



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

“CASA SMART LANDING”

Predicción del momento de aterrizaje de los aviones
en EEUU mediante métodos de machine learning

Autor: Amaia Rotaeché Montero

Director: Leonard Frank Register – UT Austin

Co-Director: Sam Soheili - Boeing

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

“Casa Smart Landing”

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2023/24 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Amaia Rotaache Montero

Fecha: 03/06/2024



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Leonard Frank Register

Fecha: 02/ 07/2024





GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

“CASA SMART LANDING”

Predicción del momento de aterrizaje de los aviones
en EEUU mediante métodos de machine learning

Autor: Amaia Rotaeché Montero

Director: Leonard Frank Register – UT Austin

Co-Director: Sam Soheili - Boeing

Madrid

“CASA SMART LANDING” – Predicción del momento de aterrizaje de los aviones en EEUU mediante métodos de machine learning

Autor: Rotaeché Montero, Amaia

Director: Leonard Frank Register

Entidad Colaboradora: Boeing

RESUMEN DEL PROYECTO

Frente a un alarmante crecimiento del número y tiempo de los retrasos en salidas y llegadas de vuelos en Estados Unidos, este proyecto tiene como objetivo optimizar la gestión de los recursos en los principales aeropuertos del país. Para ello, se han recopilado y formateado datos de distintas fuentes que afectan directa o indirectamente al sector aéreo. Se pretende entrenar y validar, mediante técnicas de machine learning, un modelo capaz de predecir, con máxima exactitud, el tiempo de llegada de un avión a su aeropuerto destino. Para interactuar con el modelo de predicción en tiempo real, se ha desarrollado un interfaz visualizador GUI que permite al usuario observar la hora de llegada prevista de los aviones que sobrevuelan su localización en ese instante.

Palabras clave: modelo de predicción, hora de aterrizaje, vuelos, receptor ADS-B, interfaz GUI

1. Introducción y definición del proyecto

La demanda de viajes aéreos eficientes está creciendo a nivel mundial, impulsada por el aumento tanto del comercio internacional como de los viajes personales. Este aumento del tráfico aéreo supone una presión considerable para los sistemas existentes de gestión del tráfico aéreo (ATM), lo que requiere avances para mejorar la eficiencia operativa y la seguridad del sector. El objetivo principal de este proyecto es desarrollar un modelo de aprendizaje automático, o machine learning, capaz de predecir con precisión los tiempos de llegada de los aviones a los principales aeropuertos de Estados Unidos.

Para ello, se ha recurrido a datos históricos y registros en tiempo real de factores intrínsecos de cada vuelo, así como atributos externos impredecibles referentes a las condiciones meteorológicas. Para la obtención de registros históricos se ha accedido a fuentes públicas como “OpenSky” para los datos de aeronaves y la Administración Nacional Oceánica y Atmosférica (NOAA) para aquellos climatológicos. Para la validación y visualización en tiempo real del modelo, se ha montado un receptor ADS-B, tecnología usada en todos los

sistemas de telecomunicaciones para la transmisión de señales entre los aviones y los puestos de control en tierra.

Existen numerosos programas de investigación ya realizados con el mismo objetivo, pero este diseño añade y se centra en aquellas características más aleatorias e impredecibles como, por ejemplo, las condiciones temporales en los alrededores del aeropuerto de destino, así como los itinerarios seguidos por el piloto para aterrizar, que afectan significativamente al tiempo de llegada del avión. Con todo ello, el proyecto tiene como objetivo optimizar el uso de las capacidades de las pistas y minimizar los retrasos.

La predicción precisa de los tiempos de aterrizaje de las aeronaves puede mejorar significativamente la flexibilidad de programación y la eficiencia de los aeropuertos. Este proyecto implica recopilar datos de fuentes autorizadas en aviación y meteorología, aplicar técnicas avanzadas de procesamiento de datos y emplear algoritmos de aprendizaje automático para predecir los tiempos de llegada. La metodología integra TensorFlow para el desarrollo y la validación de modelos, utilizando tecnología ADS-B en vivo para probar y refinar las predicciones. Este enfoque innovador no sólo aborda las necesidades inmediatas de mejorar la eficiencia ATM sino que también contribuye a implicaciones más amplias para la seguridad y el impacto ambiental. Al optimizar la predicción de los tiempos de aterrizaje de los aviones, el proyecto apoya una mejor gestión del consumo de combustible y reduce la congestión innecesaria del tráfico aéreo, lo que conduce a una disminución de las emisiones de carbono. El plan se alinea con los objetivos de desarrollo sostenible como una propuesta innovativa para la industria y una acción por el clima.

En definitiva, el proyecto representa un paso crucial hacia la integración de tecnologías de aprendizaje automático en el campo de la gestión del tráfico aéreo, proporcionando una solución escalable que podría implementarse en diferentes aeropuertos a nivel mundial.

2. Descripción del sistema

Como se muestra en la figura 0, el Sistema se divide en 3 actividades focales:

1. Búsqueda, extracción y procesamiento de datos
2. Entrenamiento de los datos y validación del modelo de machine learning mediante Tensor Flow
3. Creación de la interfaz GUI

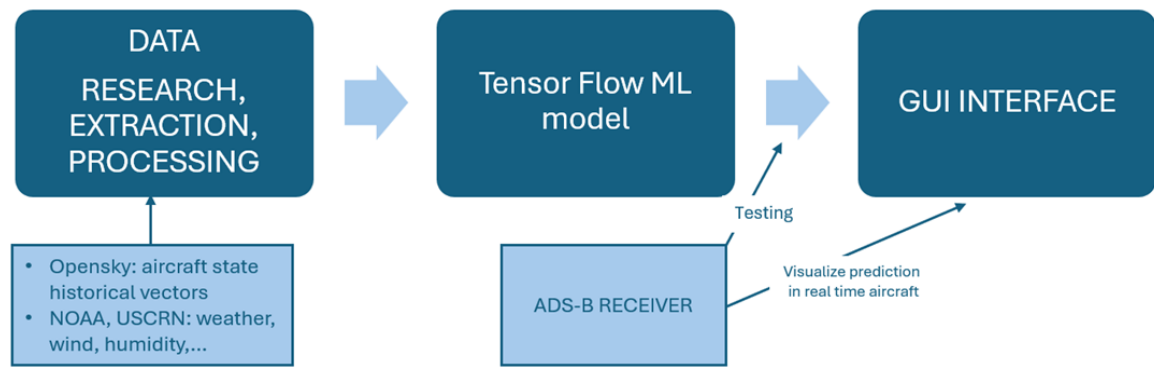


Figure 0: Esquema de la división y sucesión de actividades del proyecto

1- Búsqueda, extracción y procesamiento de datos:

Los datos sobre los que se ha trabajado provienen de dos fuentes: “OpenSky Network” y “NOAA”. “OpenSky Network” es una red de receptores ADS-B que ha estado recopilando datos de vigilancia del tráfico aéreo desde 2013 y los mantiene abiertos para investigadores académicos. Cada segundo, los sensores de la red obtienen una nueva señal de cada avión a su alcance que incluye información relevante como el identificador del avión, registro temporal, su posición (latitud, longitud y altitud), velocidad, etc. De esta fuente, se extrajeron todos los registros históricos guardados en 2022 de los aviones que sobrevuelan el territorio americano. Por otro lado, NOAA, la Administración nacional Oceánica y Atmosférica en Estados Unidos proporciona pronósticos meteorológicos diarios y monitorización del clima en océanos y territorio estadounidense. De esta fuente se extrajeron datos climáticos que afectan al sector aéreo como por ejemplo temperatura, precipitaciones, velocidad del viento, etc. Se sacaron las variables oportunas de cada tabla, añadieron nuevas variables calculadas necesarias para resolver la problemática y se juntaron todos los datos en una “mastertabla” de millones de entradas.

2- Modelo de Machine Learning con Tensor Flow

Ya obtenida la “mastertabla”, se pasaba al modelo de aprendizaje automático (Machine Learning) como conjunto de entrenamiento. Como requisito de la empresa Boieng, se trata de un modelo de regresión lineal, con posibilidad de mejora y desarrollo de nuevos modelos de regresión en el futuro. La codificación del mismo, se hizo en un entorno de programación Python a través de la librería Tensor Flow. Se trata de una biblioteca de software de código abierto desarrollada por Google para el aprendizaje automático y la inteligencia artificial. Fue diseñada para facilitar la creación y el entrenamiento de modelos de aprendizaje profundo (deep learning) y otros algoritmos de machine learning. Dado que el proyecto

involucraba a numerosas personas entre alumnos, supervisores académicos y de la empresa Boeing, la decisión de usar esta librería fue principalmente por su escalabilidad (permite desplegarse en múltiples dispositivos) además de su facilidad de integración con otras herramientas de machine learning. Una vez entrenado, y antes de pasar a un producto presentable para el cliente, se realizó la evaluación del rendimiento del modelo. Se necesita verificar la capacidad de generalización a datos no incluidos previamente en el entrenamiento. Para obtener este conjunto de datos de test, se utilizó la tecnología ADS-b previamente mencionada. Se montó un receptor ADS-B con antena incorporada capaz de obtener las señales enviadas por los aviones en vuelo detectables en el rango determinado por el fabricante. Con esto, se tomaron registros suficientes (30% de entradas test y 70% de entrenamiento) en distintos días de recepción. Se aplicó el modelo a dichos registros y se compararon los valores de la variable “target” reales y predichos. Para comparar, como métrica de comprobación de precisión se utilizó el error cuadrático medio MSE (Mean Squared Error).

3- Interfaz GUI para el usuario

El último paso es convertir el proyecto en un producto tangible y presentable a un cliente potencial. Para ello, se ha codificado una interfaz gráfica de usuario (GUI) que permite a los usuarios comunicarse con el software implementado. Se trata de una página web cuyo frontend está codificado en lenguaje html e incluye diversos elementos gráficos como mapas, iconos, celdas de entrada, etc. Esto permite a los usuarios observar los aviones en vuelo, sus características e identificar su hora estimada de llegada. El backend de esta aplicación hace llamadas a la API donde se almacenan los datos registrados en “tiempo real” por el receptor ADS-B así como el modelo de predicción. De esta forma, la aplicación obtiene de forma instantánea la hora estimada de llegada de los aviones que están sobrevolando la localización en ese instante.

