia por su apoyo durante esta última etapa. Santi, gracias por aguantarme todo este tiempo, por siempre estar dispuesto a ayudarme y ser el gran compañero que eres.

Con estas palabras y este Trabajo de Fin de Grado, pongo punto y final a un doble grado que con sus momentos buenos y malos, siempre recordaré con mucho cariño. Gracias.

Resumen

La Ciencia de Datos está presente en multiples sectores con el objetivo de mejorar los procesos mediante la explotación de los datos. En este Trabajo de Fin de Grado se ha planteado como objetivo la aplicación de las principales funcionalidades que proporciona esta ciencia en un conjunto de datos basado en el registro de vuelos operados desde los 3 principales aeropuertos de Nueva York (Aeropuereto Itl. John F.Kennedy, Aeropuerto de La Guardia y el Aeropuerto Itl. Newark Liberty) durante el año 2013.

El primer hito que se resolverá será la representación e interpretación de los datos en un *dashboard* de visualización mediante la herramienta *PowerBI*. Esto ayudará a sacar conluciones sobre el comportamiento relacionado con las operaciones de los vuelos que se ven influenciados por variables como la estacionalidad, las condiciones meteorológicas o la aerlínea que opera.

Finalmente, una vez se han interpretado los datos, se seleccionarán las principales variables que afectan a los retrasos en las salidas de los vuelos, de forma que se llevará a cabo el desarrollo de un problema de aprendizaje automático basado en aprendizaje supervisado empleando la librería *Scikit-Learn* de *Python*.

Summary

Data science is present in multiple sectors with the aim of improving processes through data exploitation. This Bachelor's Thesis aims to apply the main functionalities provided by this science to a dataset based on flight records operated from the 3 main airports in New York (John F. Kennedy International Airport, LaGuardia Airport, and Newark Liberty International Airport) during the year 2013.

The first milestone to be addressed will be the representation and interpretation of the data in a visualization dashboard using the Power BI tool. This will help draw conclusions about the behavior related to flight operations influenced by variables such as seasonality, weather conditions or the operating airline.

Finally, once the data has been interpreted, the main variables affecting flight departure delays will be selected, and a supervised Machine Learning problem will be developed using the *Scikit-Learn* library in Python.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

El sector del transporte aéreo es uno de los que mayor evolución presenta hoy en día, haciendo frente a constantes desafíos y un crecimiento de la demanda. Por este motivo, surge la necesidad de tomar decisiones de una forma efectiva y por ello la comprensión de los datos se convierte en un factor fundamental para optimizar la eficiencia de determinados procesos en un sector tan complejo como es el aeronáutico. En este Trabajo de Fin de Grado se pretende presentar un proyecto de visualización de datos orientado al servicio del transporte aéreo capaz de proporcionar una visión integral y accesible de la información.

1.1. Objetivo y motivación del proyecto.

El principal objetivo de este proyecto es transformar en inteligencia procesable los datos relacionados con la operación de vuelos de tres principales aeropuertos, de forma que se pueda ofrecer a profesionales del transporte aéreo o autoridades reguladoras una plataforma de visualización de datos intuitiva y eficiente. Se empleará PowerBi para crear una aplicación interactiva en la que se diseñará un *dashboard* en el que se graficarán los datos. Esta herramienta además permitirá la identificación de patrones y tendencias, así como oportunidades del negocio esenciales para la toma de decsiones relacionadas con la operación.

Adicionalmente, se ha empleado Inteligencia Artificial utilizando los datos para llevar a cabo el entrenamiento de modelos de aprendizaje automático supervisado y no supervisado. De esta forma, los modelos entrenados serán capaces de realizar predicciones y reconocimiento de patrones en base a la operación de vuelos durante un año.

A continuación, se van a exponer los principales puntos que han motivado a la realización de este Trabajo de Fin de Grado:

- El impacto de la Ciencia de Datos en el sector del transporte aéreo, desde la gestión del tráfico y planificación de rutas hasta la seguridad operativa y optimización de recursos.
- Con el acceso a la información de forma clara y directa se pretende fomentar la colaboración dentro de la industria, generando un impacto en la calidad del servicio.
- Este proyecto pretende representar el avance tecnológico existente en visualización de datos y como se emplean los mismos para llevar a cabo modelos basados en Inteligencia Artificial.

1.2. Contexto del trabajo

Mi formación académica en la que se complementan la Ingeniería Aeroespacial y la Ingeniería en Telecomunicaciones me ha permitido tener una visión más amplia de ambas áreas, y ello es lo que se pretende demostrar en este proyecto en el que se combinan operaciones de vuelos y Ciencia de Datos.

Actualmente, me encuentro desarrollando mi carrera profesional en una importante empresa del sector aeronáutico dedicada al mantenimiento de motores y aviones. Mi trabajo se enfoca fudamentalmente en la Ciencia de Datos para las distintas direcciones del negocio como *Supply Chain*, *Motores* o *mantenimiento pesado de aviones*.

A continuación, se van a exponer brevemente dos proyectos en los que he colaborado durante mi paso por la dirección de *Supply Chain* y han servido de inspiración para realizar este Trabajo de Fin de Grado. Estos proyectos se centran fundamentalmente en la extracción y tratamiento de datos que finalmente se visualizarán en un *dashboard* realizado con *PowerBI*.

Control de inventario

El objetivo es visualizar de forma directa y diaria el importe total del inventario e demás información relevante. Para ello se realizan extracciones masivas de datos periódicamente que tras un prepocesado de los mismos se muestran en un dashboard de visualización (ver Figura 1.1). Esto permite al equipo de inventario tener una visión global y detallada de la información necesaria para la toma de decisiones.

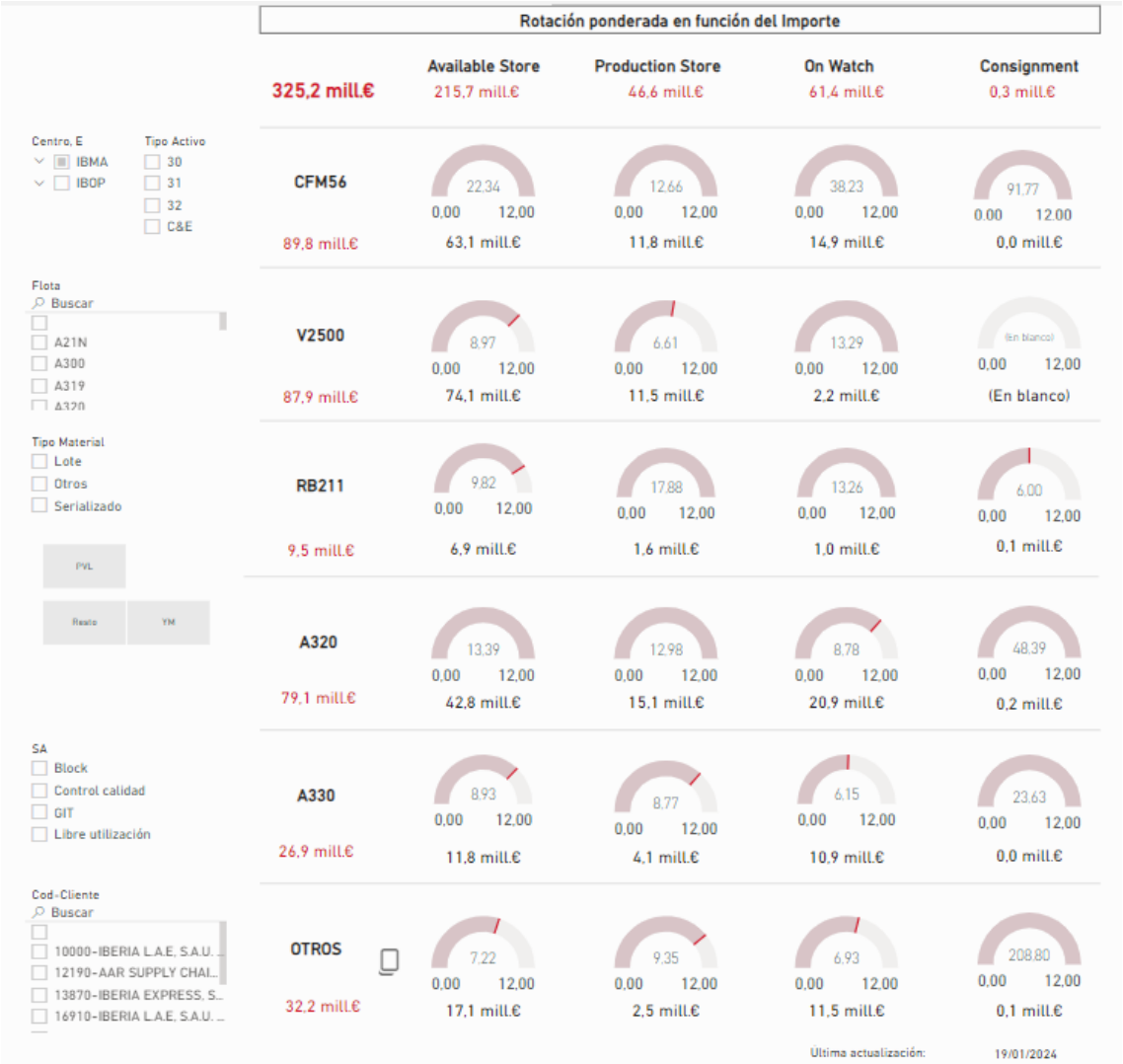


Figura 1.1: *Dashboard* de visualización de datos realizado con *PowerBI* para el control diario del inventario.

Performance de proveedores

A petición de varios equipos de la dirección de *Supply Chain*, surgió la necesidad de analizar el comportamiento de los diferentes proveedores con la empresa trabaja para llevar a cabo sus operaciones. Por ello, se trabajó en extraer todas las órdenes de compra y reparaciones de los diferentes proveedores realizadas durante el último año para posteriormente mostrar en un *dashboard* de visualización de datos si se cumplian con los *Lead Time* prometidos (ver Figura 1.2).

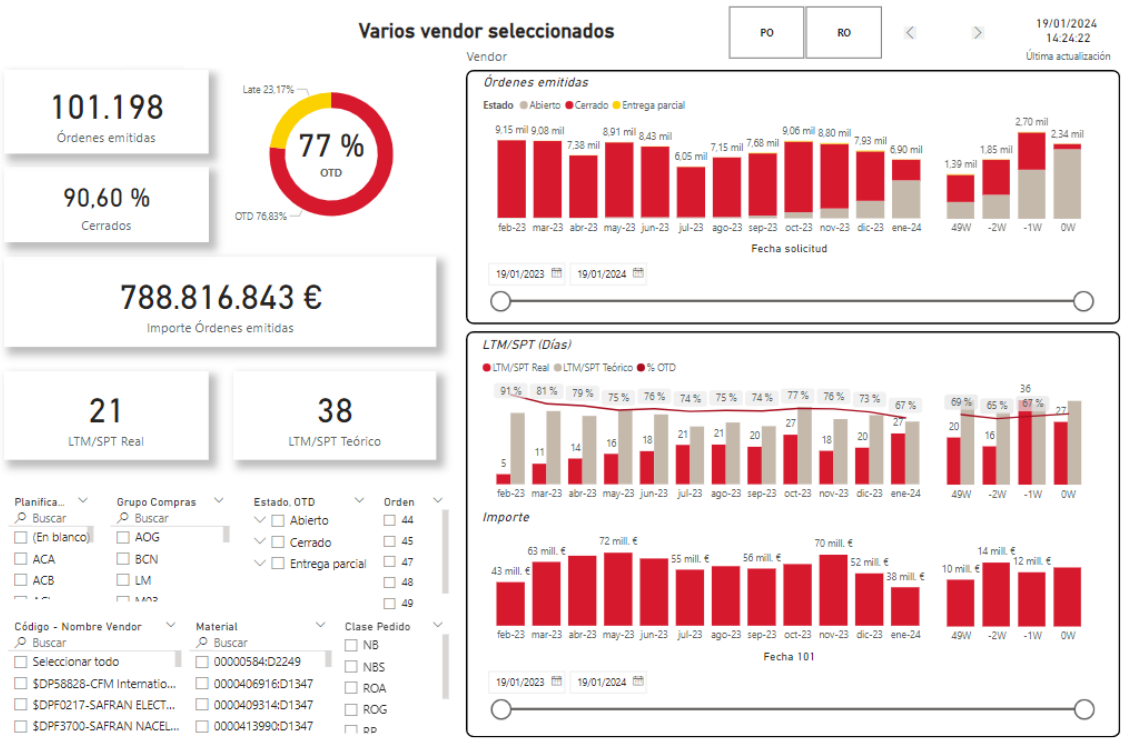


Figura 1.2: *Dashboard* de visualización de datos realizado con *PowerBI* para el análisis de proveedores.

1.3. Objetivos y planificación

En esta sección se plantearán los objetivos de este proyecto asi como las tareas que se realizarán para poder llevarlos a cabo. Además, se expondrá la planificación temporal detallada por cada tarea a lo largo de este curso académico.

1.3.1. Objetivos y tareas

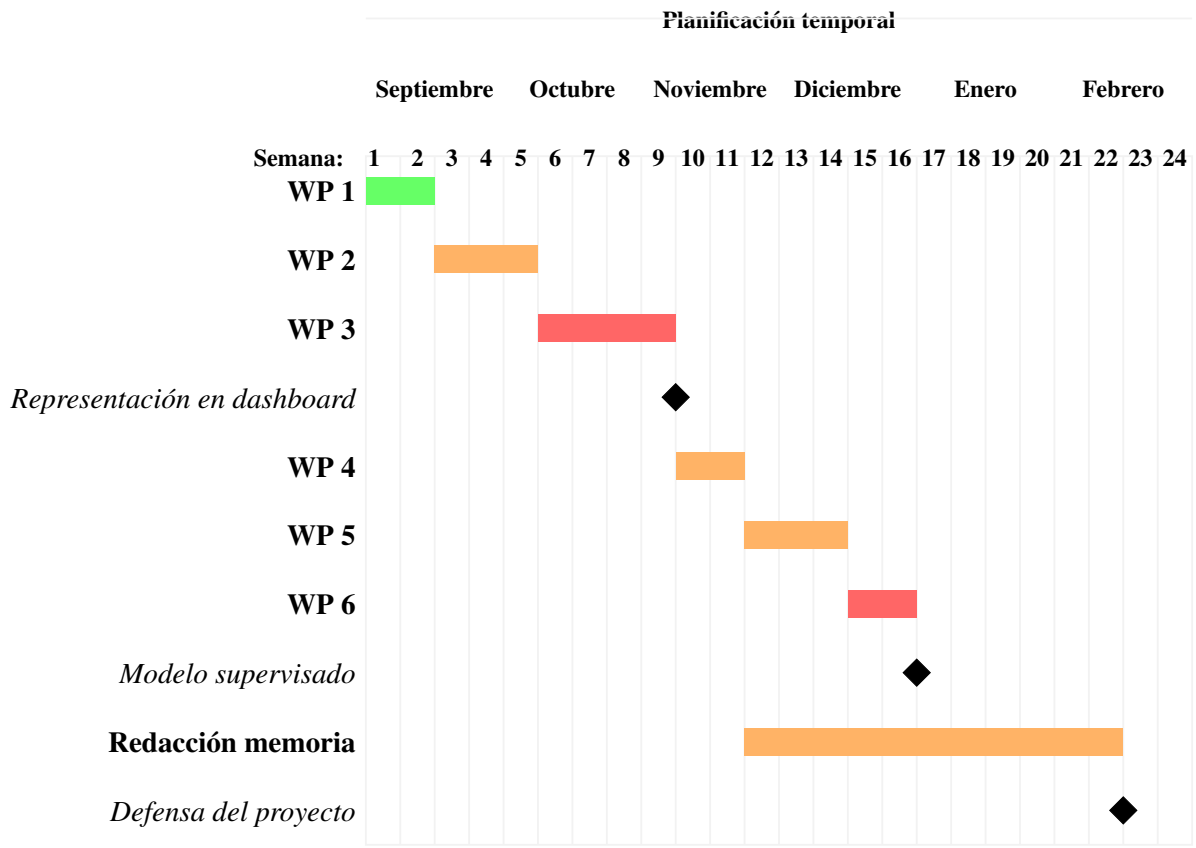
En la Tabla 1.1 exponen los objetivos principales que se deben alcanzar para el desarrollo del trabajo. A continuación, se detalla una lista de objetivos más específicos que deben de alcanzarse para cumplir el objetivo principal que corresponda. Para cada uno de los objetivos específicos se desarrolla una lista de tareas o *work package* (WP) para poder cumplir con dichos objetivos.

Cuadro 1.1: Objetivos y tareas a realizar del proyecto.