3. Resultados

Tras el desarrollo de todas las actividades mencionadas, los resultados óptimos obtenidos fueron los siguientes:

- Un modelo de predicción de regresión lineal con un error cuadrático medio que ronda los 10 minutos. Así como una apreciación de aquellas variables más influyentes en el tiempo de llegada de una aeronave a destino.
- Una interfaz GUI capaz de predecir y mostrar la hora de llegada de los aviones que sobrevuelan la ciudad de Austin.

4. Conclusiones

Este proyecto demuestra el potencial de integrar tecnologías avanzadas de machine learning en la gestión del tráfico aéreo para optimizar los recursos aeroportuarios y minimizar los retrasos en los vuelos. A través de la recopilación y procesamiento de datos históricos y en tiempo real de fuentes como OpenSky Network y NOAA, se ha desarrollado un modelo de predicción de tiempos de llegada con una precisión notable. La implementación de una interfaz gráfica de usuario (GUI) facilita la interacción con el modelo en tiempo real, permitiendo a los usuarios visualizar las horas estimadas de llegada de los aviones que sobrevuelan una localización específica, como la ciudad de Austin. En resumen, este proyecto representa un avance significativo en la integración de machine learning en la gestión del tráfico aéreo, ofreciendo una solución escalable y eficiente que puede ser implementada en aeropuertos de todo el mundo para mejorar la precisión en la predicción de los tiempos de aterrizaje y optimizar la gestión de los recursos aeroportuarios.

A continuación, se explicará con detalle cada subsistema del trabajo, desglosando el proceso de búsqueda y procesamiento de datos, el desarrollo y validación del modelo de machine learning, y la creación de la interfaz GUI, así como sus respectivas contribuciones a los objetivos del proyecto.

5. Referencias

- [1] “The OpenSky Network - Free ADS-B and Mode S data for Research”. The OpenSky Network - Free ADS-B and Mode S data for Research. Accedido el 2 de julio de 2024. [En línea]. Disponible: <https://opensky-network.org/> NOAA, “National Centers for Environmental Information (NCEI),” Noaa.gov, Jun. 06, 2019. <https://www.ncei.noaa.gov/>
- [2] N. C. for E. I. NCEI), “Quality Controlled Datasets,” www.ncei.noaa.gov. <https://www.ncei.noaa.gov/access/crn/qcdatasets.html>

- [3] “Introducción a TensorFlow”. TensorFlow. Accedido el 2 de julio de 2024. [En línea]. Disponible: <https://www.tensorflow.org/learn?hl=es-419>
- [4] “Get A Receiver,” opensky-network.org, <https://opensky-network.org/contribute/get-a-receiver> (accessed Dec. 05, 2023).

CASA SMART LANDING” – Landing time prediction using Machine learning techniques

Author: Rotaeché Montero, Amaia.

Supervisor: Register, Leonard.

Collaborating Entity: Boeing Company

ABSTRACT

In response to the alarming increase in the number and duration of flight delays for departures and arrivals in the United States, this project aims to optimize resource management at the country's major airports. To achieve this, data from various sources that directly or indirectly affect the aviation sector has been collected and formatted. This data is used to train and validate a machine learning model capable of accurately predicting the arrival time of an aircraft at its destination airport. To interact with the prediction model in real-time, a GUI visualizer interface has been developed, allowing users to observe the estimated arrival time of airplanes flying over their current location.

Keywords: prediction model, arrival time, flight, ADS-B receiver, GUI interface

1. Introduction and Project definition

The demand for efficient air travel is growing globally, driven by the increase in both international trade and personal travel. This rise in air traffic puts considerable pressure on existing air traffic management (ATM) systems, necessitating advancements to improve operational efficiency and sector safety. The primary objective of this project is to develop a machine learning model capable of accurately predicting the arrival times of aircraft at major airports in the United States.

To achieve this, historical data and real-time records of intrinsic factors for each flight, as well as unpredictable external attributes related to weather conditions, have been utilized. Historical records have been obtained from public sources such as OpenSky for aircraft data and the National Oceanic and Atmospheric Administration (NOAA) for weather data. For model validation and real-time visualization, an ADS-B receiver has been installed. This technology is used in all telecommunication systems for signal transmission between aircraft and ground control stations.

Numerous research programs with the same objective already exist, but this design adds and focuses primarily on more random and unpredictable characteristics, such as weather

conditions around the destination airport and the flight paths followed by pilots to land, which significantly affect the aircraft's arrival time. Overall, the project aims to optimize runway capacity usage and minimize delays.

Accurate prediction of aircraft landing times can significantly improve scheduling flexibility and airport efficiency. This project involves collecting data from authorized aviation and meteorology sources, applying advanced data processing techniques, and employing machine learning algorithms to predict arrival times. The methodology integrates TensorFlow for model development and validation, using live ADS-B technology to test and refine predictions. This innovative approach not only addresses the immediate needs of improving ATM efficiency but also contributes to broader implications for safety and environmental impact. By optimizing the prediction of aircraft landing times, the project supports better fuel consumption management and reduces unnecessary air traffic congestion, leading to lower carbon emissions. The plan aligns with sustainable development goals as an innovative industry proposal and a climate action initiative.

In summary, this project represents a crucial step toward integrating machine learning technologies in the field of air traffic management, providing a scalable solution that could be implemented in various airports worldwide.

2. System description

As shown in Figure 0, the system is divided into three main activities:

- 1- Data research, extraction, and processing
- 2- Tensor Flow Machine learning model
- 3- Creation of the GUI interface

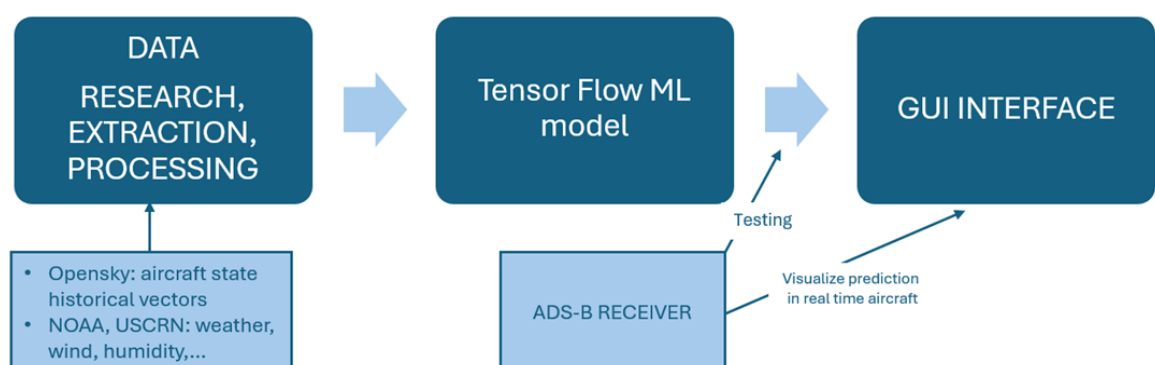


Figure 0: Diagram of the division and sequence of project activities

1- Data Research, Extraction, and Processing

The data used in this project comes from two sources: OpenSky Network and NOAA. OpenSky Network is a network of ADS-B receivers that has been collecting air traffic surveillance data since 2013 and keeps it open for academic researchers. Every second, the network's sensors receive a new signal from each aircraft within their range, including relevant information such as the aircraft identifier, timestamp, position (latitude, longitude, and altitude), speed, etc. Historical records from 2022 of aircraft flying over American territory were extracted from this source. On the other hand, NOAA, the National Oceanic and Atmospheric Administration in the United States, provides daily weather forecasts and climate monitoring over the oceans and U.S. territory. From this source, weather data affecting the aviation sector, such as temperature, precipitation, wind speed, etc., was extracted. Appropriate variables were taken from each table, new necessary variables were calculated, and all data were combined into a "master table" with millions of entries.

2- TensorFlow Machine Learning Model

Once the "master table" was obtained, it was fed into the machine learning model as a training set. As required by Boeing, it is a linear regression model, with the possibility of improvement and development of new regression models in the future. The coding was done in a Python programming environment using the TensorFlow library. TensorFlow is an open-source software library developed by Google for machine learning and artificial intelligence. It was designed to facilitate the creation and training of deep learning models and other machine learning algorithms. Since the project involved numerous participants, including students, academic supervisors, and Boeing staff, the decision to use this library was primarily due to its scalability (it can be deployed on multiple devices) and its ease of integration with other machine learning tools. Once trained, and before presenting a product to the client, the model's performance was evaluated. It was necessary to verify its ability to generalize to data not previously included in the training. To obtain this test data set, the previously mentioned ADS-B technology was used. An ADS-B receiver with an integrated antenna was installed, capable of receiving signals sent by in-flight aircraft detectable within the manufacturer's specified range. With this setup, sufficient records were taken (30% test entries and 70% training) on different reception days. The model was applied to these

records, and the actual and predicted values of the target variable were compared. The mean squared error (MSE) was used as the accuracy verification metric.

3- GUI Interface

The final step is to turn the project into a tangible product presentable to a potential client. To this end, a graphical user interface (GUI) has been coded, allowing users to interact with the implemented software. It is a webpage whose frontend is coded in HTML and includes various graphic elements such as maps, icons, input cells, etc. This allows users to observe in-flight aircraft, their characteristics, and identify their estimated arrival time. The backend of this application makes calls to the API where real-time data recorded by the ADS-B receiver, as well as the prediction model, are stored. Thus, the application instantly obtains the estimated arrival time of aircraft flying over the location at that moment.

3. Results

After completing all the mentioned activities, the optimal results obtained were as follows:

- A linear regression prediction model with a mean squared error of just over two minutes. Additionally, an understanding of the most influential variables affecting the arrival time of an aircraft at its destination.
- A GUI interface capable of predicting and displaying the arrival time of airplanes flying over the city of Austin.

4. Conclusions

This project demonstrates the potential of integrating advanced machine learning technologies into air traffic management to optimize airport resources and minimize flight delays. By collecting and processing historical and real-time data from sources such as OpenSky Network and NOAA, a remarkably accurate arrival time prediction model has been developed. The implementation of a graphical user interface (GUI) facilitates real-time interaction with the model, allowing users to visualize the estimated arrival times of airplanes flying over a specific location, such as the city of Austin. In summary, this project represents a significant advancement in the integration of machine learning into air traffic management, offering a scalable and efficient solution that can be implemented in airports

worldwide to improve landing time prediction accuracy and optimize airport resource management.

Next, each subsystem of the project will be explained in detail, breaking down the process of data collection and processing, the development and validation of the machine learning model, and the creation of the GUI interface, as well as their respective contributions to the project's objectives.

5. References

- [1] “The OpenSky Network - Free ADS-B and Mode S data for Research”. The OpenSky Network - Free ADS-B and Mode S data for Research. Accedido el 2 de julio de 2024. [En línea]. Disponible: <https://opensky-network.org> NOAA, “National Centers for Environmental Information (NCEI),” Noaa.gov, Jun. 06, 2019. <https://www.ncei.noaa.gov/>
- [2] N. C. for E. I. NCEI), “Quality Controlled Datasets,” www.ncei.noaa.gov. <https://www.ncei.noaa.gov/access/crn/qcdatasets.html>
- [3] “Introducción a TensorFlow”. TensorFlow. Accedido el 2 de julio de 2024. [En línea]. Disponible: <https://www.tensorflow.org/learn?hl=es-419>
- [4] “Get A Receiver,” opensky-network.org. <https://opensky-network.org/contribute/get-a-receiver> (accessed Dec. 05, 2023).

Índice de la memoria

Capítulo 1. Introducción	6
1.1 Motivación del proyecto.....	6
Capítulo 2. Descripción de las Tecnologías.....	9
2.1 Python como lenguaje de programación	9
2.2 Receptor ADS-B.....	9
2.3 Servicio AWS en la nube – s3 bucket	13
Capítulo 3. Estado de la Cuestión	14
3.1 Agrupación de trayectorias de los flujos de tráfico aéreo alrededor de los aeropuertos - por Xavier Olive y Jérôme Morio, 2018, Université Fédérale de Toulouse [6].....	15
3.2 Análisis de datos ADS-B de rtl sdr para la predicción de aeropuertos y rutas ats – por Shravan, Rakshit, et al. (2020, Conferencia INCET) [7]	17
3.3 Uso de datos de código abierto para la predicción del tiempo de aterrizaje con métodos de Aprendizaje Automático– por Chen gong, et al (2020, OpenSky SYmposium) [8].....	20
Capítulo 4. Definición del Trabajo	25
4.1 Justificación.....	25
4.2 Objetivos	26
4.2.1 Objetivos generales	26
4.2.2 Objetivos tangibles marcados por la empresa	27
4.3 Metodología.....	28
4.4 Planificación y Estimación Económica	30
4.4.1 Coste Económico	31
4.4.2 Planificación temporal	32
Capítulo 5. Sistema/Modelo Desarrollado.....	35
5.1 Análisis del Sistema	35
5.2 Diseño.....	35
5.3 Implementación	37
5.3.1 División, diseño y montaje de subsistemas.....	38
5.3.2 Integración de subsistemas.....	51
5.3.3 Pruebas de evaluación	53

Capítulo 6. <i>Análisis de Resultados</i>	55
6.1 Resultados parciales	55
6.1.1 Obtención de un dataset correcto y manejable	55
6.1.2 Integración de una plataforma eficiente en la nube	56
6.1.3 Obtención de una interfaz GUI	56
6.2 Resultados generales	56
6.2.1 Rendimiento del modelo predictivo - Análisis de la Importancia de las Variables	56
6.2.2 Rendimiento del modelo predictivo – Error cuadrático medio	57
Capítulo 7. <i>Conclusiones y Trabajos Futuros</i>	60
7.1 Conclusiones	60
7.2 Trabajos futuros	62
7.2.1 Ampliación del estudio a nuevos aeropuertos destino	62
7.2.2 Establecimiento de una red de receptores y antenas	63
Capítulo 8. <i>Bibliografía</i>	66
ANEXO I: <i>ALINEACIÓN DEL PROYECTO CON LOS ODS</i>	68
ANEXO II: <i>TABLA DE ASIGNACIÓN DE TAREAS EN EL GRUPO</i>	71
ANEXO III: <i>DIAGRAMA GANTT DE PLANIFICACIÓN TEMPORAL</i>	73

Índice de figuras

Figura 1: Esquema de funcionamiento de la tecnología ADS-B [4]	10
Figura 2: Foto de los componentes del receptor ADS-B.....	12
Figura 3:Diagrama de la agrupación de trayectorias resultante [6].....	16
Figura 4:Diagrama de las posibles trayectorias determinadas por un modelo de árbol de decisión [6]	16
Figura 5: Diagrama de bloques del receptor [7]	18
Figura 6: Captura del portal web de Dump 1090 [7].....	19
Figura 7: Configuración del receptor ADS-B [7].....	20
Figura 8: La diferencia en el ángulo de trayectoria entre el punto de entrada en el círculo y el punto de aterrizaje. (Izquierda) Trayectorias con menos tiempo de aterrizaje; (Derecha) trayectorias con más tiempo de aterrizaje [8].....	21
Figura 9: Histograma de predictores de aterrizaje [8]	22
Figura 10: Esquema de la división y sucesión de actividades del proyecto	37
Figura 11: Tabla de los atributos	39
Figura 12: Mapa de estaciones USCRN en todo el territorio nacional	41
Figura 13: Esquema del join entre las tablas de registros aéreos y meteorológicos.....	43
Figura 14: Ejemplo de registros en la tabla final "mastertabla"	43
Figura 15: Diagrama del flujo de datos desde la detección por la antena hasta el almacenamiento en la nube.....	47
Figura 16: Captura de los registros observados en la herramienta de visualización en el ordenador	47
Figura 17: Diagrama del flujo de datos con la interfaz GUI	48
Figura 18: Vista de la interfaz por el usuario	50
Figura 19: Arquitectura del sistema global y flujo de datos integrado.....	52
Figura 20: Gráficas del comportamiento de variables en los datos ADS-B recibidos	54
Figura 21: Gráfica de barras del peso de las variables más importantes en el modelo	57

Figura 22: Salida del RMSE para los distintos conjuntos de test.....	58
--	----

Índice de tablas

Tabla 1: Tabla de componentes del receptor ADS-B.....	12
Tabla 2: Asignación de precios a componentes del receptor ADS-B.....	32
Tabla 3: Asignación de estaciones a cada aeropuerto	41

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN DEL PROYECTO

En la actualidad, en un mundo donde la globalización lleva varios años inmersa en la vida de todos, la movilización de personas y mercancías entre todos los puntos del mundo está a la orden del día. El objetivo es llegar lo más lejos en el menor tiempo posible y la solución es el avión. Frente a una falta de competidores para el propósito mencionado, la demanda del sector aéreo sufre un crecimiento exponencial sin precedentes que no parece que se vaya a estabilizar en el corto plazo. Pero aumentar la oferta de productos en el sector para afrontar esta creciente demanda, no es fácil. Ampliar la oferta de aviones, aeropuertos y otros servicios relacionados, es un desafío debido a varios factores complejos y multifacéticos. En primer lugar, los costes de fabricación y mantenimiento de aviones y aeropuertos son altísimos. Requieren de una inversión enorme en tecnologías innovadoras, infraestructura, seguridad, ... Por otro lado, se trata de un sector muy regulado. Por la seguridad de los pasajeros, las certificaciones de calidad de toda la tecnología usada y las crecientes restricciones ambientales, la aviación se ve muy atada a la hora de crear y sacar productos viables al mercado. Y, por último, el espacio para infraestructura está limitado. En las áreas urbanas, donde la necesidad es evidente, no hay suficiente espacio para expandir aeropuertos existentes o construir nuevos. EN definitiva, ante el reto que supone la demanda creciente del sector, hacemos frente a una falta de oferta y unos recursos limitados. Por todo ello, resulta muy necesaria la optimización de los recursos ya disponibles en el sector. Aumentar la eficiencia es clave. Y es aquí donde la predicción más exacta de la hora de llegada de los aviones se vuelve crucial.

Además, los retrasos de vuelos comerciales en Estados Unidos es un problema muy grave que acarrea unas consecuencias económicas enormes a todas las empresas involucradas. En los últimos años, se ha observado un crecimiento tanto en el número como en la duración de los retrasos de vuelos en Estados Unidos. Según un informe de la Oficina de Estadísticas del

Transporte de Estados Unidos (Bureau of Transportation Statistics, BTS) [13], el porcentaje de vuelos retrasados ha aumentado gradualmente. En 2017, aproximadamente el 20% de los vuelos experimentaron retrasos, mientras que en 2022 esta cifra superó el 25%. La duración promedio de los retrasos también ha mostrado un incremento significativo. En 2017, la duración promedio de un retraso era de aproximadamente 50 minutos. Para 2022, esta cifra había aumentado a casi 70 minutos [14].

Está claro que estos retrasos causan inconvenientes significativos para todos los actores. Los clientes sufren pérdidas de conexiones, estrés, cambios en planes de viajes. Las aerolíneas deben hacer frente a costos adicionales debido al consumo extra de combustible, gastos de reorganización y compensación a pasajeros. Los aeropuertos pueden enfrentar congestión adicional y la necesidad de gestionar la logística de vuelos retrasados, lo cual puede afectar la eficiencia operativa y los ingresos. Y como consecuencia global, los retrasos de vuelos tienen un impacto económico significativo, afectando la productividad y el comercio. Un estudio de la Universidad de California en Berkeley [15] estimó que los retrasos en vuelos comerciales cuestan a la economía de Estados Unidos más de 30 mil millones de dólares anuales.

Este incremento en la duración y número de los retrasos puede atribuirse a diversos factores como la congestión en los aeropuertos en fechas más destacadas, problemas operacionales como escasez de personal, fallos técnicos o retrasos en cadena. Sin embargo, una de las mayores causas de retrasos son las condiciones meteorológicas adversas: zonas de tormentas, niebla o fuertes vientos que tienen lugar en algún punto del itinerario. Como consecuencia de estas, los pilotos se ven obligados a tomar decisiones más personales llevando a demoras cuyo tiempo resulta impredecible para los controladores del tráfico aéreo. Además, otro motivo de demora con respecto al tiempo de viaje preestablecido, y que también está muy relacionado con las decisiones propias de los pilotos, es la trayectoria de aterrizaje aplicada. Cuando los aviones se acercan a las inmediaciones del aeropuerto destino, normalmente tienen varias opciones de recorrido a seguir antes de posarse en la pista de aterrizaje. En función de su propia voluntad, dirección del viento, nivel de congestión de la pista, etc, los pilotos entran al radio que delimita la zona de aterrizaje con un ángulo de

entrada que varía. Y esta ruta seguida antes de aterrizar puede ser más o menos larga y por tanto causar tiempos de espera que, como dependen de decisiones personales más inmediatas, son imprevisibles. Por ello, el proyecto de predicción llevado a cabo busca centrarse en estas dos características menos consideradas en trabajos de investigación anteriores.