Objetivos		**Tareas**
1. Representación de los datos en un dashboard interactivo.	1.1. Interpretación del conjunto de datos.	**WP-1**: Datos obtenidos de [20] relacionado con la operación de vuelos en 3 aeropuertos.
	1.2. Definición de gráficos para representación de variables.	**WP-2**: Definir gráfico de barras, anular, de líneas, etc.
	1.3. Desarrollo de aplicación utilizando PowerBi.	**WP-3**: Representar los gráficos y filtros mediante la interfaz de usuario ofrecida por PowerBi.
2. Implementación de un modelo de Machine Learning basado en aprendizaje supervisado.	2.1. Identificación del problema a resolver y tratamiento de los datos.	**WP-4**: Del conjunto de datos, identificar las variables que pueden ser utilizadas para llevar a cabo un problema de aprendizaje supervisado y cómo realizar un preprocesado de datos.
	2.2. Comprender e implementar el modelo Regresión Logística.	**WP-5**: Llevar a cabo el modelo empleando la librería Scikit-learn de Python.
	2.3. Comprender e implementar el modelo Árbol de decisión.	**WP-6**: Llevar a cabo el modelo empleando la librería Scikit-learn de Python.

1.3.2. Diagrama de Gantt

El siguiente diagrama de Gantt muestra cómo se han distribuido las tareas definidas en la Tabla 1.1 a lo largo del curso académico 2023/2024. Se han marcado con barras el número de semanas que han sido necesarias para cumplir con una tarea concreta, empleando una escala de color en función de la dificultad (Alta , Media , Baja) y por tanto ha exigido un tiempo mayor al esperado. Además se han marcado los hitos cuando las tareas han cumplido con uno de los objetivos principales del proyecto.



1.4. Dominio de competencias

Para llevar a cabo este Trabajo de Fin de Grado ha sido necesario tener conocimientos previos en programación y Ciencia de Datos. Algunas de las asignaturas del plan de estudios del grado que más conocimiento han aportado son:

- Fundamentos de la programación.

- Estadística.
- Ingeniería de Sistemas de la Información.
- Procesamiento Digital de la Informacion.

1.5. Estructura de la memoria

La estructura que va a seguir este Trabajo de Fin de Grado es la siguiente:

- ***Capítulo 1***: se introduce el proyecto, sus principales objetivos y las competencias previas necesarias.
- ***Capítulo 2***: descripción del estado del arte, poniendo en contexto las áreas del conocimiento y las herramientas empleadas.
- ***Capítulo 3***: explicación de la arquitectura de la solución, inlcuyendo esquema general y descripción del diseño, componentes y conexiones.
- ***Capítulo 4***: ilustración de la implementación y experimentos (casos prácticos): se desarrollará detalladamente como se ha abordado el problema.
- ***Capítulo 5***: finalmente, se exponen las conclusiones de este proyecto, las limitaciones que se han encontrado y propuestas de posibles mejoras de cara al futuro.

Capítulo 2

Estado del arte

En este capítulo se recogen las tecnologías utilizadas para llevar a cabo este proyecto, así como los últimos avances y las herramientas que se han empleado. En primer lugar se pondrá en contexto ámbitos del conocimiento como la Ciencia de Datos e Inteligencia Artificial, y finalmente se describirán aquellas herramientas que se han utilizado y el porqué de su elección frente a otras alternativas.

2.1. Ciencia de Datos

Se define la Ciencia de Datos como un campo de estudio que abarca varios ámbitos y se centra en la aplicación de métodos, procesos y sistemas cuyo objetivo es entender y dar conocimiento de los conjuntos de datos en diversas formas. Ciencias como la estadística, matemáticas, informática e ingeniería son esenciales para analizar los datos y obtener información relevante para la toma de decisiones [15].

La Ciencia de Datos emplea diversas técnicas y herramientas para analizar grandes volúmenes de datos, encontrar patrones, realziar predicciones y llevar a cabo problemas complejos. Entre las tareas mas habituales están la limpieza y preparación de los datos, exploración y visualización de datos, modelado predintivo y optimización de decisiones. El proceso más común a seguir es el siguiente:

1. **Identificación y definición del problema**

   Identificar el problema que se pretende resolver mediante análisis de datos.

2. **Recopilación y tratamiento de los datos**

   Recolección de los datos más relevantes de distintas fuentes y prepararlos para su análisis, previamente hay que realizar un proceso de limpieza, eliminar valores atípicos, identificar datos faltantes y realizar las transformaciones necesarias.

3. **Exploración de datos**

   Mediante técnicas estadísticas y visuales para una comprensión de los datos (comprender la estructura, identificar patrones y relaciones entre variables).

4. **Modelado y análisis**

   Desarrollar modelos predictivos mediante *Machine Learning*, minería de datos y análisis estadístico.

5. **Evaluación y validación**

   Evaluar el rendimiento del modelo a partir de métricas adecuadas y validar la capacidad de aprendizaje generalizando nuevos datos.

6. **Despliegue e implementación**

   Implementar soluciones basadas en los hallazgos del análisis de datos y comunicar los resultados de manera efectiva a los interesados.

La Ciencia de Datos tiene multitud de aplicaciones en distintos campos, ya sean los negocios, la medicina, el marketing o las ciencias sociales. La importancia se manifiesta en la capacidad de convertir los datos en información útil necesaria para la toma de acciones y en consecuencia la mejora de procesos.

Los últimos avances en este punto se han visto marcados por el desarrollo de técnicas y tecnologías que permiten gestionar y analizar a partir de grandes volúmenes de datos. Los principales últimos desarrollos pueden resumirse en:

- ***Machine Learning* y *Deep Learning***

  Los avances en el ámbito del aprendizaje automático han permitido realizar análisis mas precisos y complejos.

- **Inteligencia Artificial Explicable (XAI)**

  Debido a la necesidad de comprender y explicar los modelos de *Machine Learning* han derivado al desarrollo de técnicas de XAI, gracias a las cuales es posible interpretar y explicar las decisiones tomadas por los modelos de manera comprensible para los humanos.

- **Automatización y AutoML**

  La automatización de procesos de Ciencia de Datos y la creación de herramientas de AutoML han simplificado la construcción y optimización de modelos, permitiendo a usuarios con menos experiencia desarrollar modelos de aprendizaje automático de manera más eficiente.

2.2. Machine Learning

El campo de la *Inteligencia Artificial* es muy amplio y es posible distinguir varios subconjuntos en función del método de aprendizaje, uno de los más representativos es el *Machine Learning*. En [14] se define el *Machine Learning* como la ciencia que permite a los computadores aprender de un conjunto de datos mediante el empleo de algoritmos estadísticos, de tal forma que sean capaces de extraer unos patrones y características comunes de los datos. Para que esto sea posible se debe realizar previamente un tratamiento de los datos óptimo, labor realizada por el ingeniero de datos.

En función del objetivo del problema que se quiere abordar, es posible clasificar en problema de *Machine Learning* en tres categorías bien diferenciadas:

- **Aprendizaje supervisado** (*Supervised Learning*)

  Este tipo de aprendizaje consiste en el entrenamiento de una muestra o conjunto de datos de entrada que han sido previamente etiquetados. El algoritmo al final del entrenamiento será capaz de poder predecir para una entrada qué etiqueta le correspondería, a esto se le conoce como *clasificación*. Por otro lado, si la entrada de datos es un conjunto numérico y el objetivo es predecir una salida, se trata de una *regresión* ya que para una entrada desconocida por el modelo, éste debe ser capaz de estimar una nueva salida.

- **Aprendizaje no supervisado** (*Unsupervised Learning*)

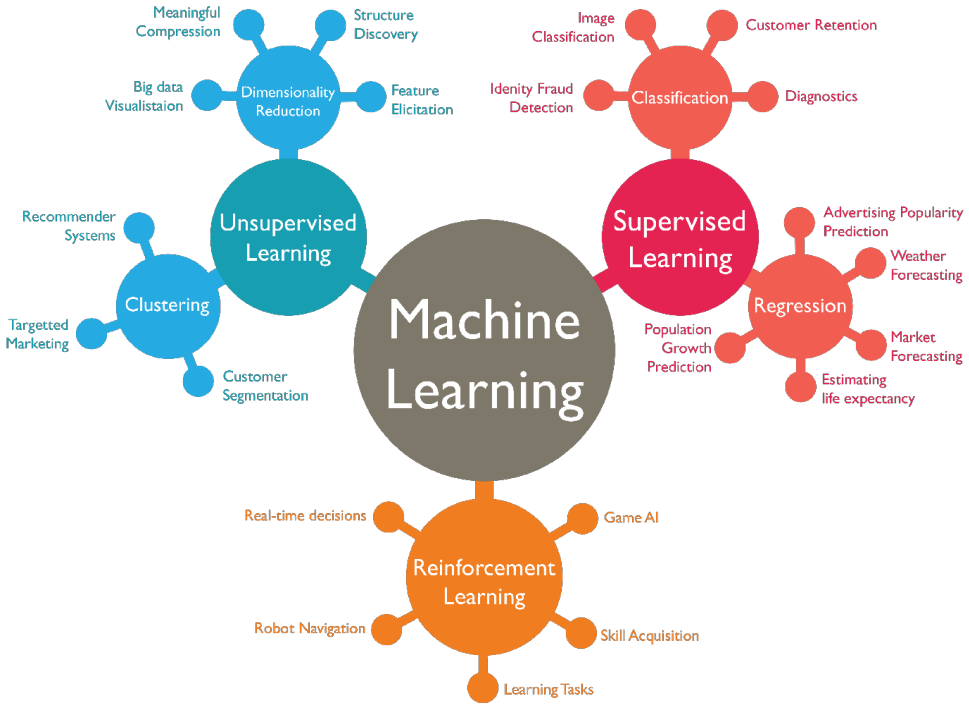


Figura 2.1: Tipos de aprendizaje automático. Imagen de [14].

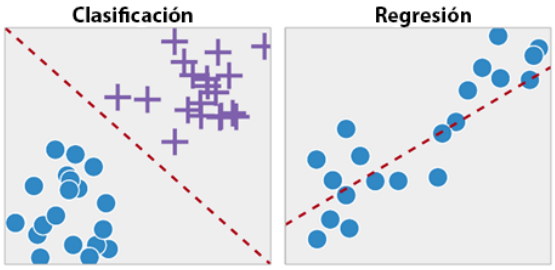


Figura 2.2: Ejemplo de la salida de un aprendizaje supervisado. Imagen de [6]

.

Al contrario que el caso anterior, el aprendizaje no supervisado se caracteriza por la ausencia de un etiquetado previo de los datos. El modelo trata de extraer unas características y patrones comunes de las muestras de entrada permitiendo agruparlos en diferentes clases, a este preceso se le conoce como *clustering*.

- **Aprendizaje por refuerzo** (*Reinforcement Learning*)

  El proceso de aprendizaje se basa en interacciones de el modelo (agente) con un entorno que dará lugar a un cambio de estado y en consecuencia se recibirá una recompensa que determinará las acciones a ejecutar en cada estado. El objetivo es un aprendizaje autónomo de forma que se maximice la recompensa acumulada

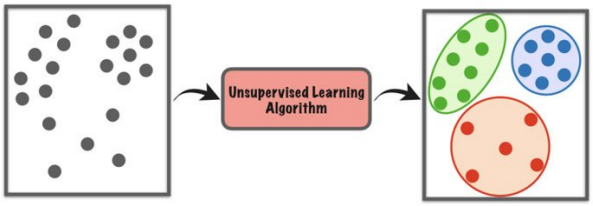


Figura 2.3: Ejemplo de la salida de un aprendizaje no supervisado. Imagen de [7]

.

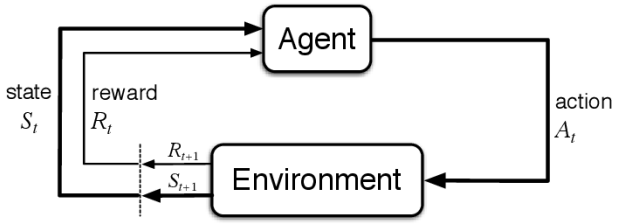


Figura 2.4: Elementos básicos de un modelo de Reinforcement Learning. Imagen de [5].

2.3. Herramientas utilizadas

En esta sección se va a explicar brevemente en qué consisten las herramientas que se han empleado en WP-3, WP-4, WP-5 y WP-6 para el correcto desarrollo e implementación del caso de estudio.

Python

Python es un lenguaje de programación semi-interpretado, es decir, que hace uso de un programa intermedio denominado intérprete pero también aprovecha las ventanjas de los lenguajes compilados. Este lenguaje de programación se caracteriza por la simplicidad y claridad de su sintaxis, el tipado dinámico, el gestor de memoria y la gran cantidad de librerías disponibles que hacen que tenga un mayor nivel de abstracción y por lo tanto que sea posible una programación más sencilla [16].

Figura 2.5: Logo de *Python*. Imagen de [11]

.

Scikit-Learn

Scikit-Learn es una librería en *Python* dedicada al aprendizaje automático que pone a disposición diversos algoritmos supervisados y no supervisadod para llevar a cabo análisis de datos y modelado predictivo.

Figura 2.6: Logo de la librería *Scikit-Learn*. Imagen de [13]

.

Las principales funcionalidades que ofrece esta librería son:

- **Implementación de algoritmos de aprendizaje automático:** proporciona implementaciones de algoritmos de aprendizaje automático (clasificación, regresión, clustering, etc.).
- **API consistente:** presenta una API sencilla, de fácil manejo para el usuario con acceso sencillo a diferentes algoritmos y funcionalidades.
- **Preprocesamiento de datos:** disponibilidad de herramientas para el preprocesado de los datos (escalado de características, manejo de valores faltantes, codificación de variables y división de los datos en subconjuntos de test y entrenamiento).
- **Integración de *NumPy y Pandas*:** integración sencilla con otras librerías populares de *Python* como *NumPy* para cálculos numéricos mediante arrays y *Pandas* para el procesado de datos. Esto permite un tratamiento de datos eficiente y una interoperabilidad fluida con otras herramientas de análisis de datos.
- **Eficiente y escalable:** está diseñada para gestionar tiempo y recursos de forma eficiente, permitiendo el manejo de grandes volúmenes de datos.
- **Documentación y comunidad:** existe una extensa documentación y comunidad de usuarios activa, así como una comunidad de desarrolladores que proporcionan soporte y contribuyen con nuevos algoritmos y funcionalidades.

Jupyter Notebook

Un cuaderno *Jupyter*, comúnmente concocido como *Jupyter Notebook*, es una herramienta basada en navegación web que funciona como un entorno de trabajo en el que es posible hacer uso de un lenguaje de marcado (*markdown*) y lenguaje de programación. Se trata de una herramienta muy útil ya que permite ejecutar código de programación por bloques y además representa una buena opción para la visualización de los resultados [19].

Figura 2.7: Logo de *Jupyter*. Imagen de [9]

.

PowerBI

PowerBI es una herramienta de inteligencia empresarial (BI) desarrollada por Microsoft que proporciona múltiples funcionalidades dedicadas a la visualización, análisis y compartición de datos.

Figura 2.8: Logo de *PowerBI Desktop*. Imagen de [12]

.

A continuación, se van a explicar brevemente en qué consisten estas funcionalidades según [18]:

- ***Conexión y transformación de datos:*** *PowerBI* permite conectarse a diversas fuentes de datos, ya sean bases de datos, archivos excel o servicios en la nube, entre otras muchas alternativas. Además, ofrece herramientas para la transformación de datos como *PowerQuery*, que permite limpiar, dar forma y combinar datos de diversas fuentes.

- ***Modelado de datos:*** El usuario es capaz de crear modelos de datos mediante la definición de relaciones entre tablas, creación de medidad y columnas calculadas y establecer jerarquías.

  Es posible la creación de modelos de datos complejos para representar la estructura subyacente de la información.

- ***Creación de informes interactivos:*** Creación de informes visuales interactivos de forma sencilla para el usuario ya que el mecanismo consiste en arrastrar y soltar los objetos visuales (gráficos, tablas, mapas, etc.) y los datos que son de interés. La herramienta ofrece una amplia variedad de visualizaciones personalizables para la representación de los datos (ver Tabla 2.1).

- ***Tableros de control (Dashboards):*** Es posible la creación de tableros de control que consolidan visualizaciones clave y permiten una vista rápida del estado general de los datos. Estos *dashboards* pueden contener elementos interactivos que permiten explorar los datos en tiempo real.

- ***Análisis avanzado:*** Capacidad de realizar un análisis de datos avanzado mediante la creción de medidas con DAX (*Data Analisys Expressions*) y la posibilidad de utilizar funciones analíticas. Permite el análisis de tendencias, comparativas e identificación de patrones a partir de *Quick Insights*.

- ***Colaboración y compartición:***

  Facilidad de colaboración ya que es posible compartir informes y tableros con otros usuarios. También se integra con otras herramientas de Microsoft, como *Sharepoint* y *Teams* para mejorar la colaboración en el entorno empresarial.