Por todo ello, saber la hora a la que un avión aterriza en destino supone un aprovechamiento óptimo de los recursos que poseen las aerolíneas, aeropuertos y empresas afectadas por el problema y contribuye a una mejora de todos los procesos de la cadena de producción del sector. Para las aerolíneas, además de suponer una mejora del servicio al cliente por la puntualidad, la gestión de conexiones y la buena reputación que obtienen por ello, saber la hora de aterrizaje de los aviones optimiza sus recursos y reduce sus costes. Pueden gestionar sus flotas de manera más eficiente, los horarios de su tripulación, reducir el consumo de combustible, etc. En el caso de los aeropuertos, esto les permite una planificación adelantada, gestión y programación y coordinación de todos sus recursos: pistas de aterrizaje, puertas de embarque, personal de asistencia en tierra, controladores aéreos. Y además esto afecta directamente a sus ingresos económicos. Un aeropuerto que opera de manera eficiente puede manejar un mayor volumen de vuelos y pasajeros, lo que se traduce en mayores ingresos por tasas aeroportuarias y servicios.

En resumen, conocer la hora de llegada y anteponerse a posibles retrasos de vuelos es fundamental para mejorar la eficiencia operativa, reducir costos, aumentar la seguridad y mejorar la satisfacción del cliente tanto para las aerolíneas como para los aeropuertos. Esto, a su vez, contribuye a una operación más competitiva y rentable en un sector altamente dinámico y exigente.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

2.1 PYTHON COMO LENGUAJE DE PROGRAMACIÓN

El proyecto realizado tiene componentes de software y hardware. Todas las etapas seguidas en la planificación (figura 0) incluyen metodología software: diseño de la “mastertabla”, entrenamiento y test del modelo, backend y frontend de la interfaz de visualización GUI. Por su popularidad, sus múltiples librerías orientadas al machine learning y por su facilidad de integración con otras tecnologías como interfaces APIs, Python fue el lenguaje elegido en la codificación de estas actividades mencionadas. Cabe destacar que se trata de un proyecto, propiedad de Boeing por lo que el código implementado y las tecnologías específicas para resolver el problema son confidenciales y no aparecerán en esta memoria.

2.2 RECEPTOR ADS-B

Por otro lado, el proyecto incluye un importante componente de hardware altamente influyente en la resolución de la problemática, así como en el producto final visible por el cliente. Se trata del diseño y montaje del receptor ADS-B capaz de captar registros de datos de aviones en vuelo utilizados para el test del modelo y para la visualización en tiempo real de las aeronaves y sus respectivas horas de llegada previstas por el modelo, palpable en la interfaz GUI.

Un receptor ADS-B (Automatic Dependent Surveillance-Broadcast) es un dispositivo utilizado para recibir y decodificar señales transmitidas por aviones equipados con transpondedores ADS-B. Estos transpondedores emiten información sobre la posición, altitud, velocidad y otros datos del vuelo, que pueden ser recibidos y utilizados por

controladores aéreos, pilotos y entusiastas de la aviación para monitorear el tráfico aéreo en tiempo real. En mi caso, la tecnología se usará para la investigación académica.

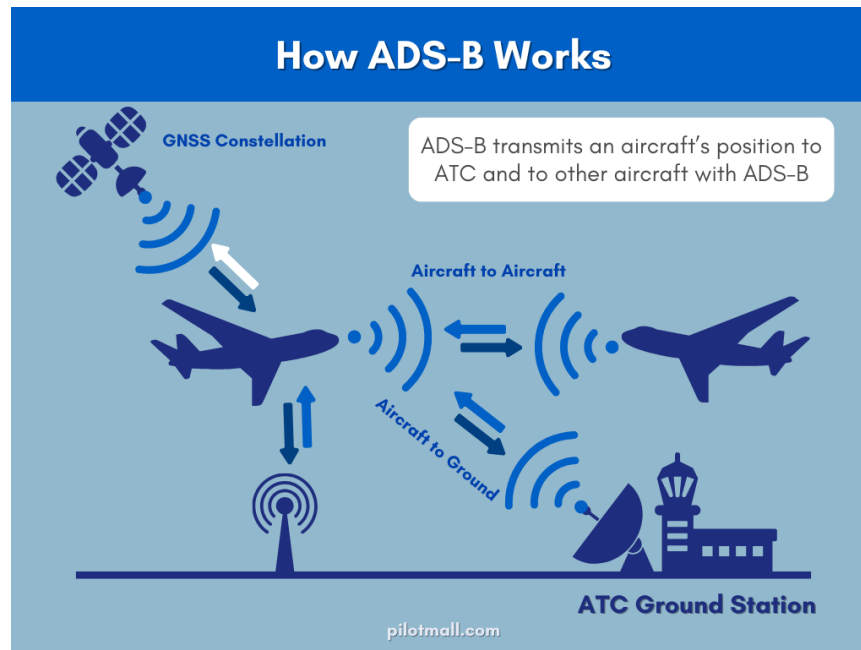


Figura 1: Esquema de funcionamiento de la tecnología ADS-B [4]

Como se muestra en la figura 1, el sistema ADS-B funciona mediante la transmisión de señales desde el avión a receptores en tierra o en otros aviones. La señal transmitida contiene datos esenciales del vuelo que son calculados por el avión usando sus sistemas de navegación (como GPS). Estos datos incluyen:

- **Identificación del avión:** Número de vuelo o matrícula.
- **Posición:** Latitud y longitud calculadas por el GPS del avión.
- **Altitud:** Altitud barométrica o GPS.
- **Velocidad:** Velocidad del aire y velocidad sobre el suelo.
- **Dirección:** Rumbo y curso del avión.
- **Información adicional:** Estado del vuelo, como si está subiendo, bajando o nivelado.

Los datos son transmitidos en frecuencias específicas (1090 MHz para aviones comerciales y de aviación general, y 978 MHz en los EE. UU. para algunos vuelos a baja altitud).

Estos receptores se diseñaron y utilizan para diferentes propósitos sobre todo relacionados con la seguridad del sector y el control del tráfico aéreo (ATC). Los controladores aéreos usan datos ADS-B para mejorar la vigilancia y gestión del tráfico aéreo, especialmente en áreas donde el radar tradicional no es efectivo. Además, esta tecnología ayuda a los pilotos a mejorar su conciencia situacional, Mejora la conciencia situacional de los pilotos, permitiéndoles ver la posición de otros aviones equipados con ADS-B en sus pantallas.

Los componentes necesarios para mostrar esta tecnología se incluyen en la tabla 1, y referenciados con números en la figura 2.

COMPONENTE	DESCRIPCION
<i>Antena ADS-B (5)</i>	<i>Puede ser una antena colineal o una antena dipolo. Debe estar sintonizada a 1090 MHz (o 978 MHz en algunos casos). Debe colocarse en un lugar alto y despejado para recibir las señales sin obstrucciones.</i>
<i>Receptor SDR (Software-Defined Radio) (4)</i>	<i>consiste en un dispositivo USB o una tarjeta específica que permite recibir señales de radio y luego procesarlas digitalmente a través de software. Este software realiza las funciones de demodulación, decodificación y procesamiento de las señales ADS-B.</i>

Ordenador / microcontrolador (1)	Herramientas dotadas con Programas como dump1090 (para decodificar las señales ADS-B) y PiAware (para compartir datos con FlightAware) son comunes.
Fuente de alimentación	Para alimentar al receptor y ordenador
Conexiones y cables (7)	Cables coaxiales: De baja pérdida para conectar la antena al receptor SDR. Adaptadores: Para conectar el SDR a la computadora o microcontrolador.

Tabla 1: Tabla de componentes del receptor ADS-B

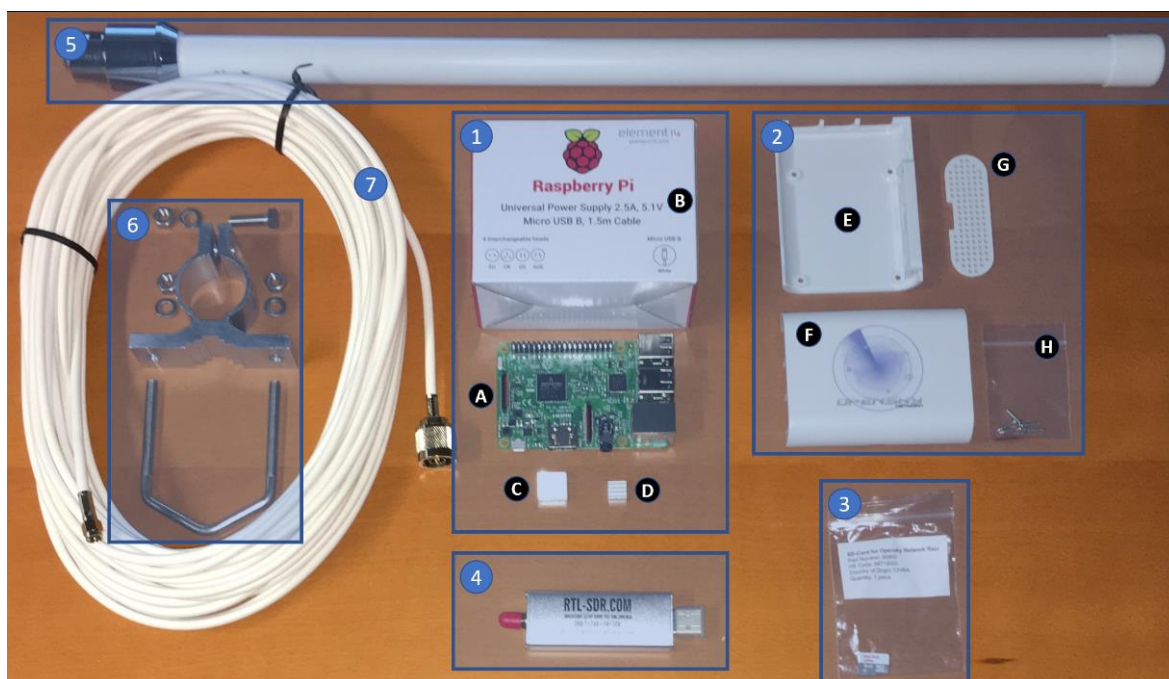


Figura 2: Foto de los componentes del receptor ADS-B

2.3 SERVICIO AWS EN LA NUBE – S3 BUCKET

Por otro lado, al tratarse de un proyecto de software que maneja una gran cantidad de datos, grandes tablas y archivos, necesita de una capacidad de almacenaje y procesamiento muy alta. Por ello en lugar de gestionar todos los procesos en local, se determinó apoyarse en servicios de la nube y particularmente en la tecnología desarrollada por Amazon: “Amazon Web Services o AWS. Se trata de una plataforma líder en servicios de computación en la nube que ofrece a individuos, empresas y organizaciones una variedad extensa de servicios y herramientas para ejecutar aplicaciones y almacenar datos en la nube. Permite a sus usuarios liberar y descongestionar su propia infraestructura física al trasladar sus recursos informáticos a la nube.