- ***Actualización en tiempo real:*** Ofrece capacidades de actualización en tiempo real para aquellos datos que son más dinámicos. Se integra con flujos de datos en tiempo real como *Azure Stream Analitycs*.

- ***Integración con Servicios en la nube***: *PowerBI* se integra con servicios en la nube de Microsoft, como Azure, para proporcionar almacenamiento y procesamiento escalables.

Objetos visuales y filtros		Definición
1. Objetos visuales: elemento gráfico o interactivo que se utiliza para representar datos en un informe o tablero de control.	Gráfico de barras	Representa los datos mediante barras horizontales o verticales cuyas longitudes son proporcionales a los valores que representan
	Gráfico de líneas	Muestra tendencias a lo largo del tiempo o de una dimensión específica a partir de líneas que conectan puntos de datos.
	Gráfico de sectores	Representa proporciones sobre el total mediante sectores circulares.
	Tablas	Muestra los datos en formato tabular (filas y columnas) con la posibilidad de mostrar los totales.
	Matriz	Similar a una tabla, pero es capaz de agrupar y resumir los datos en filas y columnas dando una visión más jerárquica.
	Tarjeta de visualización	Muestra un único valor o medida como un total o promedio.
	Mapa	Visualización de datos geoespaciales en un mapa interactivo, permitiendo analizar patrones geográficos.
	Histograma	Representa una distribución de datos en intervalos (*bins*) para analizar la frecuencia de ocurrencia.
2. Filtros: herramientas que permiten al usuario controlar la visualización de datos de un informe o tablero de control mediante la selección de valores específicos o aplicando criterios de filtrado.	De página	Aplican filtros a nivel de página, afecta a todos los objetos visuales que se encuentren en la misma página.
	De informe	Aplican filtros a nivel de informe, afecta a todos los objetos visuales que se encuentren en todas las páginas del mismo informe.
	De objeto visual	Aplican a un objeto visual en específico y únicamente afectan a dicho objeto.
	De top N	Seleccionan las N principales o inferiores en función de una medida específica.

Cuadro 2.1: Objetos visuales y filtros utilizados en el proyecto.

Capítulo 3

Arquitectura de la solución

En este capítulo se explicarán los datos que se toman de entrada y el esquema general de la solución adoptada para llevar a cabo los objetivos que se plantean en la Tabla 1.1. Esta sección se estructurará de la siguiente forma:

- Conjunto de datos.
- Representación de los datos.
- Aplicación de modelos de *Machine Learning*.

3.1. Conjunto de datos

En este apartado se va a explicar el origen y en qué consisten los datos que se van a tomar como entrada del *dashboard* de visualización y de los modelos de *Machine Learning*. Como fuente principal de datos se ha escogido un *dataset* relativo a la operación diaria de vuelos, además se han empleado *datasets* adicionales relativos a la información relevante de aeropuertos y aerolíneas.

3.1.1. Contexto de los datos

Ciudades de origen y destino

Los vuelos que se van a tratar en el conjunto de datos han sido operados en Estados Unidos durante el año 2013. Se han considerado los aeropuertos de origen ubicados en la ciudad de

Nueva York (ver Figura 3.1).

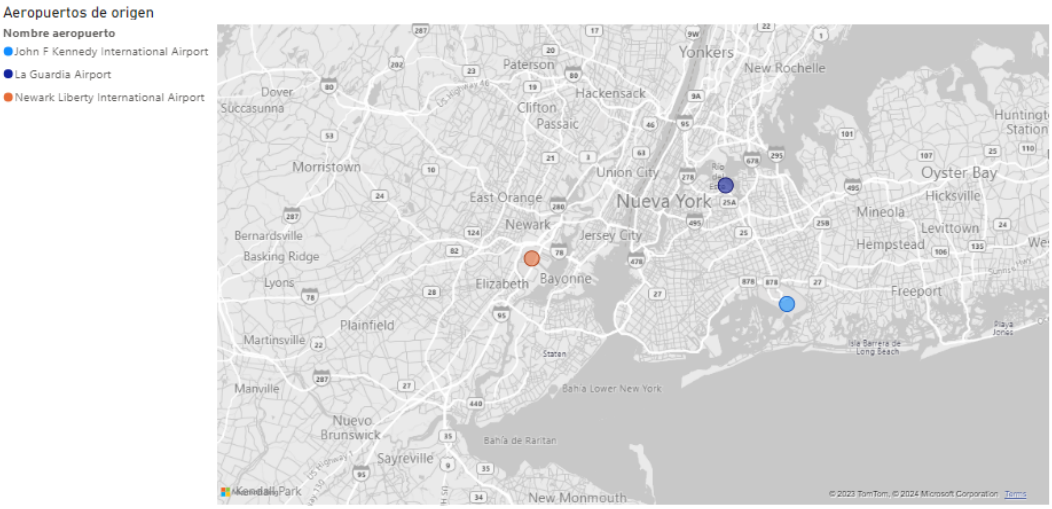


Figura 3.1: Localización de los aeropuertos de origen.

Los aeropuertos de destino se encuentran ubicados por toda la geografía norteamericana (ver Figura 3.2). En el Anexo A se especifican el nombre de los aeropuertos y las ciudades de destino.

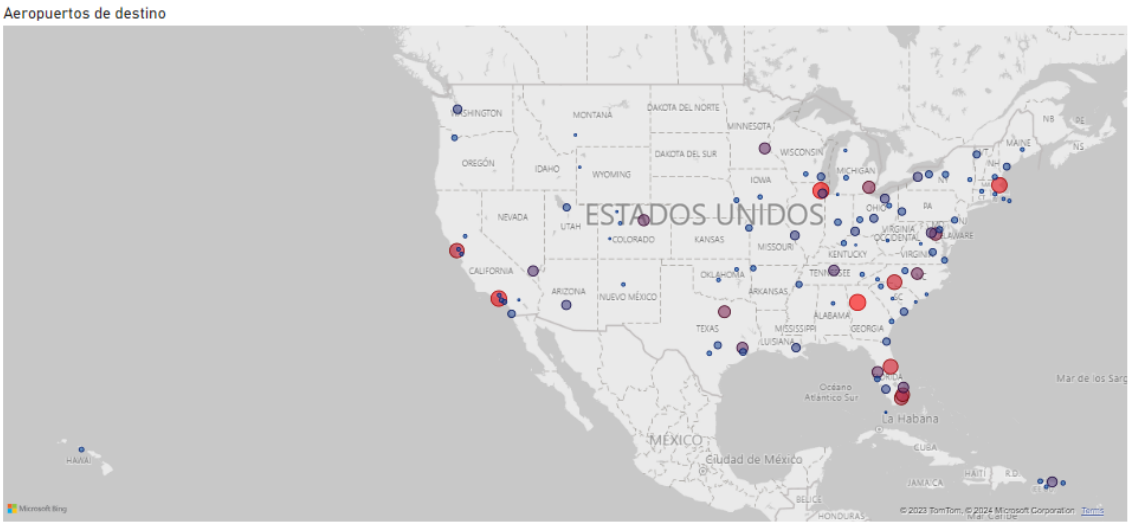


Figura 3.2: Localización de los aeropuertos de destino.

En la Figura 3.2 se han representado las ubicaciones de los aeropuertos de destino con burbujas cuyo tamaño y color es proporcional al número de vuelos operados, en azul número bajo de vuelos y en rojo los más altos. Se puede apreciar que la mayor operación de vuelos se localizan en las costas Oeste (aeropuertos de San Francisco y los Ángeles) y Este (Boston, Chicago, Atlanta y Orlando).

Nubosidad (*sky conditions*)

La nubosidad es un factor importante para los pilotos tanto para vuelos visuales (VFR) o por instrumentos (IFR). La fracción del cielo (domo celestial) cubierto por nubes es denominado *sky cover*. Se mide en *octas* de acuerdo a la Organización Meteorológica Mundial (WMO). En la Figura 3.3 se especifican la clasificación de la nubosidad en función de las octas [1].

(a) **SKC** 0/8 octas

(b) **FEW** 1/8 octas

(c) **FEW** 2/8 octas

(d) **SCT** 3/8 octas

(e) **BKN** 5/8 octas

(f) **OVC** 8/8 octas

Figura 3.3: Ejemplos de nubosidad por octas de [1].

Aerolíneas

En el *dataset* se muestran las operaciones de hasta 16 aerolíneas estadounidenses distintas. En la tabla 3.1 se muestran los códigos y el nombre de las distintas aerolíneas operadoras.

Cuadro 3.1: Aerolíneas

Código aerolínea	Nombre aerolínea
9E	Pinnacle Airlines
AA	American Airlines
AS	Alaska Airlines
B6	JetBlue Airways
DL	Delta Air Lines
EV	Atlantic Southeast Airlines
F9	Frontier Airlines
FL	AirTran Airways
HA	Hawaiian Airlines
MQ	American Eagle Airlines
OO	SkyWest
UA	United Airlines
US	US Airways
VX	Virgin America
WN	Southwest Airlines
YV	Mesa Airlines

Visibilidad

La visibilidad es una medida de la distancia a la que se puede distinguir un objeto, y puede variar con la dirección y el ángulo de visión, asi como la altura del observador. Entre los principales factores que afectan a la visibilidad se encuentran las precipitaciones, la niebla o la nubosidad [3].

En aeronáutica, se define la visibilidad como la mayor de las siguientes distancias:

1. La mayor distancia a la que un objeto negro de dimensiones adecuadas puede ser observado y reconocido, cerca del suelo sobre un fondo brillante.

2. la mayor distancia a la que se pueden ver e identificar luces de alrededor de 1000 candelas sobre un fondo negro.

Figura 3.4: Efecto de visibilidad en vuelos.

3.1.2. Tabla de vuelos

Se ha tomado [20] como origen de datos, debido al buen preprocesado de los mismos. Este *dataset* se basa en el registro de vuelos diarios de 3 principales aeroepuertos (JFK, EWR y LGA) de EEUU durante el año 2013. Los datos proporcionados para un vuelo son los siguientes:

Datos numéricos

- **Hora de salida**: hora de salida real del vuelo en un formato numérico específico.
- **Hora de salida programada**: hora de salida en que el vuelo fue programado inicialmente, en formato numérico específico.
- **Hora de llegada**: hora de llegada real del vuelo en un formato numérico específico.
- **Hora de llegada programada**: hora de llegada en que el vuelo fue programado inicialmente, en formato numérico específico.
- **Fecha del vuelo**: día en que el vuelo fue operado.
- **Distancia**: distancia recorrida del vuelo en km entre el aeropuerto de origen y destino.
- **Temperatura**: valor numérico que representa la temperatura en ºF.
- **Humedad relativa**.
- **Grado de visibilidad**: valor numérico en millas que representa el rango de visibilidad.

Datos categóricos

- **Aerolínea**: código IATA de la compañía aérea que opera el vuelo.
- **Aeropuerto de origen**: código IATA de la ciudad de origen del vuelo desde cualquiera de los 3 aeropuertos mencionados (JFK, EWR y LGA).
- **Aeropuerto de destino**: código IATA de la ciudad de destino del vuelo a distintos aeropuertos de norteamérica.
- **Retraso**: indicador de si el vuelo salió antes, después o en la hora programada.
- **Condiciones del cielo**: acrónimos que representan las distintas condiciones posibles del cielo. En la Tabla 3.2 se muestra el significado de cada una.

Cuadro 3.2: Categorías de condiciones del cielo. De [2]

Código	Descripción
CLR	*Sky clear at or below 12,000AGL*
FEW	*Few cloud layer 0/8ths to 2/8ths*
SCT	*Scattered cloud layer 3/8ths to 4/8ths*
BKN	*Broken cloud layer 5/8ths to 7/8ths*
OVC	*Overcast cloud layer 8/8ths coverage*
VV	*Vertical Visibility, indefinite ceiling*

3.1.3. Tabla de aeropuertos

De [4] se ha obtenido el *dataset* relacionado con la información relevante de los aeropuertos como:

- **Nombre**: nombre del aeropuerto.
- **Ciudad**: ciudad en la que se encuentra el aeropuerto.
- **Código IATA**: codigo IATA unívoco para cada aeropuerto.
- **Información adicional**: adicionalmente se especifica el tipo de aeropueto, la altitud o la ubicación en coordenadas.

3.1.4. Tabla de aerolíneas

De forma análoga a los aeropuertos, se ha obtenido de [17] la información relacionada con las aerolíneas:

- **Nombre**: nombre de la aerolínea.
- **País**: país al que pertenece la aerolínea.
- **Código IATA**: codigo IATA unívoco para cada aerolínea.

3.1.5. Relación de tablas

Para trabajar con los todos los datos descritos en las tablas anteriores, es necesario establecer un diagrama relacional de tablas (ver Figura 3.5)

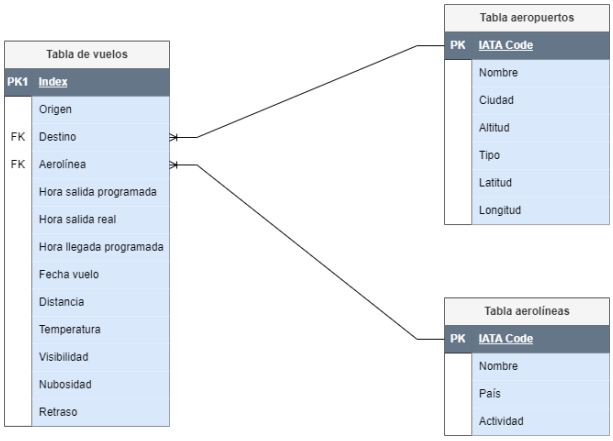


Figura 3.5: Digrama relacional de las tablas de datos.

En el diagrama de la Figura 3.5 se detalla claramente como se han relacionado las tablas. La **tabla de vuelos** es la tabla principal que contiene la información necesaria relacionada con la operación de los vuelos, cada registro se identifica con una *Primary Key* [1], que en este caso sería un índice unívoco para cada vuelo. Por otro lado, la **tabla de aeropuertos** y la **tabla de aerolíneas** tienen como *Primary Key* los códigos IATA correspondientes a los aeropuertos y aerolíneas respectivamente. Estas claves serán las *Foreign Key* [2] en la tabla de vuelos.

[1] Una o más columnas que identifican de forma única cada registro o fila de la tabla. No pueden existir PK repetidas.

[2] Grupo de una o más columnas que hacen referencia a una PK de otra tabla. Una FK puede ser parte de una PK.

3.2. Representación de los datos

La representación de los datos se realizará mediante una aplicación interactiva empleando *PowerBi* como herramienta de visualización. La arquitectura que se va a emplear para la representación se describe en la figura 3.6.

Figura 3.6: Flujo del proceso de representación de los datos.

En 3.6 se muestra el proceso que se va a seguir para representar los datos en un *dashboard*. En primer lugar, se toman los datos de la fuente de origen (ver [20]) en formato *csv*.[3] A pesar de que los datos ya se encuentran limpios y debidamente preprocesados, se ha decidido introducir una etapa de preprocesado de datos adicional para adaptarlos a nuestras necesidades como se describirá en el siguiente capítulo. Este tratamiendo de datos se realizará una parte con la librería *Pandas* de *Python* y otra con las herramientas que proporciona *PowerBi*.

Finalmente, se seleccionan los gráficos que mejor se adapten a la información que se dispone (gráfico de barras, lineal, anular, etc.) y gracias a las funcionalidades que proporciona *PowerBi* se construirá una aplicación en base a los gráficos y filtros que se hayan seleccionado.

[3]Un fichero *CSV* (Comma-Separated Values) es un tipo de archivo de texto empleado para almacenar datos de forma tabular, es decir, organizados en filas y columnas de forma que cada valor se encuentra separado por un delimitador, generalmente una coma.

3.3. Aplicación de un modelo de *Machine Learning*

Otro de los objetivos que se plantean en este proyecto es emplear los datos definidos en la sección 3.1 para entrenar un modelo de *Machine Learning* gracias a las funcionalidades que proporciona la librería *Scikit-Learn* de *Python*.

En primer lugar, es necesario un tratamiento previo para poder adaptar los datos a la entrada del modelo, y posteriormente categorizarlos en datos de entrenamiento y test (ver Figura 3.7).

Figura 3.7: Preprocesado y separación en datos de entrenamiento y test.

Esta categorización de los datos permite entrenar el modelo seleccionado con la muestra dedicada al entrenamiento, generalmente un 80 % del conjunto de datos total (ver Figura 3.8).

Figura 3.8: Entrenamiento del modelo.

El tipo de modelo seleccionado dependerá del tipo de problema a resolver, y por tanto influirá en el tipo de aprendizaje:

- **Aprendizaje supervisado**: problemas de regresión o clasificación. Modelos como regresión lineal o logística.

- **Aprendizaje no supervisado**: problemas de *clusterización*. Modelos como *K-means* son muy utilizados.