AWS proporciona una amplia gama de servicios en la nube que incluyen computación, almacenamiento, bases de datos, redes, análisis, inteligencia artificial, Internet de las cosas (IoT), seguridad entre otros muchos. En definitiva, todos estos servicios nos han permitido agilizar nuestros procesos sin necesidad de invertir en hardware costoso.

Dentro de los diferentes servicios ofrecidos por esta tecnología, se ha usado el Amazon S3 (Simple Storage Service) Bucket. esencialmente es un servicio de almacenamiento en la nube diseñado para almacenar y recuperar datos a través de internet. Se utiliza para almacenar una amplia variedad de datos, incluidos archivos, imágenes, videos, bases de datos, copias de seguridad y más, en forma de objetos.

Un S3 Bucket es un contenedor fundamental dentro del servicio Amazon S3, donde se almacenan los objetos. Cada objeto dentro de un bucket tiene una clave única que lo identifica. Para acceder, almacenar y extraer los objetos o datos dentro del S3 bucket, AWS proporciona distintas alternativas incluyendo la consola web de AWS, SDKs (Software Development Kits) para varios lenguajes de programación, y comandos de línea de comandos (CLI) para operaciones de push y pull de datos.

Capítulo 3. ESTADO DE LA CUESTIÓN

En el campo de la aviación, se han llevado a cabo numerosos estudios para determinar la hora de llegada de un avión con la mayor precisión posible. Especialmente en las últimas décadas, en las que la eficiencia del sector se ha visto amenazada por la alta demanda. Se han desarrollado numerosos modelos probabilísticos y deterministas. La mayoría de ellos relacionan la llegada de una aeronave con numerosos parámetros físicos del propio vuelo y su trayectoria. Sin embargo, estos modelos presentan algunos problemas de precisión, ya que se basan en condiciones "ideales" sin tener en cuenta limitaciones actuales y la intervención del factor humano, que resultan ser los principales factores en el retraso de un vuelo. Por ejemplo, una tormenta imprevista causará una desviación en la ruta basada en el criterio del pensamiento humano, y este tiempo extra no fue contemplado en los estudios.

Por otro lado, en los últimos años y gracias a la influencia del machine learning y la inteligencia artificial, se ha avanzado en el campo de la predicción del tiempo de vuelo. Estos nuevos modelos han permitido mejorar los problemas probabilísticos anteriores relacionados con la incertidumbre de los factores influyentes en el objetivo.

Es necesario un estudio exhaustivo de los trabajos previos realizados para resolver problemas relacionados con nuestro objetivo y la base sobre la cual apoyarnos y tomar ejemplo para guiar el desarrollo de nuestro proyecto. A continuación, se incluyen ejemplos de artículos y patentes, previamente desarrollados, enfocados en encontrar una solución parcial o completa a mi problema. Haré un breve estudio de cada caso, y relacionaremos y compararé sus soluciones con mis necesidades y requisitos.

3.1 AGRUPACIÓN DE TRAYECTORIAS DE LOS FLUJOS DE TRÁFICO AÉREO ALREDEDOR DE LOS AEROPUERTOS - POR XAVIER OLIVE Y JÉRÔME MORIO, 2018, UNIVERSITÉ FÉDÉRALE DE TOULOUSE [6]

Basado en discusiones con expertos y el mentor de la compañía Boeing, además de factores como el clima, el viento, la altitud o la congestión en el propio aeropuerto de llegada, otros factores que parecen tener poca relación resultan afectar significativamente el tiempo de llegada de un avión. Tanto la trayectoria seguida, el área de entrada en el círculo concéntrico (cuyo centro es el aeropuerto de llegada), como el ángulo de incidencia en dicho círculo son puntos clave para determinar el tiempo de llegada. Dependiendo de la trayectoria que sigue y, por lo tanto, del área por la que entra al aeropuerto, así como del ángulo de entrada a este, el avión se verá obligado a recorrer más o menos distancia antes de aterrizar, tomando más o menos tiempo en aterrizar. En resumen, y después de hablar con el mentor de la compañía, es necesario crear agrupaciones que difieran en el comportamiento de entrada (área de entrada, ángulo de incidencia, ...) de cada aeronave al círculo de radio determinado cuyo centro es el aeropuerto de llegada.

En 2018, Xavier Olive y Jérôme Morio presentaron un nuevo enfoque para separar las trayectorias del tráfico aéreo en un área restringida por procedimientos operativos alrededor del aeropuerto de Toulouse. Las agrupaciones resultantes fomentan una buena comprensión de la estructura del tráfico y de cómo los controladores programan los aterrizajes en el aeropuerto de Toulouse-Blagnac.

Su estudio se basa en un registro de 3 semanas de un ADS-B capaz de enviar la posición en coordenadas de cada avión y así poder definir su trayectoria uniendo cada punto registrado. Simplificaron los datos eligiendo solo aquellos cuyo aeropuerto de aterrizaje era Toulouse y utilizaron algoritmos de simplificación de trayectorias para que la mayoría consistieran solo de 5/7 puntos.

Las trayectorias resultantes se muestran en la figura 3.



Figura 3: Diagrama de la agrupación de trayectorias resultante [6]

El esquema de los datos recogidos sobre las trayectorias muestra claramente patrones comunes. Hay áreas completamente rojas con alto flujo de tráfico aéreo y áreas que son prácticamente inutilizables para volar sobre el aeropuerto de Toulouse. Para acceder a la pista de un aeropuerto, los aviones entran a través de áreas específicas que afectan la ruta y el tiempo de llegada al destino.

También crearon árboles de decisión para determinar y predecir trayectorias posibles. El nodo raíz es el aeropuerto y todos los caminos hacia los nodos hoja describen un flujo potencial de aeronaves siguiendo los procedimientos de aterrizaje (figura 4).

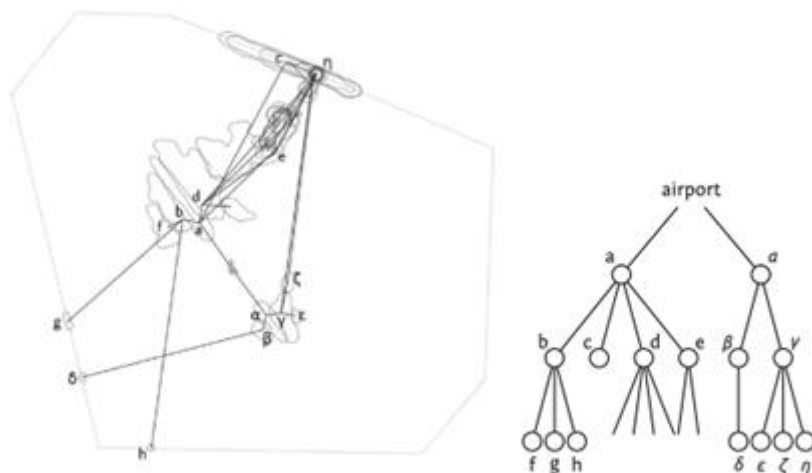


Figura 4: Diagrama de las posibles trayectorias determinadas por un modelo de árbol de decisión [6]

En resumen, la trayectoria que sigue cada avión no es ni incierta ni aleatoria, sino que está predeterminada y cada avión puede identificarse y pertenecer a un grupo según la trayectoria que sigue antes de aterrizar. Y esto, aunque no representa una solución directa a mi problema respecto al tiempo de aterrizaje, está claramente conectado. Crear agrupaciones y asociar patrones comunes en la trayectoria de llegada de un avión es clave para simplificar el problema. Esta agrupación de aviones basada en su comportamiento al llegar al área del aeropuerto no nos dirá directamente su tiempo de aterrizaje, pero simplificará en gran medida el modelo. Con este estudio se puede agrupar cada avión basado en características comunes en el trayecto y aterrizaje y así poder estimar un tiempo de llegada basado en su pertenencia al grupo. Por ejemplo, en el caso del aeropuerto de Toulouse en la figura 4, aquellos aviones que entran por el punto de entrada g al área que rodea el aeropuerto recorrerán menos distancia hasta la pista y, por lo tanto, tomarán menos tiempo para llegar a su destino. Sin embargo, aquellos que entran por el punto alfa tendrán que tomar un ángulo de incidencia diferente y recorrer más distancia hasta la pista de aterrizaje, lo que tomará más tiempo. Y esto ya representa un claro avance hacia mi objetivo.

3.2 ANÁLISIS DE DATOS ADS-B DE RTL SDR PARA LA PREDICCIÓN DE AEROPUERTOS Y RUTAS ATS – POR SHRAVAN, RAKSHIT, ET AL. (2020, CONFERENCIA INCET) [7]

Este estudio de arte previo se centra en utilizar sistemas ADS-B para mejorar el seguimiento de aeronaves. El equipo en este informe de investigación empleó una radio RTL-SDR, específicamente, el R820T RTL2832U en combinación con una computadora de placa única Raspberry Pi-3B para capturar y analizar datos ADS-B de aeronaves que viajan en Bangalore, India. La configuración requerida, como se documenta, se utilizó para determinar las trayectorias de varias aeronaves con quince rutas únicas del Servicio de Tráfico Aéreo y para determinar los aeropuertos y direcciones de pista más probables.