Una vez el modelo haya sido entrenado, se procede a un proceso de validación empleando las muestras de test (sobre el 20 % del conjunto de datos total) para obtener las métricas que medirán la calidad del modelo entrenado. (ver Firgura 3.9).

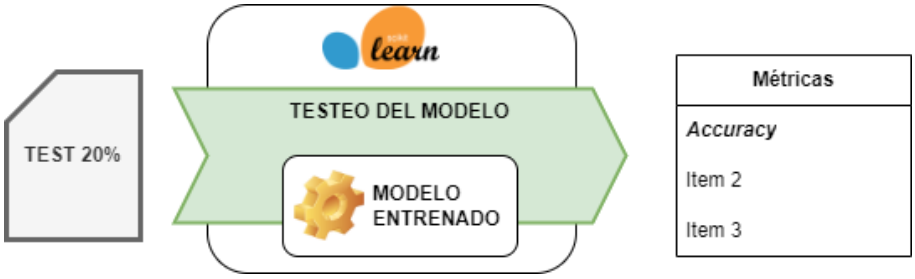


Figura 3.9: Validación del modelo entrenado.

En el capítulo 4 se detallará cómo se han implementado los procesos anteriormente descritos, asi cómo los modelos de *Machine Learning* que finalmente se han decidido emplear.

3.3.1. Algoritmos de clasificación

En este proyecto se emplearan algoritmos de clasificación para el problema que se planteará en el cappítulo 4, que consistirá en entrenar un modelo que sea capaz de clasificar en vuelos con retraso en salida o no. Para ello se emplearán los algoritmos de clasificación que se describen en este apartado:

- Regresión logística.
- Árbol de decisión.

Regresión logística

En primer lugar, destacar que los fundamentos de la regresión logística se basan en la regresión lineal. Esta última es empleada en problemas de regresión, es decir, el objetivo es *predecir un valor*, mientras que la regresión logística busca *etiquetar* los datos que pertenecen a un mismo grupo [14].

La regresión lineal se basa en la ecuación de la recta:

$$y = wx + b \tag{3.1}$$

y por lo tanto, también la regresión logística. Dado que el principal objetivo de la regresión logística es la clasificación, las características presentan más de una dimensión, por lo que se tratarán problemas binarios o multiclase. Se puede formalizar la ecuación 3.1 en forma vectorial de forma que:

$$\begin{aligned} h(x) &= b + w_1x_1 + w_2x_2 + ... + w_nx_n \\ &= \theta_0 + \theta_1x_1 + \theta_2x_2 + ... + \theta_nx_n \\ &= \Theta \cdot X^T \end{aligned} \tag{3.2}$$

donde:

$$\Theta = [\theta_0, \theta_1, \theta_2, ..., \theta_n] = [b, w_1, w_2, ..., w_n] \tag{3.3}$$

$$X = [x_1, x_2, ..., x_n] \tag{3.4}$$

El objetivo es que la función devuelva una etiqueta de clase, por ello es necesario aplicar una función no lineal. El siguiente desarrollo se centrará en una clasificación binaria, por ello se empleará como funcion no lineal la sigmoidal (ecuación 3.5), la cual devuelve 0 para valores menores que 0, y 1 en caso contrario.

$$\sigma(t) = \frac{1}{1 + e^{-t}} \tag{3.5}$$

Aplicando la función sigmoidal a 3.2 quedaría lo siguiente (representado en 3.10):

$$h(x) = \sigma(\Theta \cdot X^T) = \frac{1}{1 + e^{-\Theta \cdot X^T}} \tag{3.6}$$

El siguiente paso es definir una función de coste ($\phi(t)$), cuya misión es minimizar el error entre las entradas x y las salidas y.

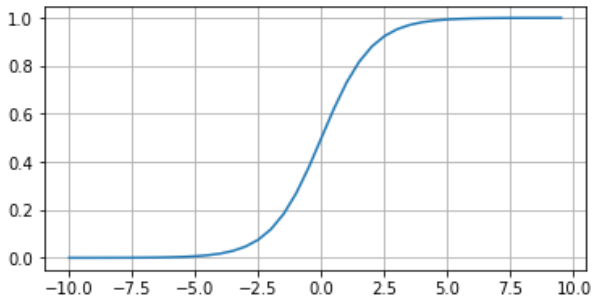


Figura 3.10: Función sigmoidal.

$$\phi(h(x)) = \begin{cases} -log(h(x)) & \text{si } y = 1 \\ -log(1 - h(x)) & \text{si } y = 0 \end{cases} \tag{3.7}$$

En el gráfico 3.11 se han representado las funciones de la expresión 3.7. Esta función de coste procesa la función $-log(h(x))$ cuando la etiqueta es $y = 1$, de forma que el error se minimiza penalizando si $y = 0$. Se obtiene un comportamiento análogo si la etiqueta es $y = 0$, ya que en este caso procesa $-log(1 - h(x))$, penalizando valores cercanos a 1.

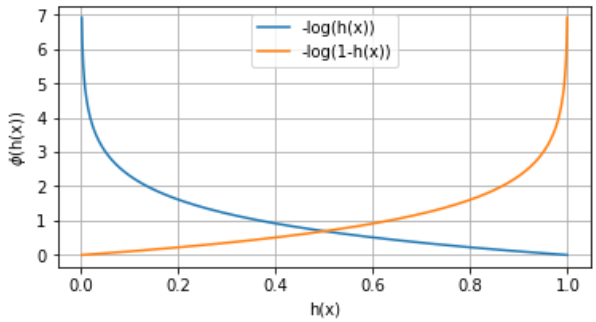


Figura 3.11: Función de coste.

Dado que la expresión 3.7 no es derivable, surge la necesidad de reescribirla de la siguiente forma:

$$\phi(h(x), y) = -y \cdot log(h(x)) - (1 - y) \cdot log(1 - h(x)) \tag{3.8}$$

En la ecuación 3.8 si $y = 1$ entonces $\phi(h(x)) = -log(h(x))$, y por lo tanto si $h(x) = 1$ o cercano a 1 se premia la predicción correcta $y = 1$. Cuando $y = 0$ la función de coste es $\phi(h(x)) = -log(1 - h(x))$, si $h(x) = 0$ o cercano a 0 se premia la predicción correcta $y = 0$.

Para hallar los parámetros de Θ óptimos, se aplicará el algoritmo de *descenso por gradiente*. Por ello, es necesario discretizar la expresión 3.8 y aplicar la derivada parcial respecto a los parámetros de Θ.

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \cdot log(h(x_i)) - (1 - y_i) \cdot log(1 - h(x_i))] \tag{3.9}$$

La ecuación 3.9 representa la expresión 3.8 discretizada, a la cual se le aplicarán derivadas parciales sobre Θ (ecuación 3.10).

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{n}\sum_{i=1}^{n}[(h(x_i) - y_i) \cdot x_j] \tag{3.10}$$

El algoritmo del descenso por gradiente [4] optimiza los parámetros de Θ (expresión 3.3) de forma que minimice la función de coste 3.9 en la dirección calculada a partir del gradiente, es decir, a partir de sus derivadas parciales (ecuación 3.10). El objetivo es que los valores de Θ converjan a un valor óptimo.

Árbol de decisión

Un modelo de árbol en aprendizaje automático se configura como una estructura de árbol análoga a un gráfico de flujo, véase la Figura 3.12. En este esquema, un nodo central representa una característica (o atributo), la conexión es una pauta de decisión, y cada extremo del nodo o la hoja indica el resultado, siendo el nodo superior denominado el nodo raíz. Este tipo de estructura arbórea es reconocido como **árbol de clasificación**, en el cual cada desviación engloba un conjunto de atributos o normativas de categorización vinculadas a una etiqueta de clase específica ubicada al final de la desviación. [21]

El objetivo de un árbol de decisión es aprender a dividir los datos en función del valor de un atributo, de forma que busca un mínimo local de forma recursiva tomando como medidas la entropía en base a un conjunto masivo de datos.

Las ventajas de los árboles de decisión en el aprendizaje automático se manifiestan de la siguiente manera:

- El gasto asociado al empleo del árbol para predecir datos disminuye a medida que se

[4] El descenso por gradiente es un algoritmo de optimización utilizado para encontrar el mínimo de una función, iterativamente ajustando los parámetros en la dirección del gradiente negativo de la función para minimizarla.

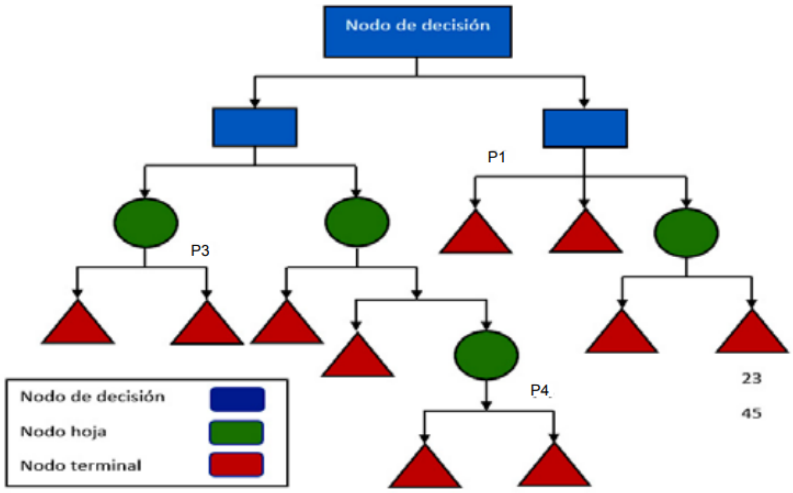


Figura 3.12: Estructura de un árbol de decisión.

incorporan más puntos de datos.

- Es eficaz tanto para datos numéricos como categóricos.
- Tiene la capacidad de modelar problemas con múltiples resultados.
- La fiabilidad de un árbol puede ser cuantificada y sometida a pruebas.
- Tiende a mantener su precisión, incluso cuando se aparta de las suposiciones de los datos de origen.

No obstante, se presentan ciertas limitaciones:

- La inclusión de datos categóricos con múltiples niveles puede sesgar los resultados a favor de los atributos con mayor cantidad de niveles.
- Los cálculos pueden volverse complicados al enfrentarse a la incertidumbre y a numerosos resultados interrelacionados.

Las **medidas de selección** de los atributos es una heurística para seleccionar un criterio que divida los datos de forma óptima. Entre estas medias las más utilizadas son:

- **Ganancia de información:** propiedad estadística que mide homgeneidad. Es utilizado para determinar qué atributo debe ser seleccionado como nodo de decisión en cada nivel del árbol, es decir, mide cuánto un atributo contribuye a reducir la incertidumbre en la clasificación de los datos.

La ganancia de información se calcula comparando la incertidumbre antes y después de la partición del conjunto de datos en función del atributo seleccionado. El objetivo es maximizar esta ganancia de información con el atributo elegido en cada paso, a mayor ganancia más útil es el atributo seleccionado para la clasificación de los datos.

- **Entropía:** es una medida de homogeneidad de datos aleatorios de un un nodo del árbol. El valor de la entropía es muy bajo o nulo si todos los datos son similares. Se puede medir la entropía de la siguiente forma:

$$E = -\sum_{i=1}^{k} P_i \cdot log_2 P_i \tag{3.11}$$

donde P_i es la probabilidad de un atributo dado. La entropía es 1 cuando existen el mismo número de muestras positivas y negativas.

- **Gini:** es una medida de impureza de las observaciones contenidas en un nodo. Si es 0, el nodo es totalmente puro. El gini se calcula como:

$$Gini = \sum_{i=1}^{k} P_i(1 - P_i) \tag{3.12}$$

Capítulo 4

Implementación y experimentos

En este capítulo se desarrollarán los procesos planteados en el capítulo 4 esructurándose de la siguiente forma:

- *Dashboard* de visualización de datos y análisis de los datos representados.
- Aplicación de un modelo de regresión logística.
- Aplicación de un modelo basado en un árbol de decisión.

4.1. *Dashboard* de visualización de datos

El primero de los objetivos planteados en la Tabla 1.1 es la representación de los datos explicados en 3.1 en un *dashboard* interactivo, de forma que el usuario sea capaz de visualizar los datos más relevantes de forma sencilla e interactiva.

Para la implementación de la aplicación se ha utilizado *PowerBi* como herramienta de visualización de datos. A continuación, se detallarán los *tabs* en los que se ha clasificado la información y los gráficos que se han utilizado para mostrar los datos.

4.1.1. Vuelos

En esta pestaña se ha representado la información general como el número de registros de vuelos total, aeropuertos de destino o las aerolíneas que operan (ver Figura 4.1).

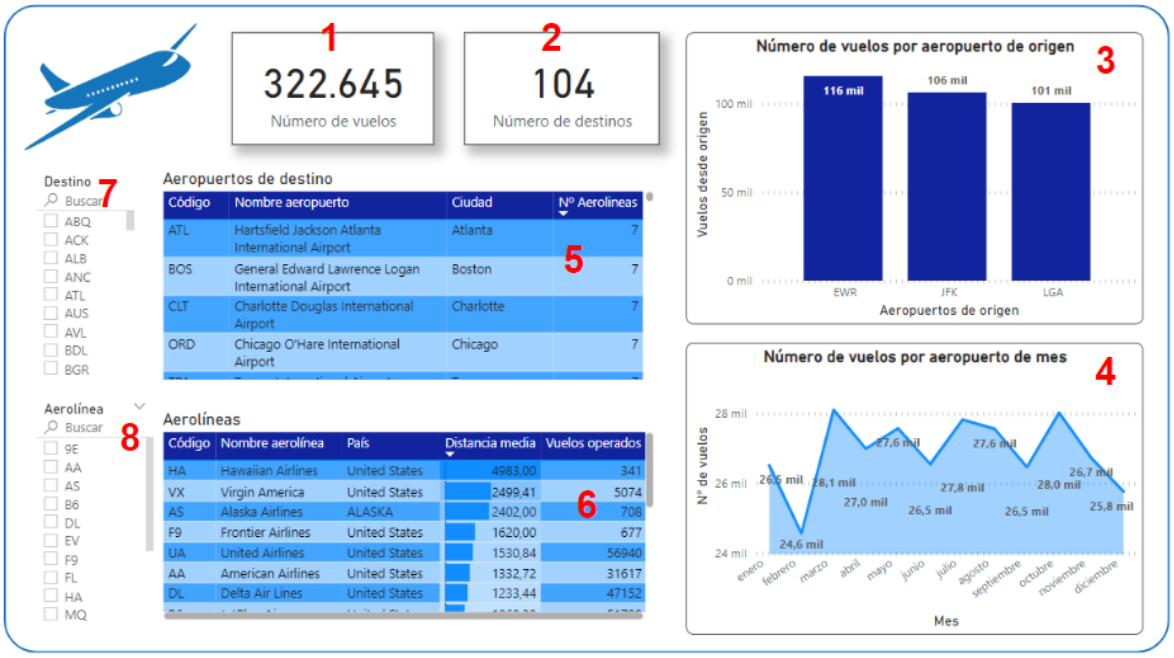


Figura 4.1: Información general de vuelos.

En la Figura 4.1 se muestra la representación de gráficos y tablas relacionados con datos generales de vuelos. Los elementos empleados para la representación de los datos se pueden clasificar en:

Tarjetas de visualización

1. Número de registros (vuelos) seleccionados. Si no se ha aplicado ningún tipo de filtrado, el número de registros total es de 322.645 vuelos.

2. Número de ciudades destino. Se trata de una cantidad unívoca, por lo que no se está considerando la frecuencia en la que operan los vuelos, únicamente si dicho destino se encuentra al menos una vez entre los vuelos seleccionados.

Gráficos

3. Número de vuelos operados por cada aeropuerto de origen. Este conjunto de datos contemplaba 3 aeropuertos de origen (EWR, JFK y LGA).

4. Número de vuelos operados por cada mes. Se ha empleado un gráfico lineal para apreciar la evolución temporal.

Tablas

5. Para los vuelos seleccionados, se muestran los diferentes aeropuertos de destino y datos relevantes como el nombre del aeropuerto la ciudad y el número de aerolíneas que operan en esos aeropuertos.

6. Información relevante de aerolíneas para los vuelos seleccionados como el nombre de la operadora, el país, la distancia media que operan y el número de vuelos.

Filtros

Para hacer una interacción más dinámica se han utilizado filtros para customizar los elementos de visualización anteriores en base a las preferencias del usuario.