Los componentes utilizados en esta investigación fueron la Raspberry Pi 3 Modelo B, la Radio y Antena RTL-SDR, y por último el software de código abierto: Dump1090. Durante la discusión de la metodología, los autores destacaron la versatilidad del dongle RTL-SDR y su compatibilidad con software como MATLAB, GNU Radio y Simulink, que facilitaron el diseño de sistemas basados en SDR. Durante el estudio, los autores notaron la eficiencia del dongle RTL-SDR en la monitorización del movimiento de aeronaves y en proporcionar pantallas de seguimiento de vuelos actualizadas. Otra área que se examinó fueron las consideraciones éticas, tales como preocupaciones de seguridad con sugerencias como mejorar la precisión de los datos GPS y asegurar la protección de datos. La figura 5 muestra el diagrama de bloques del sistema completo utilizado durante su investigación.

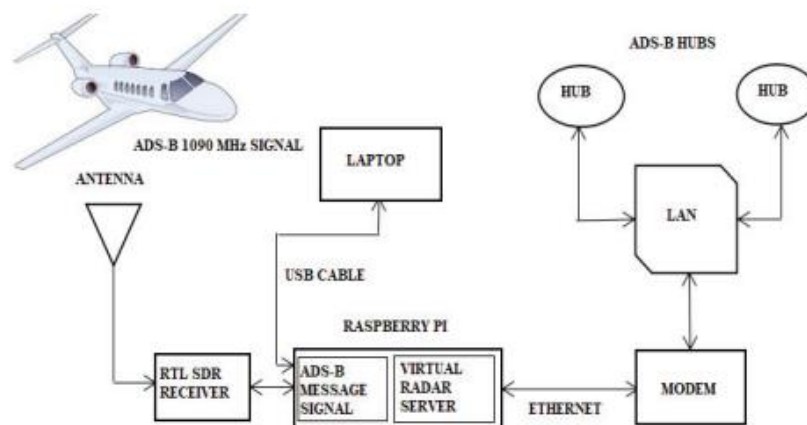


Figura 5: Diagrama de bloques del receptor [7]

El dongle RTL-SDR específico que se utilizó aquí cubre un rango de frecuencias desde 24 MHz hasta 1850 MHz, que cubre la señal ADS-B de 1090 MHz. Dado que el receptor puede detectar estas frecuencias y sus datos de aeronaves recopilados, pudieron determinar la posición de aterrizaje y de pista. A partir de sus datos de aeronaves recopilados, el rango de altitudes fue de 1000 pies a 45000 pies, sin embargo, se asignó un rango de altitudes de 1000 pies a 3000 pies para detectar cada aeropuerto que esté a punto de aterrizar o despegar de un aeropuerto. Dump1090 es un software compatible con RTL-SDR que permite la decodificación ADS-B. Una vez que Dump1090 está en funcionamiento, la Raspberry Pi actúa como un procesador de datos ADS-B. Dump1090 también sobresale en la

decodificación de señales débiles. Puede decodificar mensajes DF11 y DF17. Estos dos mensajes diferentes se emiten a una tasa nominal de 1 hertz por transpondedores Mode S que se encuentran en el receptor. Estos dos mensajes se emiten periódicamente, lo que se refiere a transmisiones automáticas periódicas de los datos de una aeronave. La emisión corta se refiere a la transmisión de información básica sobre la aeronave. Las emisiones extendidas, como los mensajes DF17, también se utilizan para el sistema ADS-B, pero contienen información adicional que no se encuentra en los mensajes de emisión corta, como la posición y velocidad de la aeronave. La figura 6 muestra cómo Dump1090 obtiene datos ADS-B del receptor y los muestra en la aplicación.

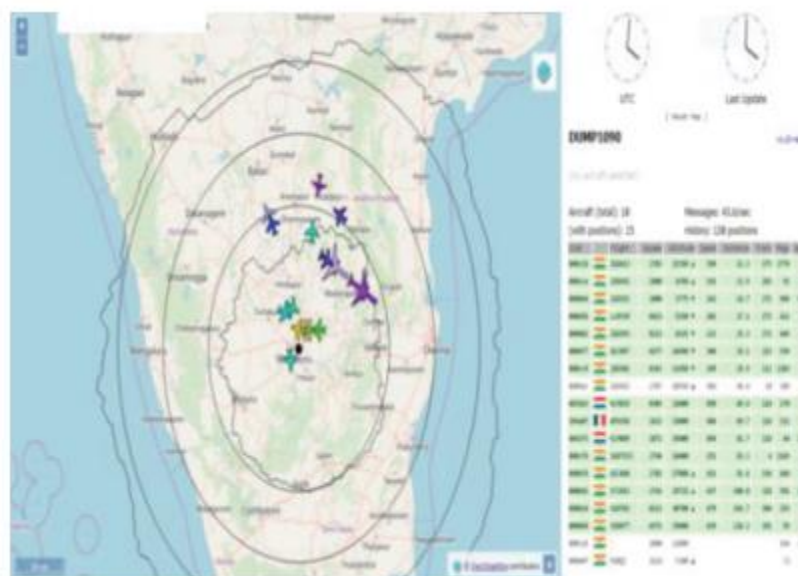


Figura 6: Captura del portal web de Dump 1090 [7]

La figura 7 muestra la configuración del receptor para su investigación que es para obtener datos de vuelo durante cuatro días en Bangalore, India. La figura muestra la ubicación de la antena, el RTL-SDR y la Raspberry Pi y cómo están montados.

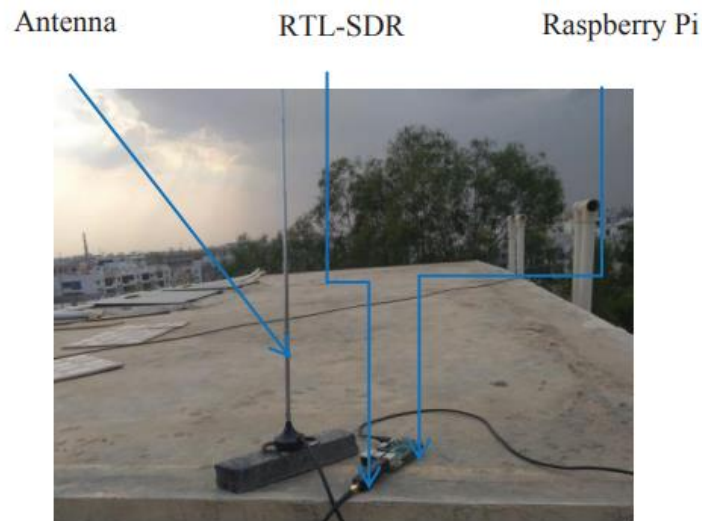


Figura 7: Configuración del receptor ADS-B [7]

Los hallazgos de esta investigación son relevantes para el proyecto actual. El uso efectivo de RTL-SDR y Raspberry Pi para rastrear y analizar trayectorias de aeronaves se alinea con los objetivos de utilizar hardware similar para predecir los tiempos de aterrizaje de aeronaves. Este estudio demuestra la viabilidad de emplear estos componentes rentables y eficientes para capturar, procesar y analizar datos de aviación, lo cual es crucial para el desarrollo de modelos de aprendizaje automático. Además, la implementación exitosa de la Raspberry Pi 3 Modelo B como procesador de datos en el estudio sugiere su potencial aplicabilidad en el proyecto actual para tareas de análisis de datos en tiempo real y predicción.

3.3 USO DE DATOS DE CÓDIGO ABIERTO PARA LA PREDICCIÓN DEL TIEMPO DE ATERRIAJE CON MÉTODOS DE APRENDIZAJE

AUTOMÁTICO— POR CHEN GONG, ET AL (2020, OPENSky SYMPOSIUM) *[8]*

Este trabajo previo está más estrechamente relacionado con los objetivos de este proyecto ya que utiliza el mismo centro de datos de código abierto, OpenSky, para la adquisición de datos históricos y toma en cuenta factores humanos como el ángulo de entrada y algunas decisiones de enrutamiento, además de considerar los datos meteorológicos.

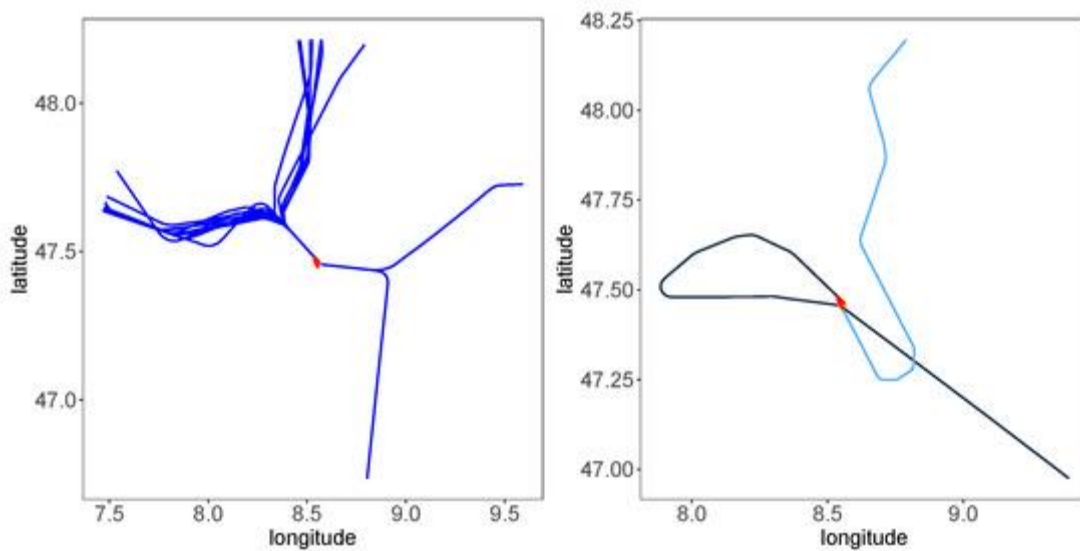


Figura 8: La diferencia en el ángulo de trayectoria entre el punto de entrada en el círculo y el punto de aterrizaje. (Izquierda) Trayectorias con menos tiempo de aterrizaje; (Derecha) trayectorias con más tiempo de aterrizaje [8]