7. Filtro por ciudad de destino.

8. Filtro por aerolínea.

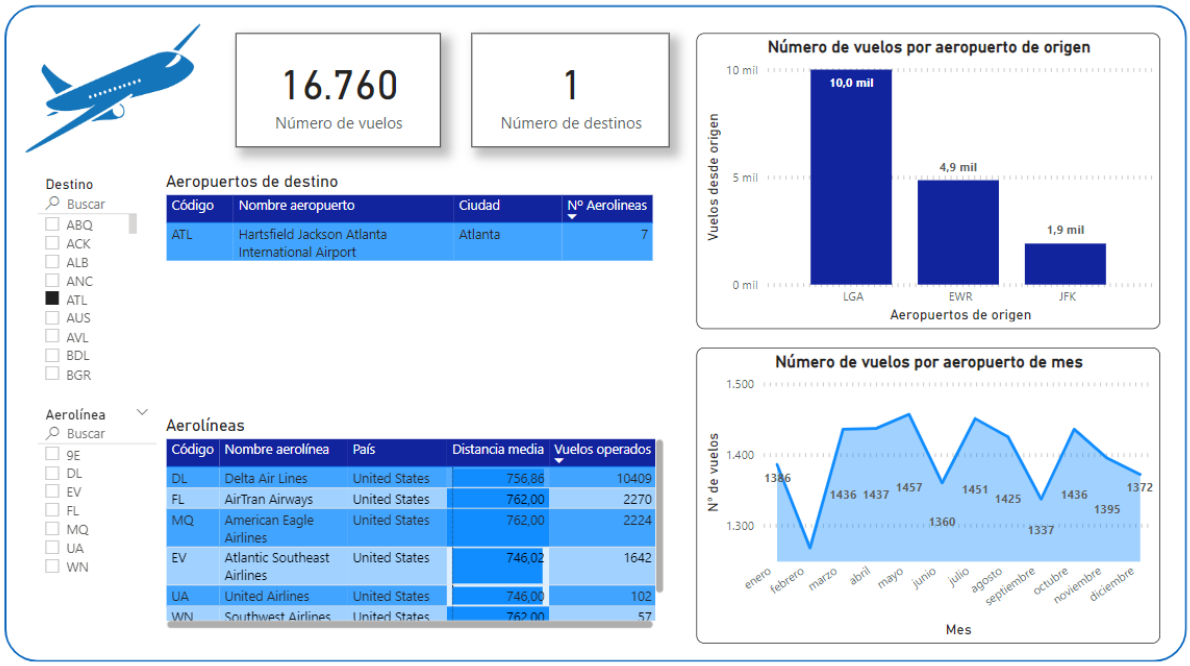


Figura 4.2: *Tab* de vuelos filtrando por ciudad de destino (ATL)

Esta *tab* permite visualizar de forma directa información general de los datos, ya que es posible ver el número de vuelos asi como los aeropuertos de origen y destino afectados. Considerando el siguiente ejemplo práctico, si se aplica un filtro por ciudad de destino como Atlanta

(ver Figura 4.2), se observa que existen hasta 16.760 vuelos con ese destino, además que la mayoría de esos vuelos (10.000 vuelos) operan desde LGA. Se pueden apreciar épocas estacionales a lo largo del año. En la tabla de aerolíneas se pueden ver todas las aerolíneas (hasta 7) que ofrecen como destino ATL. Ordenando por *Vuelos operados*, la aerolínea que mas vuelos opera a ATL es *Delta Airlines* con 10.409 vuelos.

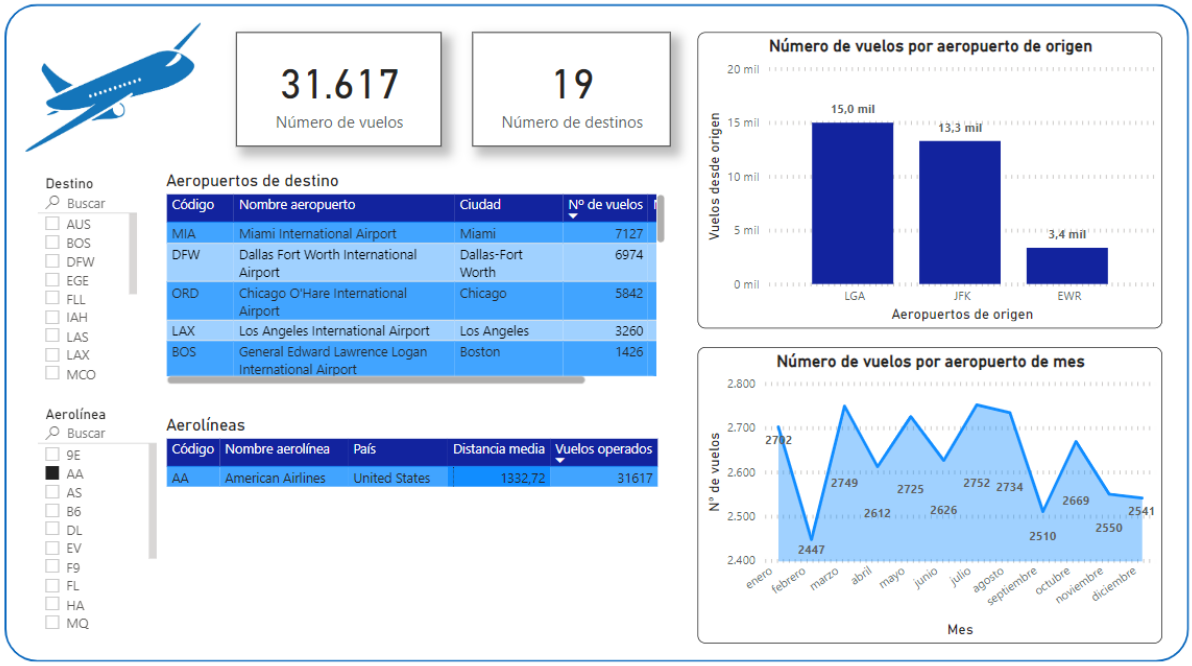


Figura 4.3: *Tab* de vuelos filtrando por aerolínea (AA)

Si ahora se considera el filtro por aerolínea (ver Figura 4.3), por ejemplo, *American Airlines* (AA) se observa que el número de vuelos que opera es de 31.617, con un total de 19 destinos distintos, siendo el más frecuente Miami con 7.127 vuelos. Opera fundamentalmente desde los aeropuertos LGA y JFK.

4.1.2. Mapa de destinos

El objetivo de esta *tab* es situar geograficamente las diferentes ciudades de destino y clasificarlas con una escala de color en función del número de vuelos que operan a dichos destinos (ver Figura 4.4).

Tarjetas de visualización

Se trata de las mismas tarjetas utilizadas en la *tab* anterior.

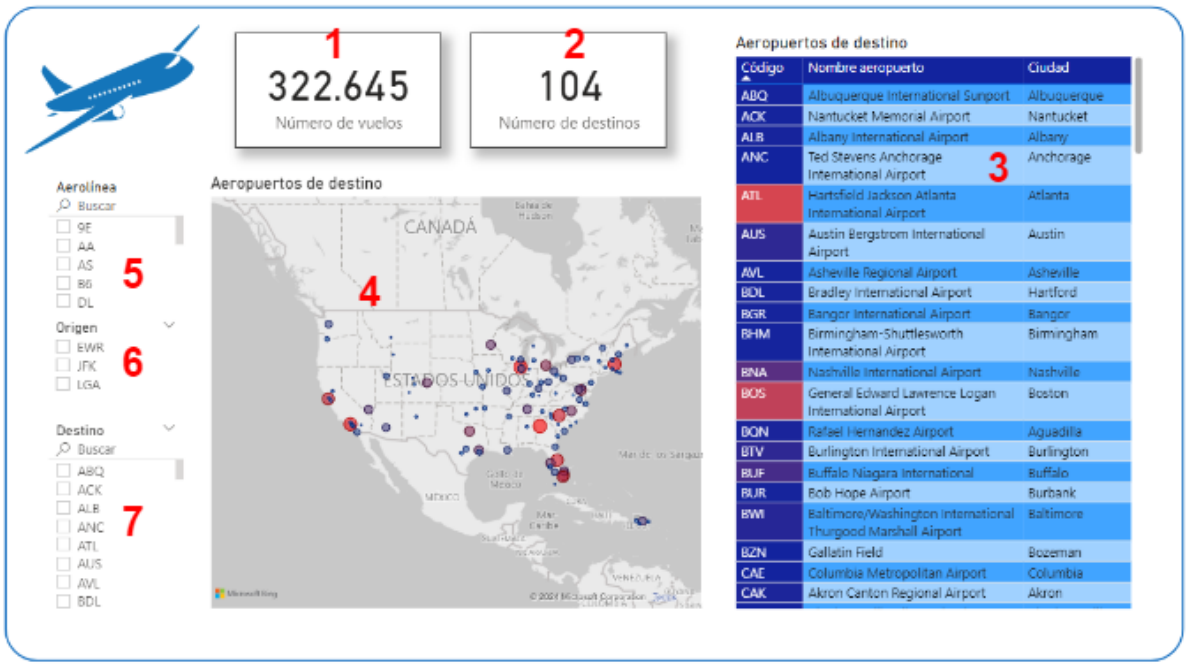


Figura 4.4: *Tab* de ciudades de destino representados en un mapa interactivo.

1. Número de registros (vuelos) seleccionados.

2. Número de ciudades destino.

Tabla

3. Aeropuertos y ciudades de destino seleccionados. En la columna código se ha aplicado una regla para resaltar aquellos aeropuertos con mayor operación, siguiendo una escala de color degradada de azul a rojo de forma que el azul representa número bajo de vuelos y el rojo una mayor operación hacia ese destino.

Mapa geográfico

5. Se ha empleado una funcionalidad de *PowerBI* que consiste en un mapa como objeto visual. Para reprsentar los puntos se ha necesitado incorporar de la tabla de aeropuertos la información relativa a la localización (latitud y longitud), ver apartado 3.1.3. Además las ubicaciones se han representado con burbujas o *bubbles* que serán de mayor o menor tamaño proporcional al número de vuelos operados a ese destino, de forma análoga el color de las burbujas varían en función de la operación, siendo azul destinos con menor

número de vuelos y rojo en caso contrario.

Filtros

6. Filtro por aerolínea.

7. Filtro por aeropuerto de origen.

8. Filtro por aeropuerto de destino.

Esta visualización permite localizar de forma directa las diferentes ciudades de destino mediante los filtros. Por ejemplo, es posible ubicar los destinos operados por *American Airlines* desde el aeropuerto EWR modificando los filtros de *Aerolínea* y *Origen* (ver Figura 4.5). En este caso se observa hay vuelos a 3 ciudades, siendo el de mayor operación el aeropuerto de Dallas (DFW).

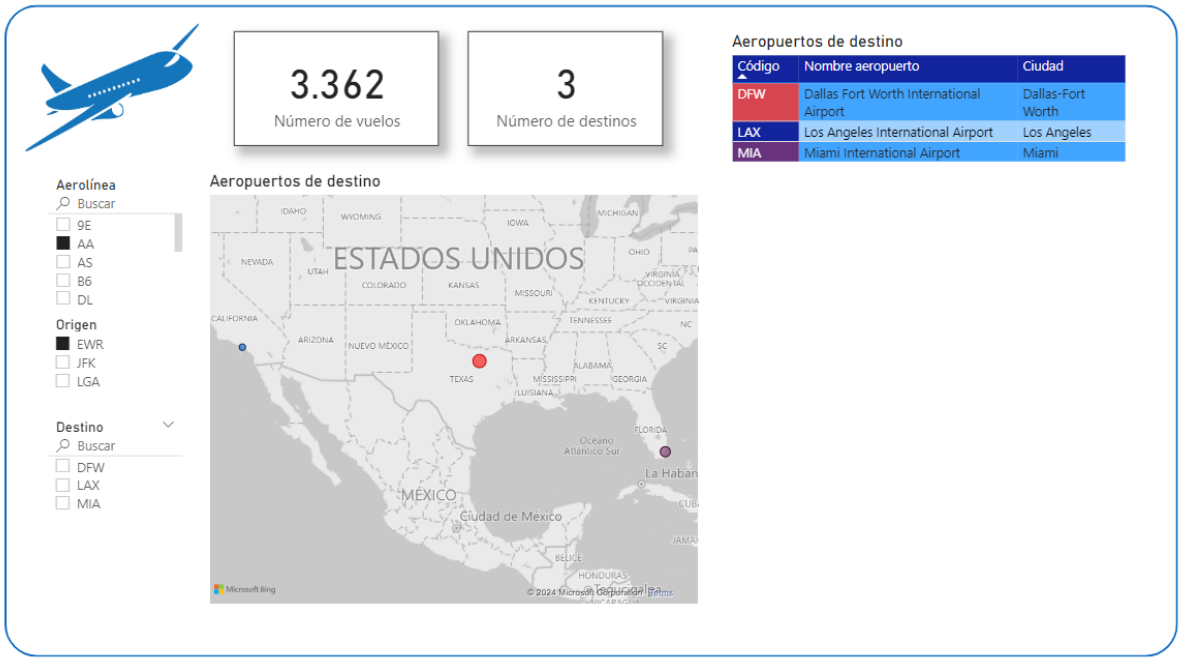


Figura 4.5: Mapa de ciudades filtrando con aerolínea y aeropuerto de origen.

Por otro lado, filtrando por ciudad de destino únicamente se puede visualizar la ubicaión exacta del aeropuerto (ver Figura 4.6).

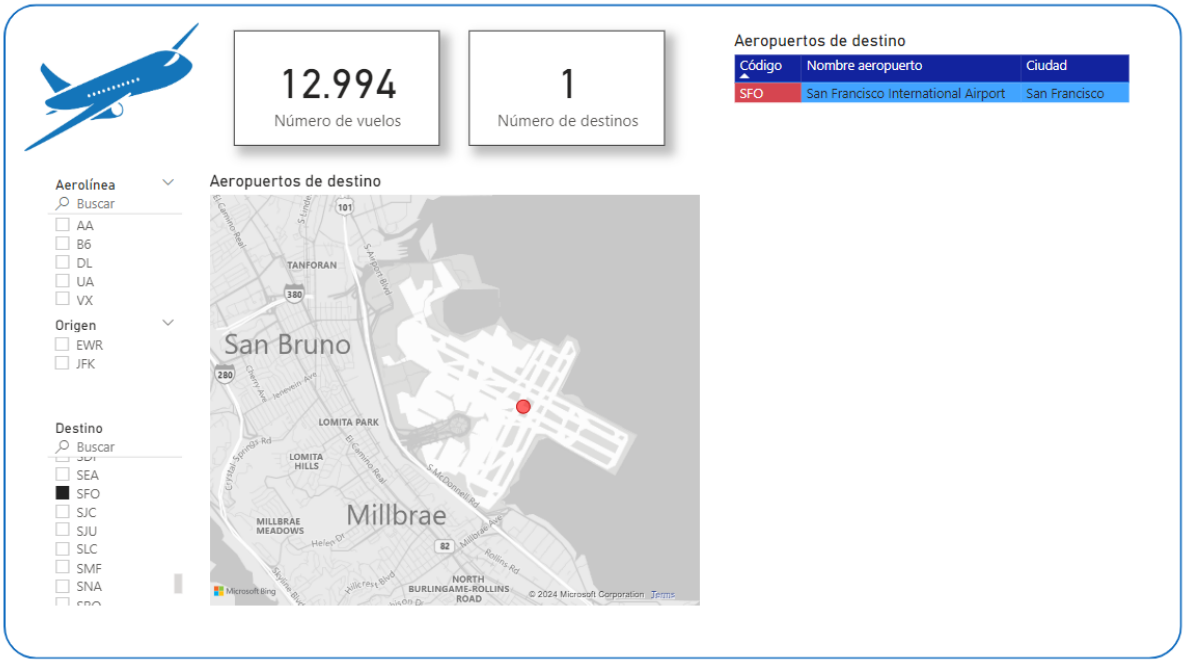


Figura 4.6: Ubicación del aeropuerto de un destino seleccionado.

4.1.3. Salidas y llegadas

El objetivo de esta pestaña es representar la evolución horaria a lo largo de un día de las salidas y llegadas respectivamente. Además se han categorizado las salidas/llegadas en hora, adelantadas o atrasadas (ver Figura 4.7).