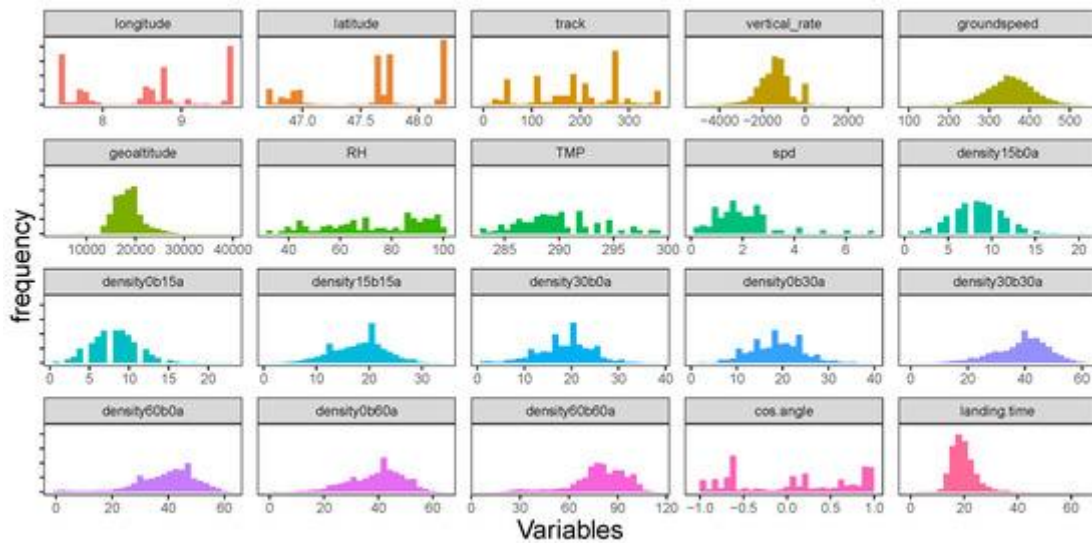


Figura 9: Histograma de predictores de aterrizaje [8]

Para la exploración de sus datos en el aeropuerto elegido (Zúrich), los autores hicieron descubrimientos significativos sobre cómo los planes de vuelo y el enrutamiento impactaban en los tiempos de aterrizaje. Especialmente a medida que cambiaban las distancias, el cambio en el tiempo predicho no cambiaba al mismo ritmo para todas las trayectorias. Algo tan simple como el ángulo de trayectoria (el ángulo relativo al aeropuerto medido cuando la aeronave está a una cierta distancia radial del aeropuerto) mostró tener un efecto significativo en el tiempo tomado para aterrizar, como se muestra en la figura 8. Además, como se visualiza en la figura 9, las distribuciones de los factores que afectan los tiempos de aterrizaje no son las mismas; en la mayoría de los casos, no siguen una distribución gaussiana, es decir, no están distribuidas normalmente. Por lo tanto, deben ser tratadas de manera diferente cuando se ingresan al modelo para evitar normalizar las distribuciones y aumentar significativamente los errores.

Después de extraer las características, se utilizan tres métodos de aprendizaje automático: la red neuronal feedforward penalizada (NN) y el método de árbol de decisión, incluidos Gradient Boosting Machines (GBM) y Random Forests (RF), para predecir el tiempo de aterrizaje. Las máquinas de boosting de gradiente usan una colección de aprendices débiles,

modelados por árboles de decisión, para construir un modelo de predicción fuerte. GBM utiliza un método iterativo para actualizar el modelo. Una red neuronal feedforward permite que la información en el espacio de características fluya a través de funciones o capas para mapear al valor objetivo. Esta información es particularmente útil ya que define los pasos que tomaremos para crear un modelo preliminar una vez que los datos hayan sido limpiados. Aunque en nuestro caso, seguiremos un modelo de regresión, estos modelos de predicción nos ayudan a valorar la certeza de nuestros resultados.

Este estudio también proporciona una base para probar la precisión del modelo utilizando dos ecuaciones, una para el Error Cuadrático Medio (RMSE) (ecuación 1) y otra para el Error Absoluto Medio (MAE) (ecuación 2). Los investigadores encontraron que los errores en el modelo de red neuronal parecían aumentar significativamente más allá de las 200 nm radialmente desde el aeropuerto. Por ello, para verificar la precisión del modelo desarrollado será preciso utilizar alguna de estas métricas de comprobación.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$$

Ecuación 1: Error cuadrático medio

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

Ecuación 2: Error absoluto medio

Los hallazgos de la investigación son bastante útiles para el proyecto ya que establecen un punto de partida y unos pilares que sostienen las bases de mi trabajo. Siendo la estructura y características de este proyecto muy parecidas al desarrollado, puedo prever las limitaciones que llevaron a diferencias significativas entre el tiempo de aterrizaje predicho y el real, y cómo mitigarlas. Además, el artículo proporciona una incursión en futuros experimentos que se pueden llevar a cabo para entrenar más eficazmente un modelo utilizando variables adicionales como pista asignada, tipo de aeronave o aeropuerto de origen, entre otros, características añadidas en mi estudio.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

En la industria aeroespacial, la puntualidad es crucial tanto para las aerolíneas como para los fabricantes de aviones y los retrasos de pocos minutos están generando un efecto dominó imparable que acarrea grandes pérdidas económicas. En la actualidad, esta está saturada y, si existen, las soluciones diseñadas para descongestionarla no son suficientes para resolver el problema.

En la última década, aprovechando el auge de la inteligencia artificial y el machine learning, se han desarrollado numerosos modelos probabilísticos para predecir la hora de llegada de un avión a su destino, una tarea que responde a la creciente necesidad de mejorar la puntualidad y eficiencia en la industria de la aviación. Estos modelos se han convertido en herramientas esenciales para aerolíneas, aeropuertos y controladores aéreos que permiten anticipar retrasos y optimizar la gestión del tráfico aéreo.

Las soluciones propuestas proporcionan estimaciones cada vez más precisas y dinámicas. Sin embargo, a pesar de los progresos significativos, estos modelos todavía enfrentan varios desafíos. Por ejemplo, la incertidumbre inherente a las condiciones meteorológicas, el tráfico aéreo imprevisto y otros factores externos estrechamente relacionados con las decisiones humanas, no siempre se puede modelar con exactitud, lo que a veces conduce a predicciones inexactas. Además, la integración de estos modelos en los sistemas operativos existentes puede ser compleja y costosa. Estos problemas evidencian que, aunque se ha avanzado mucho, aún es necesario seguir investigando y refinando estos modelos para abordar las limitaciones actuales y mejorar su precisión y utilidad.

Por ello, cualquier investigación que genere progresos en la eficiencia del sector, será un avance muy valioso para cualquier entidad de la industria. Cabe destacar que el proyecto expuesto es una actividad que se hace en colaboración con Boeing, una empresa líder en la

gestión y producción de los sistemas aeronáuticos en Estados Unidos. Boeing, presente en la mayoría de los procesos, ha detectado un problema y en lugar de contratar personal de investigación o pedir soporte a empresas externas, ha acudido a una entidad académica como la Universidad de Texas en Austin, para que los estudiantes de último curso realicen este servicio como un trabajo de fin de grado. En resumen, desarrollar un modelo preciso que pueda predecir la hora de llegada de los aviones es una necesidad real cuyos resultados suponen valiosos activos y beneficios para Boeing. Además, lo que diferencia a este proyecto de lo que ya está implementado en el sector, es la integración de las limitaciones de los modelos anteriores: variables impredecibles que alteran las decisiones del piloto. Se trata de características meteorológicas, ángulos de inferencia al aeropuerto destino y rutas de aterrizaje seguidas, siendo todas ellas causa de una alteración del trayecto del vuelo y por tanto una variación en su duración estimada.

4.2 OBJETIVOS

Este apartado se centra en los objetivos del proyecto, detallando tanto los generales como los específicos establecidos por la empresa. Estos objetivos incluyen la creación de un modelo de regresión para predecir la hora de llegada de vuelos y el desarrollo de una interfaz accesible para usuarios inexpertos. Además, se mencionan los requisitos y metas intermedias que guiarán el avance del proyecto hacia su finalización exitosa.

4.2.1 OBJETIVOS GENERALES

Por un lado, el objetivo final del proyecto es crear un producto manejable por un usuario inexperto en el ámbito. La intención es crear una interfaz a través de la cual el cliente, mediante un dispositivo con conexión a Internet, pueda interactuar y sea capaz de obtener información sobre la hora de llegada prevista de un determinado vuelo. Por otro lado, el objetivo más importante es conseguir un modelo de regresión capaz de predecir con máxima exactitud la hora de llegada de un vuelo a su aeropuerto destino. La idea es enfocar el entrenamiento del mismo en, además de características intrínsecas al propio vuelo (modelo de avión, velocidad, ...), variables impredecibles del factor humano como la ruta de

aterrizaje, o variables que afectan directamente al mismo como las condiciones meteorológicas adversas. En definitiva, la idea es crear una aplicación final en cuyo frontend se visualice un mapa de las inmediaciones de un aeropuerto y los aviones que lo sobrevuelan y cuyo backend realice las siguientes actividades:

- Acceder a los datos de aviones en vuelo en tiempo real en la nube
- Acceder al modelo de regresión almacenado en la nube
- Insertas estas nuevas entradas en tiempo real al modelo y predecir su hora de aterrizaje.

En paralelo con esta meta, se han preestablecido objetivos propios y marcados por la empresa, de igual o menor magnitud que requieren incluso más trabajo que la propia generación de la interfaz GUI. Se trata de los subsistemas del producto global, procesos llevados a cabo de forma separada cuya integración dará lugar al mecanismo comercializable requerido por Boeing.

4.2.2 OBJETIVOS TANGIBLES MARCADOS POR LA EMPRESA

Esto es un trabajo dirigido por Boeing, luego la empresa es la primera interesada en que el producto óptimo salga a la luz. Sin embargo, al tratarse de un proyecto tan ambicioso destinado a solventar un problema actualmente irresoluble por los líderes del sector, cualquier avance es un logro. Por ello, la propia empresa ha establecido requisitos y objetivos de menor escala, metas intermediarias que lleven al producto final o que, en su defecto, sean el punto de partida de investigación futura.