Gráficos

1. Número de salidas por cada hora. Se muestra un promedio de todos los registros con la misma hora de salida.

2. Número de llegadas por cada hora de forma análoga a las salidas.

Filtros

3. Filtrado por mes, lo que permite ajustar el promedio por horas a un mes concreto.

4. Filtro para aerolíneas, permite observar el comportamiento por operadora.

5. Por aeropuerto de origen, entre los 3 ya mencionados anteriormente.

6. Por aeropuerto de destino.

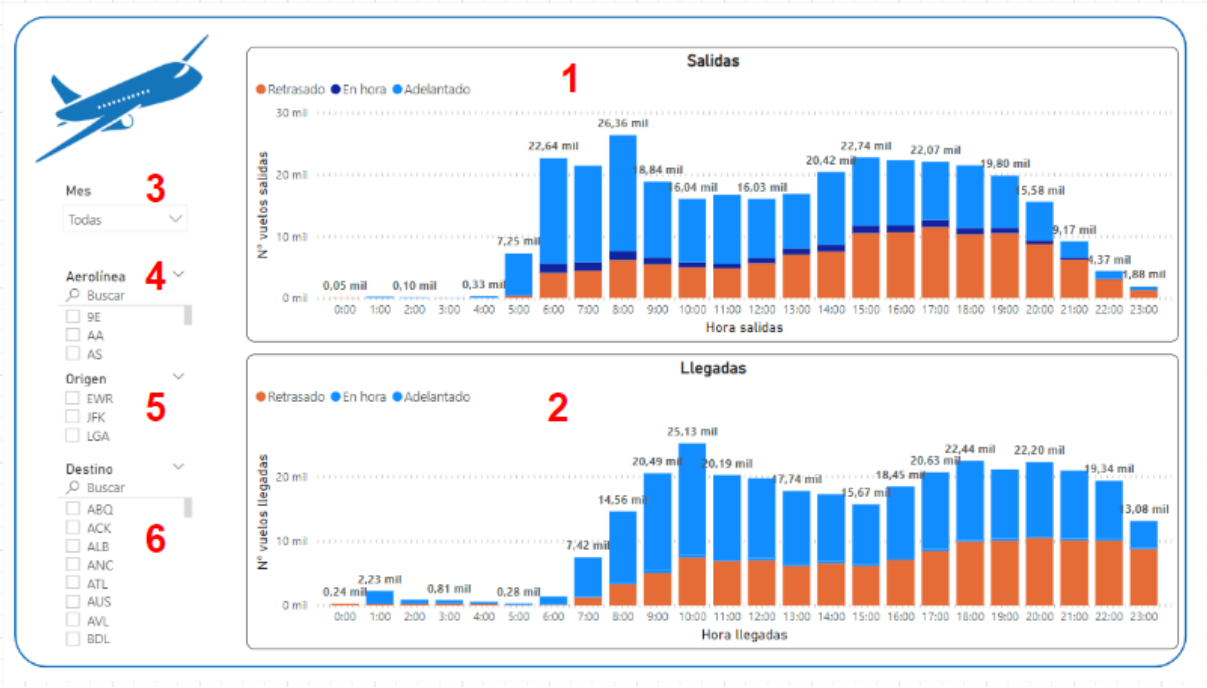


Figura 4.7: Información de salidas y llegadas por horas.

Observando los gráficos, se puede apreciar un comportamiento en la distribución de los vuelos, ya que es en las horas pico (entorno a las 8:00 y a las 17:00) donde hay un mayor tráfico aéreo. Además, se puede apreciar un patrón en la distribución de los vuelos con retraso (naranja), pues se aprecia que existen 2 picos de retraso correspondiente a las horas punta como era de esperar. Sin embargo, es entorno a las 17:00 donde más retrasos se han registrado, a pesar de que es a las 8:00 donde existe una mayor operación de vuelos.

4.1.4. Retrasos en vuelos

En esta pestaña esta dedicada a mostrar la información relacionada con aquellos vuelos que han sufrido retrasos en salidas o llegadas (ver Figura 4.8).

Gráficos

1. Gáfico anular, representa la proporción de vuelos (número total y porcentaje) con retraso, adelantados o en hora para salidas. Además se ha integrado una tarjeta de visualización que muestra el retraso promedio en salidas.

2. Gráfico de barras apiladas que muestra para los aeropuertos de origen el número de vuelos

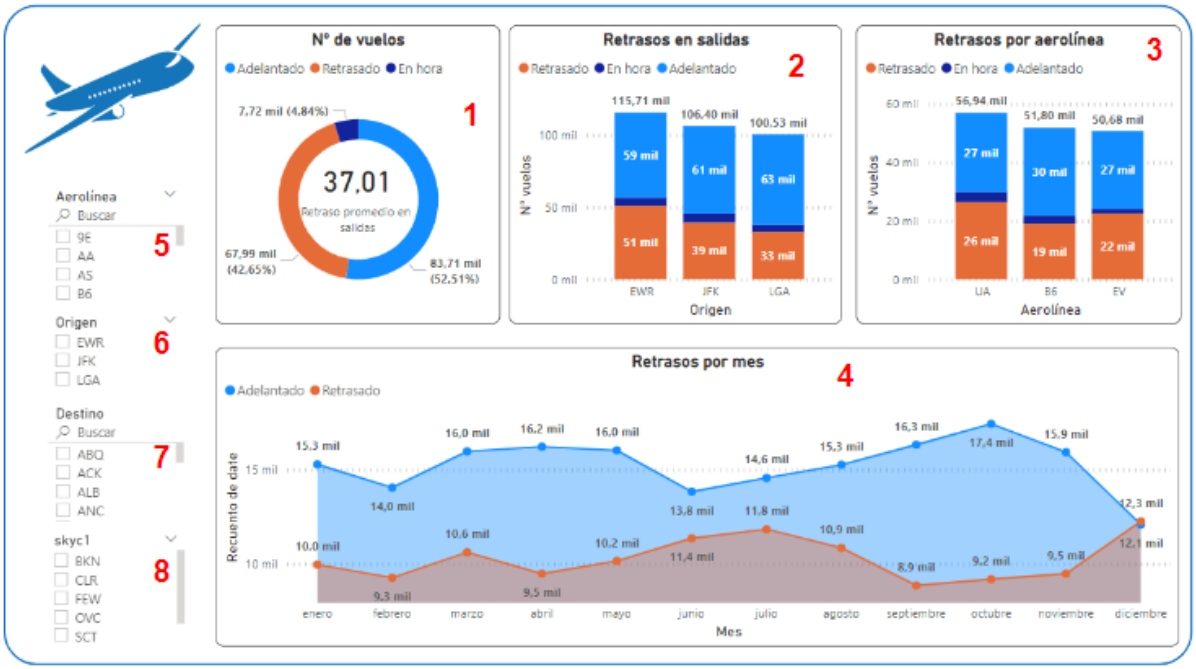


Figura 4.8: Información de retrasos en vuelos.

operados categorizando las salidas en retrasos, adelantadas o en hora.

3. Gráfico de barras apiladas, muestra el top 3 de aerolíneas con mayor número de vuelos operados categorizados por salidas con retraso, adelantadas o en hora.

4. Gráfico de áreas. Muestra la evolución mensual del número de vuelos que sufrieron retrasos frente a los que no diferenciando por areas de distinto color.

Filtros

Los gráficos anteriormente mencionados pueden ajustarse a las preferencias del usuario mediante los siguientes filtros:

5. Filtro por aerolínea.

6. Filtro por aeropuerto de origen.

7. Filtro por aeropuerto de destino.

8. Filtro por categorías de condiciones del cielo.

Observando el gráfico de retrasos por aeropuerto de origen, el comportamiento de retrasos es lógico, pues a mayor número de vuelos operados existe un mayor número de retrasos. Sin embargo, en el top 3 de aerolíneas, se aprecia que *JetBlue Airways* (B6) tiene un menor número de retrasos (19.000) frente a los 22.000 que tiene *Atlantic Southest Airlines* (EV) a pesar de operar este un mayor número de vuelos. Por otro lado, en la evolución mensual se puede apreciar de forma clara los meses en los que se sufren mayor número de retrasos.

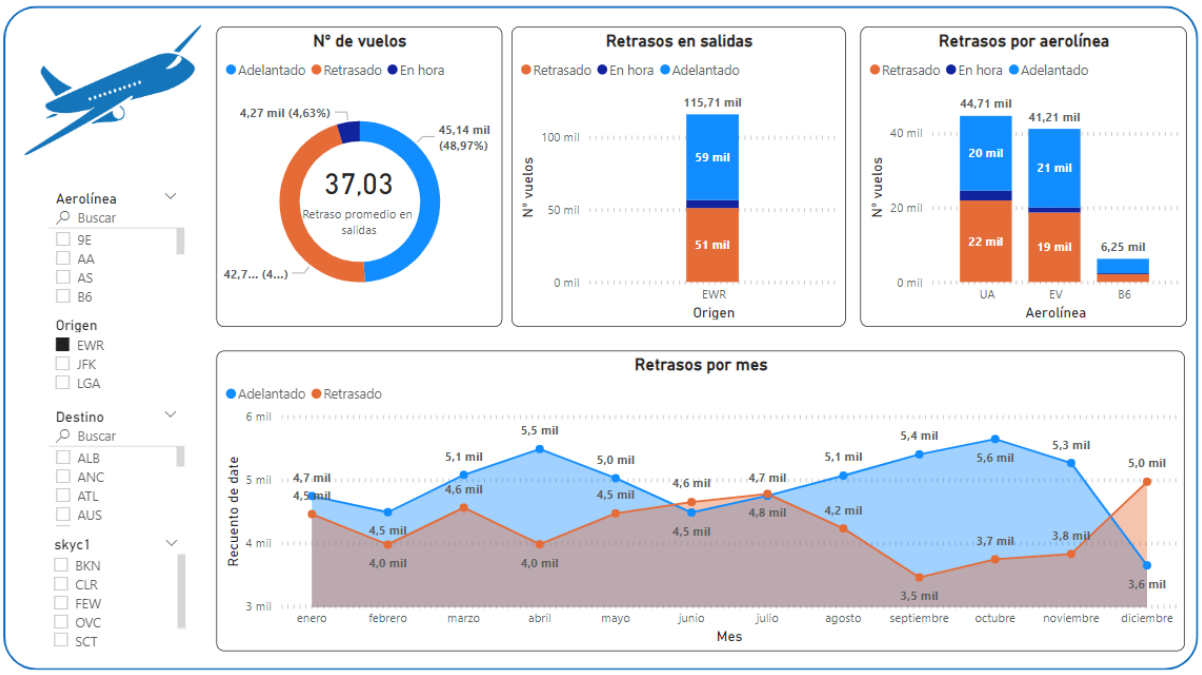


Figura 4.9: Pestaña de retrasos filtrando por aeropuerto de origen (EWR).

A continuación, se expone el ejemplo práctico filtrando por aeropuerto de origen EWR observando lo siguiente (ver Figura 4.9):

- Atendiendo al gráfico anular, la proporción de vuelos con retraso frente a los que no es muy equilibrada con un 46,4 % de retrasos y 37 minutos en promedio.

- Observando el top 3 de serolíneas, las que más operan desde EWR son *United Airlines* (UA) y *Atlantic Southest Airlines* (EV) con mucha diferencia respecto al tercero.

- En la evolución mensual, se aprecia claramente un sorpaso de retrasos frente a vuelos en hora en meses clave (verano-Navidad).

4.1.5. Condiciones climáticas

Los datos también muestran un registro de las condiciones climáticas para cada vuelo como la temperatura, la humedad relativa, visibilidad y condiciones del cielo. Estas variables también se han representado en el *dashboard* (ver Figura 4.10).

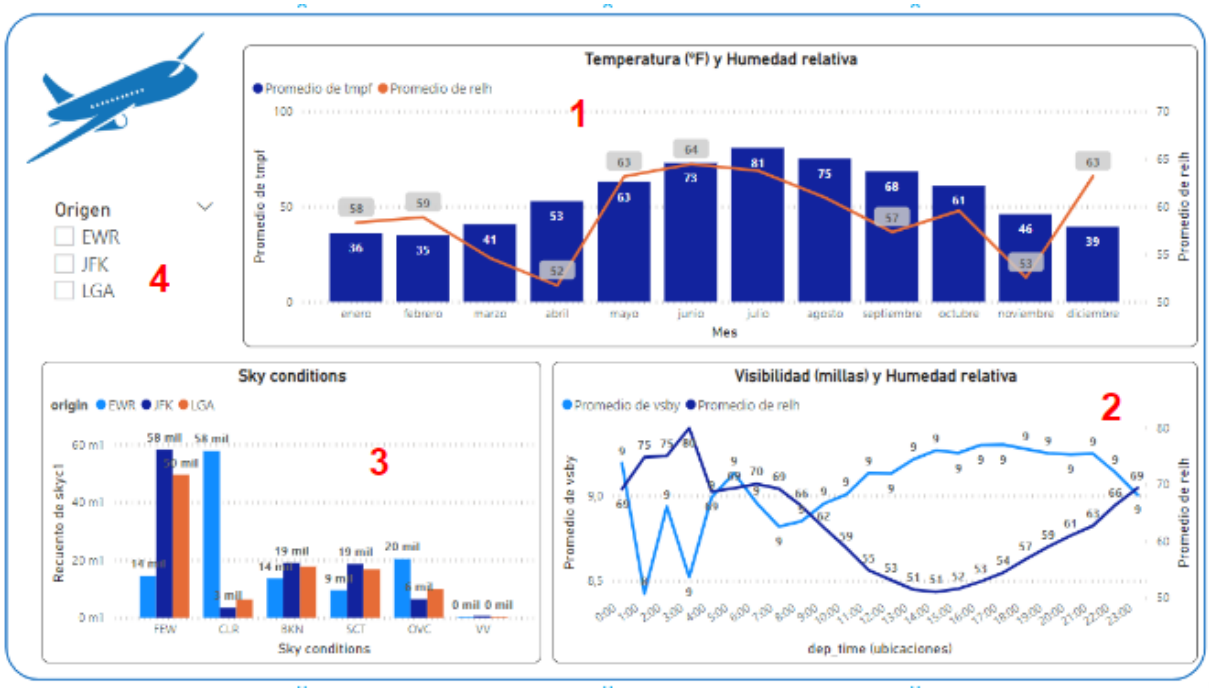


Figura 4.10: Información sobre las condiciones climáticas.

Gráficos

1. Gráfico de barras para representar la temperatura (ºF) y gráfico lineal para la humedad relativa en promedio por mes.

2. Gráficos lineales para representar la visibilidad y la humedad relativa promedio con el objetivo de ver la evolución horaria en el día.

3. Gráfico de barras agrupadas por cada categoría de condición del cielo (ver Tabla 3.2) mostrando el número de vuelos por cada aeropuerto de origen.

Filtros

4. Filtro por aeropuerto de origen.

En el gráfico 1 se observa claramente el comportamiento estacional de la temperatura y la humedad relativa, siendo más altas en la estación de verano más bajas en invierno. En el gráfico 2 se observa un comportamiento del grado de visibilidad frente a la humedad, y es que se aprecia una tendicia antagónica entre ambas, es decir, cuando aumenta el grado de visibilidad disminuye la humedad relativa y viceversa.

Atendiendo al gráfico 3, la mayoría de los vuelos operan en condiciones FEW y CLR, es decir, de poca nubosidad.

4.1.6. Ranking aerolíneas y destinos

El objetivo es mostrar un *ranking* de aerolineas y aeropuertos de destino con mayor número de salidas y llegadas con retrasos (ver Figura 4.11).

Figura 4.11: Ranking por aerolíneas y aeropuertos de destino.

Gráficos

1. Gráfico de barras horizontal para mostrar *ranking* de aerolíneas con mayor número de vuelos.

2. De forma análoga a las aerolíneas, se muestra un gráfico de barras horizontal con el *ranking* de los destinos con mayor número de vuelos.

Filtros

3. Filtro por mes.

4. Filtro por aeropuerto de origen.

5. Filtro de salidas por retrasos, adelantados o en hora (ver Figura 4.12).

6. Filtro de llegadas por retrasos, adelantados o en hora (ver Figura 4.13).

En la Figura 4.11 se muestra un ranking de aerolíneas y destinos por número de vuelos. Sin aplicar ningún filtrado, se muestran únicamente las aerolíneas con mayor operación de vuelos y los destinos más frecuentados.

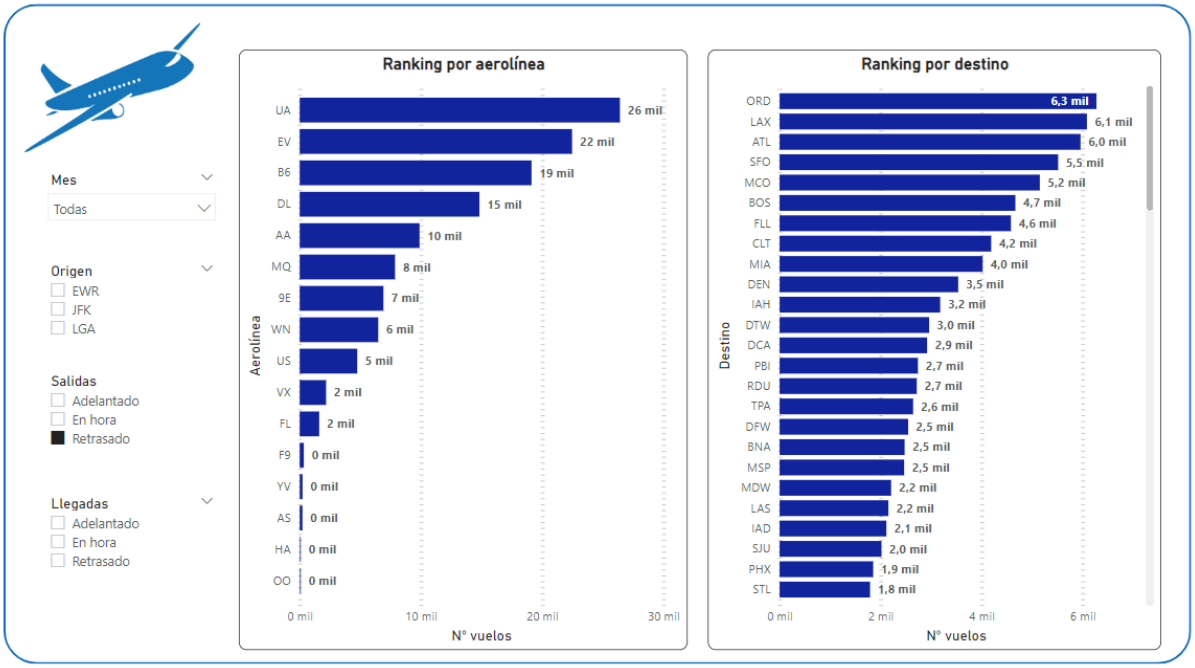


Figura 4.12: Ranking por aerolíneas y aeropuertos de destino con mayor retraso en salidas.

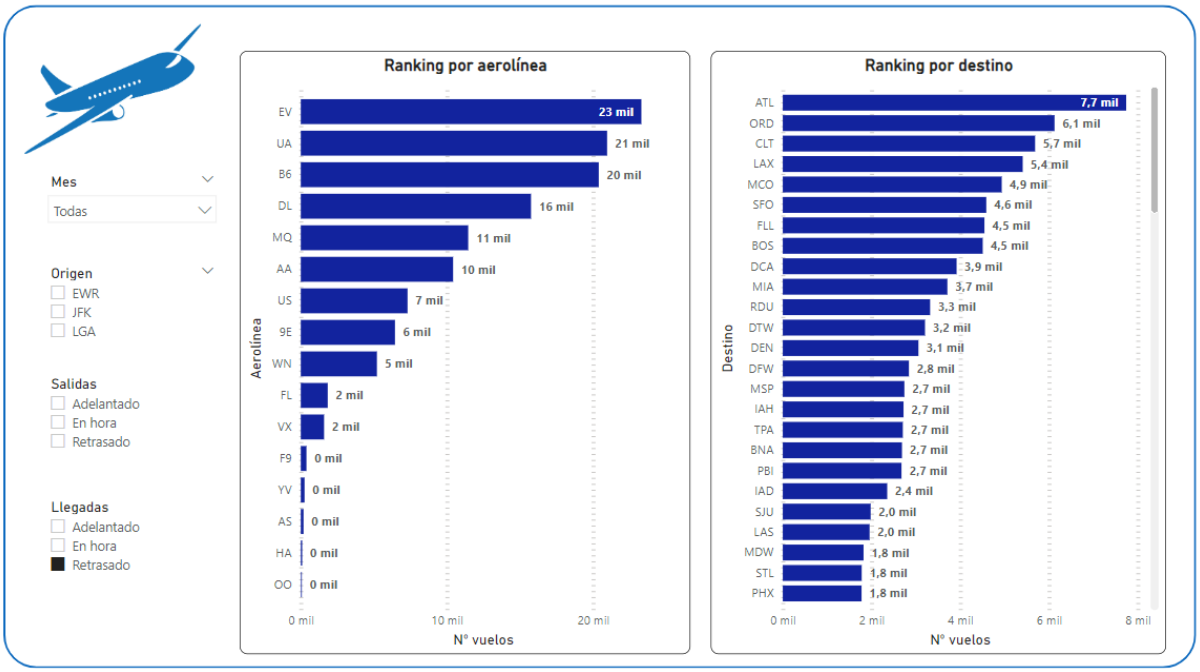


Figura 4.13: Ranking por aerolíneas y aeropuertos de destino con mayor retraso en llegadas.

4.2. Modelo de *Machine Learning*

En esta sección se detallará la aplicación de un modelo de *Machine Learning* como se ha descrito anteriormente en 3.3. Para ello, se hará uso de la librería *Scikit-Learn* de *Python* (ver sección 2.3). Se emplearán los algoritmos de clasificación denominados **regresión logística** y **árbol de decisión** descritos en 3.3.1. El objetivo será determinar si un vuelo según unas características de entrada va a sufrir retraso o no.

4.2.1. Preprocesamiento de datos

Previamente a la aplicación de los modelos, es necesario un preprocesado de los datos para adaptarlos a la entrada del modelo.

Selección de inputs numéricos

En el apartado 3.1.2 se describen los datos de tipo numérico que se encuentran en el *dataset* de origen. Para este caso se han seleccionado como características numéricas del modelo:

- Hora de salida programada.

- Visibilidad.
- Trimestre.

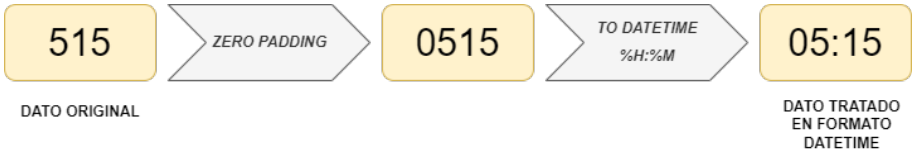


Figura 4.14: Procesado de datos relativo a horas.

Para el tratamiento de la hora de salida se realiza el proceso descrito en la Figura 4.14. En primer lugar se verifica si el número es de 4 o 3 dígitos. En el primer caso, los 2 primeros dígitos representan la hora y los dos últimos los minutos, de forma que mediante el formato **%H %M** la función *datetime* de la librería *pandas* es capaz de hacer la conversión correctamente a formato *timestamp*. Por ejemplo, el número 2235 representa la hora 22:35 (22 horas y 35 minutos). Sin embargo, para las horas comprendidas entre las 01:00 y 09:59 es necesario un paso adicional, ya que el dato se representa con 3 dígitos, por ejemplo, las 05:15 se muestra como 515, por ello es necesario añadir un cero adicional, tal como se describe en la Figura 4.14.

Una vez realizada la conversión de las horas de salida a formato *timestamp*, resulta sencillo identificar las horas y los minutos. En este caso, únicamente serán considerados las horas, de esta forma, los vuelos comprendidos entre las 09:00 y las 09:59 serán categorizados como hora de salida a las 9. Es importante destacar que las fechas se ignoran en este proceso, ya que ello supondría añadir una fuerte correlación en los datos, lo cual no interesa en un modelo de *Machine Learning*.

La variable *trimestre* se ha calculado a partir de la fecha del vuelo de tipo *datetime*, ya que gracias al atributo *quarter* devuelve el trimestre del año que corresponda:

- *Trimestre 1*: Enero - Marzo
- *Trimestre 2*: Abril - Junio
- *Trimestre 3*: Julio - Septiembre
- *Trimestre 4*: Octubre - Diciembre

Finalmente, es necesario añadir un último paso al tratamiento de datos numéricos, y es normalizar los valores para que el modelo no distorsione el peso de las características durante el entrenamiento. La librería *Scikit-Learn* dispone de la función *MinMaxScaler* dentro del módulo *preprocessing* que internamente aplica la ecuación 4.1.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

donde x representa la característica a transformar, es decir, la hora de salida, la visibilidad y el trimestre del año que corresponda.

Selección de inputs categóricos

En el apartado 3.1.2 se definen los diferentes datos de tipo categóricos que forman parte del *dataset* original. Este tipo de datos requieren un tratamiento especial, ya que necesariamente deben convertirse a tipo numérico. El motivo radica en que los modelos de *Machine Learning* no dejan de ser funciones matemáticas que requieren valores numéricos para poder ser entrenados.

Para este problema, se han seleccionado como variables categóricas descritos en 3.1.1:

- Aeropuertos de origen (JFK, EWR y LGA).
- Aerolíneas.

A este tipo de datos, se le aplicará una codificación binaria, conocido en la librería *Scikit-Learn* como *One-Hot-Encoder*. En la Figura 4.15 se muestra el procedimiento que emplea la librería para llevar a cabo esta transformación de datos.

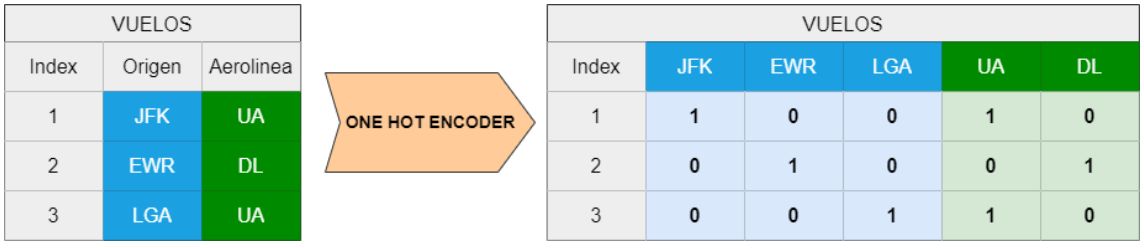


Figura 4.15: Procesado de datos categóricos mediante *One Hot Encoder*.

En primer lugar debe realizar un *cast* [1] de las variables a un tipo de datos específico, en este caso es el tipo *category* empleado por la librería *pandas*. La función *OneHotEncoder* del

[1] Denominado en el ámbito de la programación al procedimiento de convertir las variables con un tipo de datos a otro. Por ejemplo, de una variable de tipo *int* a *float*.

módulo *preprocessing* de la librería *Scikit-Learn* transforma los datos tal como se describe en la Figura 4.15. En este ejemplo se habrían codificado los 3 aeropertos de origen y 2 aerolíneas.

Debido al procedimiento empleado por *OneHotEncoder*, el número de variables se multiplica según el número de categorías por característica. En el caso de los aeropuertos de origen, se obtendrían ahora 3 características o variables (una por cada aeropuerto). De forma análoga, se obtendrían hasta 16 nuevas características debido a las aerolíneas. En este último caso, incrementar demasiado el número de variables se traduce en una disminución considerable en la resolución del problema, por lo que el entrenamiento del modelo se vería afectado. Para resolver esta casuística, se simplificarán el número de aerolíneas a las más operadas (según el gráfico 4.16), dejando fuera del modelo aquellas que presentan pocas operaciones y distorsionarian el modelo final.

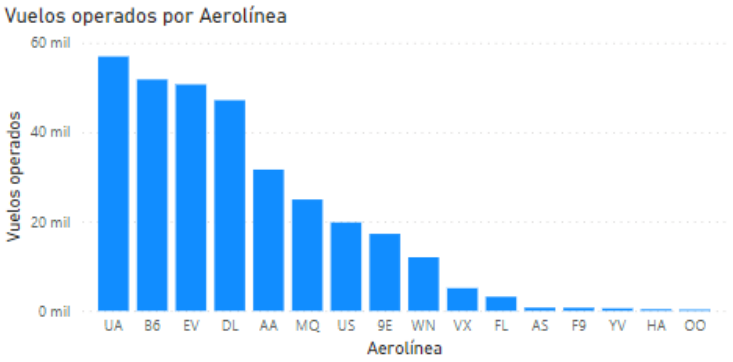


Figura 4.16: Número de vuelos operados por aerolínea.

Selección del output

La variable de salida del modelo es binaria, es decir, la etiqueta de los datos (*tag*) será 0 si los vuelos no han sufrido retraso y 1 en caso contrario. Para ello se han etiquetado los datos comparando la hora programada del vuelo frente a la hora real de salida (ver Figura 4.17).

4.2.2. Inputs etiquetados del modelo

El *input* final quedaría como un *dataframe* de 300.131 registros debido a que únicamente se están considerando las aerolíneas con mayor operación. La Tabla 4.1 se corresponde a una muestra de 10 líneas aleatorias con las variables que se consideraran como *input* y los *tags* (*dep delayed*) antes de codificar las variables.

Vuelos			
Index	Hora salida programada	Hora salida real	Tag
1	18:45	19:12	1
2	20:30	20:25	0

Figura 4.17: Proceso de etiquetado de los datos.

Cuadro 4.1: Variables input antes de la transformación y etiquetas.

Hour	vsby	quarter	origin	carrier	dep_delayed
12	10.0	1	LGA	MQ	0.0
8	10.0	4	EWR	EV	0.0
7	8.0	2	EWR	EV	0.0
16	10.0	2	JFK	AA	1.0
20	8.0	1	EWR	UA	1.0
7	10.0	1	LGA	DL	0.0
6	6.0	3	JFK	B6	1.0
14	10.0	4	LGA	US	0.0
10	10.0	4	LGA	9E	0.0
10	10.0	2	LGA	DL	0.0

Tras la transformación de la variables númericas (*Min Max Scaler*) y categóricas (*One Hot Encoder*), la entrada del modelo y las etiquetas quedarían tal y como se muestra en la Tabla 4.2.

Cuadro 4.2: Variables input después de la transformación y etiquetas.

Hour	vsby	quarter	EWR	JFK	LGA	9E	AA	B6	DL	EV	MQ	UA	US	dep_delayed
0.388889	1.0	0.000000	0	0	1	0	0	0	0	0	1	0	0	0
0.166667	1.0	1.000000	1	0	0	0	0	0	0	1	0	0	0	0
0.111111	0.8	0.333333	1	0	0	0	0	0	0	1	0	0	0	0
0.611111	1.0	0.333333	0	1	0	0	1	0	0	0	0	0	0	1
0.833333	0.8	0.000000	1	0	0	0	0	0	0	0	0	1	0	1
0.111111	1.0	0.000000	0	0	1	0	0	0	1	0	0	0	0	0
0.055556	0.6	0.666667	0	1	0	0	0	1	0	0	0	0	0	1
0.500000	1.0	1.000000	0	0	1	0	0	0	0	0	0	0	1	0
0.277778	1.0	1.000000	0	0	1	1	0	0	0	0	0	0	0	0
0.277778	1.0	0.333333	0	0	1	0	0	0	1	0	0	0	0	0

Datos de entrenamiento y validación

Llevar a cabo el entrenamiento de un modelo de *Machine Learning* requiere de separar previamente el *dataset* inicial en dos partes diferenciadas (proceso definido en 3.3):

- **Datos de entrenamiento:** un 80 % del conjunto total de datos etiquetados. Se entrenará al modelo empleando este subconjunto de forma que irá aprendiendo en base a las etiquetas, puesto que se trata de un aprendizaje supervisado.
- **Datos de validación:** un 20 % del conjunto total de datos. Una vez entrenado el modelo, se validará con este subconjunto a partir de unas métricas definidas.

La librería *Scikit-Learn* proporciona la función *train test split* del módulo *model selection* que permite separar el *dataset* en datos de entrenamiento y validación de forma aleatoria en las proporciones que se definan.

4.2.3. Regresión logística

En 3.3.1 se ha desarrollado una explicación teórica de los fundamentos de la regresión logística. Actualmente, herramientas como la librería *Scikit-Learn* (ver sección 2.3), proporcionan APIs [2] sencillas para llevar el entrenamiento de modelos de *Machine Learning*.

La regresión logística, como ya se definió en 3.3.1, es un modelo de aprendizaje supervisado para llevar a cabo problemas de clasificación binaria. En la librería *Scikit-Learn* de *Python*, el modelo de regresión logístico se encuentra implementado en la clase ***LogisticRegression***.

```
1 from sklearn.linear_model import LogisticRegression
2 from sklearn.preprocessing import MinMaxScaler
3 from sklearn.preprocessing import OneHotEncoder
4 from sklearn.model_selection import train_test_split
```

En las líneas de código superiores muestran como se importan los módulos correspondientes de la librería *Scikit-Learn* para llevar a cabo el preprocesado de datos explicado en 4.2.1 y el entrenamiento del modelo. La definición del modelo se realizaría con la siguiente declaración de la variable *model*:

[2] es el acrónimo en inglés de *interfaz de programación de aplicaciones*, se trata de un software intermediario que permite la comunicación entre aplicaciones.

```
model = LogisticRegression(solver='liblinear')
```

Para problemas de clasificación binaria, el *solver* específico empleado en regresión logística se denomina *liblinear* en *Scikit-Learn*. Se denomina *solver* a un algoritmo numérico que trata de encontrar los coeficientes óptimos en un modelo matemático. El *solver 'liblinear'* esta orientado a problemas de optimización de pequeña y mediana escala, lo que le convierte en una buena opción para los conjuntos de datos que no son muy grandes y la clasificación es binaria. Cuando creas una instancia del objeto ***LogisticRegression*** en scikit-learn, puedes especificar el *solver* que deseas utilizar mediante el parámetro ***solver***. Por defecto, *liblinear* se utiliza para problemas de clasificación binaria.

Es importante mencionar que si no se especifica el *solver*, *Scikit-Learn* elegirá automáticamente el *solver* más adecuado según las características de los datos y la configuración del modelo. Para problemas multiclase, *Scikit-Learn* utiliza automáticamente el *solver lbfgs* por defecto. Puedes ajustar este comportamiento utilizando el parámetro *multi class* junto con el parámetro *solver* si es necesario.

El entrenamiento del modelo se realiza mediante la siguiente instrucción:

```
model.fit(X_train, y_train)
```

donde *X train* se corresponden a las variables de entrada e *y train* a las etiquetas correspondientes, como se muestra en la Tabla 4.2.

Para medir la precisión del modelo, se emplea el método *score* de la siguiente forma:

```
model.score(X_test, y_test))
```

donde *X test* e *y test* se corresponden con el 20 % de los datos. La precisión o *accuracy* se define como el porcentaje de predicciones acertadas frente al número total de instancias. En este problema se ha obtenido un **64.37 % de *accuracy***. Es decir, el modelo es capaz de acertar si un vuelo va a sufrir retraso o no con un acierto del 64 % aproximadamente en base a las variables de *input* definidas en 4.2.1.

4.2.4. Árbol de decisión

En 3.3.1 se han definido los conceptos básicos de un árbol de decisión. La librería *Scikit-Learn* presenta una clase denominada ***DecisionTreeClassifier*** que implementa árboles de decisión dedicados a problemas de clasificación.

```
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.metrics import accuracy_score
```

Además de la clase para instanciar el modelo, se han importado otras clases como **export graphviz** para visualizar el árbol de nodos y **accuracy score** del módulo *metrics* para medir el *accuracy* del modelo entrenado.

```
tree = DecisionTreeClassifier(max_depth=15)
tree.fit(X_train, y_train)
```

Esta clase permite configurar varios parámetros que afectan la construcción del árbol, como la profundidad máxima del árbol, el número mínimo de muestras requeridas para dividir un nodo, la función de criterio utilizada para medir la calidad de una división, entre otros. Estos parámetros se pueden ajustar según las características del problema y los datos. En este caso, se ha decidido emplear una profundidad máxima de 15. Con el método *fit* se entrena el modelo a partir de los datos de entrenamiento. Igual que en la regresión logística se mide la precisión del modelo.

```
prediction = tree.predict(X_test)
accuracy_score(prediction,y_test)
```

Tras testear el modelo con los datos de test, se ha obtenido un *accuracy* del **65,92 %**.

Capítulo 5

Conclusiones

Este capítulo expone un análisis de los objetivos alcanzados y los resultados obtenidos. Se valoran cuáles han sido los principales factores limitantes y qué desarrollos se podrían realizar para mejorar los resultados, siguiendo esta línea de trabajo.

5.1. Trabajo realizado

En este Trabajo de Fin de Grado se ha mostrado el procedimiento para analizar los datos relativos a la operación de vuelos durante 1 año desde 3 aeropuertos de origen. Se ha creado una herramienta capaz de representarlos en un *dashboard* de visualización de datos, mostrando la información de forma que el usuario sea capaz de sacar conclusiones de forma eficiente y directa.

Por otro lado, se han utilizado los datos para llevar a cabo el entrenamiento de 2 modelos de clasificación basados en aprendizaje supervisado. Para ello, se ha marcado como objetivo la capacidad de clasificar un vuelo en función de si va a sufrir o no retraso en la salida, tomando como *input* variables relevantes de la operación como la hora de salida, estacionalidad del año, visibilidad, aeropuerto de origen y aerolíneas con mayor operación.

5.2. Análisis de objetivos alcanzados

En el capítulo 3 se ha puesto en contexto los datos que se van a utilizar en el proyecto y un esquema general del trabajo a realizar. En la sección 4.1 se detallan las diferentes secciones

(*tabs*) que se han creado para representar los datos empleando la herramienta *PowerBI*. Con esto quedaría alcanzado el **Objetivo 1** definido en la Tabla 1.1.

En la sección 3.3 se han desarrollado los fundamentos teóricos de los modelos basados en aprendizaje supervisado para problemas de clasificación. En 4.2 se ha explicado el procedimiento de preprocesado de aquellas variables con las que el modelo de *Machine Learning* será entrenado y cómo se ha empleado la librería *Scikit-Learn* para instaciar y entrenar los modelos en lenguaje *Python*. Con esto quedaría alcanzado el **Objetivo 2** de este proyecto definido en la Tabla 1.1.

5.3. Análisis de resultados obtenidos

En la sección 4.1 se han ido exponiendo de forma detallada los distintos diseños (*layouts*) que conforman el *dashboard* de visualización, además se aporta una breve explicación de aquellos casos donde los datos mostraban un determinado comportamiento o patrón.

En la sección 4.2.3 y 4.2.4 se muestra la implementación de los modelos de clasificación **regresión logística** y **árbol de decisión** respectivamente, así como el resultado final obtenido en términos de *accuracy* (ver Tabla 5.1).

Cuadro 5.1: *Accuracy* de los modelos entrenados.

Modelo de clasificación	Accuracy
Regresión logística	64,37 %
Árbol de decisión	65,92 %

A la vista de los resultados obtenidos, se puede afirmar que el árbol de decisión proporciona un mejor resultado en términos de *accuracy*. Algunas razones por las que el árbol de decisión muestra mejores resultados de aprendizaje pueden ser:

- Los árboles de decisión son capaces de capturar mejor las relaciones no lineales entre las características y la salida, puesto que la regresiñon logística asume una relación de linealidad entre las variables.

- Capturan automáticamente las conexiones entre las características, como la combinación de características para tomar las decisiones.
- Mayor robustez frente a datos ruidosos, ya que presentan la capacidad de ignorar los datos irrelevantes durante la construcciñon del árbol.
- Fácil de interpretar, como un conjunto de reglas de decisión sencillas.

5.4. Limitaciones

Entre las principales limitaciones encontradas en este proyecto se encuentran:

- Licencias para herramientas. *PowerBI* es una herramienta de pago, por lo que si se necesitan más funcionalides como más objetos visuales o permisos de compartición de informes es necesario de disponer la respectiva licencia.

5.5. Trabajos futuros

Siguiendo la línea de trabajo de este proyecto se plantean las siguientes mejoras o trabajos a futuro de interés:

- **Aplicación de un problema de regresión.** Este proyecto se ha centrado en un modelo de clasificación binaria. Se propone como alternativa un problema de regresión cuyo objetivo es estimar el retraso de los vuelos en minutos en función de unas variables de entrada. El modelo puede considerar, por ejemplo, técnicas de ajuste flexibles que capturen mejor relaciones complejas entre las variables de entrada y el retraso de los vuelos.
- **Implementación de un modelo de aprendizaje no supervisado.** La librería *Scikit-Learn* ofrece módulos dedicados al aprendizaje no supervisado, por ejemplo, *k-means* es el más conocido.
- **Modelos basados en *Deep Learning***. A partir de implementación de redes neuronales multicapa es posible llevar a cabo diversos problemas de aprendizaje supervisado y no supervisado. Una librería dedicada al desarrollo de redes neuronales es *Keras* [10], cuya API resulta sencilla en comparación a sus alternativas, basada en el lenguaje *Python*.

- **Como alternativa a *PowerBI* se propone utilizar *Dash*** [8], una librería de *Python* dedicada a la creación de objetos de datos visuales (gráficos y tablas) utilizando este lenguaje y *Pandas* para el tratamiento de datos.

Apéndice A

Aeropuertos y ciudades de destino

Cuadro A.1: Nombre y ciudades de los aeropuertos de destino.

Código aeropuerto	Nombre aeropuerto	Ciudad destino
ABQ	Albuquerque International Sunport	Albuquerque
ACK	Nantucket Memorial Airport	Nantucket
ALB	Albany International Airport	Albany
ANC	Ted Stevens Anchorage International Airport	Anchorage
ATL	Hartsfield Jackson Atlanta International Airport	Atlanta
AUS	Austin Bergstrom International Airport	Austin
AVL	Asheville Regional Airport	Asheville
BDL	Bradley International Airport	Hartford
BGR	Bangor International Airport	Bangor
BHM	Birmingham-Shuttlesworth International Airport	Birmingham
BNA	Nashville International Airport	Nashville
BOS	General Edward Lawrence Logan International Airport	Boston
BQN	Rafael Hernandez Airport	Aguadilla
BTV	Burlington International Airport	Burlington
BUF	Buffalo Niagara International Airport	Buffalo

Cont. en pág. sig.

Cuadro A.1: Nombre y ciudades de los aeropuertos de destino.

Código aeropuerto	Nombre aeropuerto	Ciudad destino
BUR	Bob Hope Airport	Burbank
BWI	Baltimore/Washington International Thurgood Mar...	Baltimore
BZN	Gallatin Field	Bozeman
CAE	Columbia Metropolitan Airport	Columbia
CAK	Akron Canton Regional Airport	Akron
CHO	Charlottesville Albemarle Airport	Charlottesville
CHS	Charleston Air Force Base-International Airport	Charleston
CLE	Cleveland Hopkins International Airport	Cleveland
CLT	Charlotte Douglas International Airport	Charlotte
CMH	John Glenn Columbus International Airport	Columbus
CRW	Yeager Airport	Charleston
CVG	Cincinnati Northern Kentucky International Airport	Cincinnati / Covington
DAY	James M Cox Dayton International Airport	Dayton
DCA	Ronald Reagan Washington National Airport	Washington
DEN	Denver International Airport	Denver
DFW	Dallas Fort Worth International Airport	Dallas-Fort Worth
DSM	Des Moines International Airport	Des Moines
DTW	Detroit Metropolitan Wayne County Airport	Detroit
EGE	Eagle County Regional Airport	Eagle
EYW	Key West International Airport	Key West
FLL	Fort Lauderdale Hollywood International Airport	Fort Lauderdale
GRR	Gerald R. Ford International Airport	Grand Rapids
GSO	Piedmont Triad International Airport	Greensboro
GSP	Greenville Spartanburg International Airport	Greenville

Cont. en pág. sig.

Cuadro A.1: Nombre y ciudades de los aeropuertos de destino.

Código aeropuerto	Nombre aeropuerto	Ciudad destino
HDN	Yampa Valley Airport	Hayden
HNL	Daniel K Inouye International Airport	Honolulu
HOU	William P Hobby Airport	Houston
IAD	Washington Dulles International Airport	Washington
IAH	George Bush Intercontinental Houston Airport	Houston
ILM	Wilmington International Airport	Wilmington
IND	Indianapolis International Airport	Indianapolis
JAC	Jackson Hole Airport	Jackson
JAX	Jacksonville International Airport	Jacksonville
LAS	McCarran International Airport	Las Vegas
LAX	Los Angeles International Airport	Los Angeles
LEX	Blue Grass Airport	Lexington
LGB	Long Beach /Daugherty Field/ Airport	Long Beach
MCI	Kansas City International Airport	Kansas City
MCO	Orlando International Airport	Orlando
MDW	Chicago Midway International Airport	Chicago
MEM	Memphis International Airport	Memphis
MHT	Manchester-Boston Regional Airport	Manchester
MIA	Miami International Airport	Miami
MKE	General Mitchell International Airport	Milwaukee
MSN	Dane County Regional Truax Field	Madison
MSP	Minneapolis-St Paul International/Wold-Chamberl...	Minneapolis
MSY	Louis Armstrong New Orleans International Airport	New Orleans
MTJ	Montrose Regional Airport	Montrose

Cont. en pág. sig.

Cuadro A.1: Nombre y ciudades de los aeropuertos de destino.

Código aeropuerto	Nombre aeropuerto	Ciudad destino
MVY	Martha's Vineyard Airport	Martha's Vineyard
MYR	Myrtle Beach International Airport	Myrtle Beach
OAK	Metropolitan Oakland International Airport	Oakland
OKC	Will Rogers World Airport	Oklahoma City
OMA	Eppley Airfield	Omaha
ORD	Chicago O'Hare International Airport	Chicago
ORF	Norfolk International Airport	Norfolk
PBI	Palm Beach International Airport	West Palm Beach
PDX	Portland International Airport	Portland
PHL	Philadelphia International Airport	Philadelphia
PHX	Phoenix Sky Harbor International Airport	Phoenix
PIT	Pittsburgh International Airport	Pittsburgh
PSE	Mercedita Airport	Ponce
PSP	Palm Springs International Airport	Palm Springs
PVD	Theodore Francis Green State Airport	Providence
PWM	Portland International Jetport	Portland
RDU	Raleigh Durham International Airport	Raleigh/Durham
RIC	Richmond International Airport	Richmond
ROC	Greater Rochester International Airport	Rochester
RSW	Southwest Florida International Airport	Fort Myers
SAN	San Diego International Airport	San Diego
SAT	San Antonio International Airport	San Antonio
SAV	Savannah Hilton Head International Airport	Savannah
SBN	South Bend Regional Airport	South Bend
SDF	Louisville Muhammad Ali International Airport	Louisville
SEA	Seattle Tacoma International Airport	Seattle

Cont. en pág. sig.

Cuadro A.1: Nombre y ciudades de los aeropuertos de destino.

Código aeropuerto	Nombre aeropuerto	Ciudad destino
SFO	San Francisco International Airport	San Francisco
SJC	Norman Y. Mineta San Jose International Airport	San Jose
SJU	Luis Munoz Marin International Airport	San Juan
SLC	Salt Lake City International Airport	Salt Lake City
SMF	Sacramento International Airport	Sacramento
SNA	John Wayne Airport-Orange County Airport	Santa Ana
SRQ	Sarasota Bradenton International Airport	Sarasota/Bradenton
STL	St Louis Lambert International Airport	St Louis
STT	Cyril E. King Airport	Charlotte Amalie, Harry S. Truman Airport
SYR	Syracuse Hancock International Airport	Syracuse
TPA	Tampa International Airport	Tampa
TUL	Tulsa International Airport	Tulsa
TVC	Cherry Capital Airport	Traverse City
TYS	McGhee Tyson Airport	Knoxville
XNA	Northwest Arkansas Regional Airport	Fayetteville/Springdale/ Rogers

Bibliografía

[1] Cloud coverage. https://www.eoas.ubc.ca/courses/atsc113/flying/met_concepts/01-met_concepts/01c-cloud_coverage/index.html.

[2] Metar and taf weather reports. http://www.moratech.com/aviation/metaf-abbrev.html.

[3] Visibility - skybrary. https://skybrary.aero/articles/visibility.

[4] Airports codes, 2020. https://datahub.io/core/airport-codes.

[5] PPO. Intuitive guide to state-of-the-art Reinforcement Learning, 2022. https://medium.com/mlearning-ai/ppo-intuitive-guide-to-state-of-the-art-reinforcement-learning-410a41cb675b.

[6] Aprendizaje Automático Supervisado, 2023. https://masterdatascience.online/aprendizaje-automatico-supervisado/.

[7] Aprendizaje supervisado vs no supervisado en 3 minutos, 2023. https://manualestutor.com/aprendizaje-automatico/aprendizaje-supervisado-vs-no-supervisado-en-3-minutos/.

[8] Dash python user guide, 2023. https://dash.plotly.com/.

[9] Jupyter, 2023. https://jupyter.org/.

[10] Keras: Deep learning for humans, 2023. https://keras.io/.

[11] Python. software foundation, 2023. https://www.python.org/community/logos/.

[12] Power bi - microsoft logo png vector, 2024. https://seeklogo.com/vector-logo/400711/power-bi-microsoft.

[13] Sckit learn. machine learning in python, 2024. https://scikit-learn.org/stable/.

[14] J. Bobadilla. *Machine learning y deep learning: usando Python, Scikit y Keras*. Ediciones de la U, 2021.

[15] J. García, J. Molina, A. Berlanga, M. Patricio, A. Bustamante, and W. Padilla. Ciencia de datos. *Técnicas Analíticas y Aprendizaje Estadístico. Bogotá, Colombia*, 2018.

[16] R. González Duque. Python para todos, 2011.

[17] A. MOHAMMAD. Airlines codes, 2022. https://www.kaggle.com/datasets/arbazmohammad/world-airports-and-airlines-datasets/.

[18] B. Power, U. Excel, P. Desktop, and P. Tiles. Microsoft power bi. *Available here: https://powerbi. microsoft. com/en-us*, 130, 2021.

[19] B. M. Randles, I. V. Pasquetto, M. S. Golshan, and C. L. Borgman. Using the jupyter notebook as a tool for open science: An empirical study. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–2. IEEE, 2017.

[20] A. H. Sedaghati. Jfk, ewr, and lga flights, 2023. https://www.kaggle.com/datasets/amirhoseinsedaghati/jfk-ewr-and-lga-flights/data.

[21] F. O. R. Suguiura. Árbol de decisión en aprendizaje automático. *Revista Varianza*, pages 39–46, 2022.