Para ello ha creado un repositorio Github confidencial que almacene todos los productos generados durante la actividad. Esta carpeta incluye lo siguiente:

- Diseño detallado del sistema del servicio de control de tráfico aéreo (ATC) SmartLanding: un conjunto de esquemas preciso que explique la integración, diseño y montaje de los diferentes subsistemas
- Data: obtención de todos los datos para entrenar el modelo. Una mastertabla que incluya:

- Todos los registros históricos tomados en año 2022 para los aeropuertos de Miami, Nueva York, Los Ángeles, Denver, Dallas, Chicago y Atlanta; filtrados, limpiados e incluyendo variables necesarias para el objetivo.
- Todos los registros históricos del año 2022 de las condiciones meteorológicas en las inmediaciones de los aeropuertos seleccionados.
- Diseño, montaje e instrucciones de configuración de los receptores de datos ADS-B
- Todo el código fuente, documentación, materiales de diseño implementados para el entrenamiento del modelo, acceso a la nube, obtención de registros por ADS-B, creación de la interfaz

4.3 METODOLOGÍA

Como ya se ha comentado anteriormente, la fase final del proyecto consiste en la integración de todos los subsistemas, asegurar la correcta conexión entre todas las partes y crear un proyecto complejo listo para funcionar, sin necesidad de intervención ni soporte técnico. Pero esto es el último paso en la cadena de actividades y por ello resulta necesario planificar los tiempos de realización de cada subsistema, coordinar a las partes implicadas y garantizar que estos cumplen con su funcionalidad individual antes de la integración. Por ello, en esta sección se explicará el método seguido durante el curso para la planificación, seguimiento y soporte de las distintas actividades y el cumplimiento en tiempo de los objetivos marcados.

Cabe destacar que el trabajo de fin de grado presentado en este documento forma parte de un gran proyecto empresarial. Este involucra a un significativo número de personas entre las cuales está un equipo de estudiantes, diversos tutores académicos y mentores o jefes de proyecto de Boeing, todos ellos muy diferentes, provenientes de ámbitos variados y con formas de trabajar distintas. Por esta razón, un reto añadido a la propia realización de actividades es la coordinación de todos los actores y la necesidad de mantenerlos informados sobre la evolución de todas las partes. Se trata de fijar un camino común, establecer directrices que delimiten los marcos de actuación de cada uno con el fin de favorecer que

todas las personas entiendan el objetivo final y como sus trabajos y objetivos individuales deben contribuir a la realización del mismo.

En primer lugar, es necesaria una división de tareas. Las tablas del anexo 2 (copiar del Project definition del drive) incluye las tareas asignadas a cada estudiante en función de sus conocimientos y habilidades. Como condición personal, al convalidar mi trabajo de fin de grado por este proyecto, requerí estar involucrada en todos los subsistemas de tal forma que mi aportación individual estuviese generalizada al conjunto del proyecto. Asumí la función de líder y además de tomar un papel activo en las distintas tareas, me aseguré de que la metodología de comunicación preestablecida y el intercambio de toda la información entre los miembros del grupo se mantuviese. Como se observa en la tabla, las tareas asignadas dividen a los estudiantes en subgrupos y estos trabajan conjuntamente, con una libertad de organización, para realizar los trabajos oportunos y cumplir con los requisitos generales. La coordinación e información del estado de todas las actividades se aseguró mediante reuniones de equipo semanales de 3 tipos. En la primera, todos los integrantes del grupo se reunían para comentar sus respectivos avances y problemas encontrados. Con esto, se y en los casos oportunos, se solicitaba ayuda al resto de compañeros. Un poco con el mismo fin, se agendaba una segunda reunión que incluía al asistente asignado al grupo, una figura académica con autoridad que daba soporte técnico y mantenía un seguimiento del correcto desarrollo de las tareas. La tercera y última sesión semanal juntaba a todos los integrantes con el jefe del proyecto de Boeing para mantenerle informado sobre los avances y compartir directrices. Por otro lado, se organizaban encuentros con menor asiduidad con los tutores de tecnología y redacción facilitados por la propia Universidad. Cuando las actividades realizadas requerían tareas técnicas más complejas, el tutor tecnológico ofrecía consejos y daba su punto de vista sobre la manera en la que debíamos afrontarlas. Estas tareas estaban principalmente ligadas a tareas de software más densas como la adquisición y limpieza de la mastertabla o el manejo de herramientas y librerías complejas para la creación del modelo. Como su nombre indica, el tutor de redacción daba feedback en los trabajos de redacción como informes para la empresa, diapositivas de presentaciones y papers para exposiciones.

Con esta metodología se mantenía a todos los actores y afectados por el proyecto en contacto, informados sobre la evolución de todas las tareas. Además, esto facilitaba la comprensión e implicación de todos en todas las partes del proyecto. Cada uno era responsable de una tarea determinada pero también entendía los demás segmentos y podía dar soporte en caso de necesitarlo. Esto incentivaba una motivación común para cumplir con el objetivo final, colaborar en conjunto para que ningún eslabón se desenganchase de la cadena de producción y cumplir con los tiempos estimados.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

Una gestión de proyectos efectiva requiere una comprensión profunda y un reporte transparente de las inversiones tanto monetarias como temporales. Esta sección evalúa los gastos financieros y de tiempo involucrados en nuestro proyecto, presentando un desglose detallado de los costes asociados al desarrollo e implementación del sistema de predicción de la hora de llegada de aeronaves. Al evaluar estos aspectos, nuestro objetivo es proporcionar a las partes interesadas una visión clara de las demandas económicas y de programación del proyecto, que son cruciales para mantener la alineación con el presupuesto y asegurar la finalización del proyecto a tiempo. Este análisis no solo ayuda a justificar los recursos asignados, sino que también facilita la planificación y gestión de futuros proyectos.

Los costes económicos y planificación temporal asociados al proyecto son aspectos vitales que considerar para su viabilidad y eficiencia global. A continuación, se detalla la gestión de gastos incurridos en la realización de todas las actividades, así como su planificación en el intervalo temporal asignado para el desarrollo de las mismas.

Como se describe en la Sección 4.4.1, el análisis de costes monetarios revela que el proyecto incurrió en aproximadamente \$200, asignados principalmente a la adquisición de hardware, licencias de software y acceso a API. Cabe destacar que los costos fijos se centran predominantemente en la antena, el ensamblaje del receptor y el software asociado, mientras que los costos variables se refieren a componentes de software que se cobran en función del

uso. Además, la Sección 4.4.2 profundiza en las implicaciones de los costes de tiempo, destacando los esfuerzos significativos invertidos en la recolección de datos, el procesamiento y la configuración de infraestructuras esenciales como el “bucket AWS S3”. Estas ideas proporcionan un contexto valioso para evaluar la asignación de recursos y la eficiencia operativa del proyecto.

4.4.1 COSTE ECONÓMICO

Al tratarse de un proyecto principalmente desarrollado en software que no incluye equipamiento hardware muy extenso, la inversión es baja y está principalmente destinada al dispositivo de recepción ADS-B y a los permisos requeridos en la elaboración del software. El coste monetario total del proyecto asciende a una cuantía de aproximadamente \$200, que incluye todo el hardware para el receptor, las licencias de software y el acceso a la API. De este valor, aproximadamente \$173 se gastaron en la antena y el resto en el software. La antena, el ensamblaje del receptor y su software correspondiente son los únicos costes fijos del producto, es decir, su precio no cambia con el uso del producto y es una tarifa única. La Tabla IV muestra la cuantía económica requerida para construir todo el proyecto, e incluye elementos como el costo, la cantidad, la descripción y el fabricante.

Cantidad	Tipo	Descripción	Fabricante	Número de Parte del Fabricante	Distribuidor	Número de Parte del distribuidor	Coste unitario	Coste total	Información
1	RCV	RTL-SDR USB Dongle	RTL-SDR	WRL-22957	Jetvision	474-WRL-22957	\$33.95	\$33.95	Receptor de la señal ADS-B
1	ANT	A3 ADS-B Antenna	Jetvision	SCO-1090-5	Jetvision	67020	\$83.00	\$83.00	Antena para recepción de señales ADS-B.

2	MTB	Antenna Mounting Bracket	Jetvision	N/A	Jetvision	67204	\$8.49	\$16.98	Soportes de montaje para la antena
1	CBL	15m Low Loss Coax Cable	Jetvision	70415	Jetvision	70415	\$38.76	\$38.76	Cable para conectar la antena y el receptor

Tabla 2: Asignación de precios a componentes del receptor ADS-B

Los costes variables del producto se relacionan con los componentes de software que se cobran en función del uso. Es importante notar que, al momento de las pruebas, los cargos totales por API y software ascendieron a menos de \$10, pero no hay un límite máximo dado el suficiente tráfico en el sitio web. Se estima que, con un uso moderado, la tarifa para el host no debería superar los \$50 en un mes dado en cargos por API y software.

4.4.2 PLANIFICACIÓN TEMPORAL

Con el fin de asegurar el correcto desarrollo de todas las actividades en tiempo se ha intentado seguir un planning de trabajo. Pero esta guía temporal ha estado sujeta a la primera etapa del proyecto, la colecta de datos. Además de ser el pilar fundamental, el punto de partida, sin el cual no puedo alcanzar ningún objetivo marcado, se ha tratado sin duda de la actividad más densa y engorrosa. El acceso a la base de datos histórica de OpenSky no es fácil. En primer lugar, se trata de datos densos, mucha información de todos los vuelos registrados durante mucho tiempo. En resumen, es una tabla que no se puede descargar en la memoria de un dispositivo en forma de archivo CSV o Excel y solo se puede acceder a través de consultas en Python. Y esta dificultad en el acceso ha formado un cumulo de obstáculos generando retrasos de varias semanas.

A continuación, se mencionan los problemas encontrados y las soluciones o alternativas llevadas a cabo para no bloquear la cadena:

El acceso a los datos históricos se realiza a través de consultas proporcionadas por una biblioteca "traffic" descargable en el entorno de Python. Sin embargo, una vez que se instancian estos comandos, el repositorio de OpenSky solicita la autenticación del usuario para acceder a los datos históricos. Los permisos necesarios no fueron inmediatos y mantuvieron bloqueado el proyecto casi un mes. Después de solicitar dicha autenticación al repositorio, el acceso a dichos datos sigue sin cumplirse. Ante la urgente necesidad de adquirir información, a principios de noviembre, tras reuniones con expertos y discutir con el tutor tecnológico de la facultad, se tomó la decisión de avanzar con el proyecto basado en una base de datos preliminar proporcionada por el propio repositorio OpenSky. Aunque no son los datos históricos sobre los que se pretende trabajar en última instancia, esta base de datos incluye una compilación más pequeña de los mismos datos cuya estructura es muy similar a la base que estamos buscando y las características presentadas son equivalentes. Las actividades relacionadas con los cálculos de variables, limpieza y procesamiento de datos se realizaron sobre esta tabla preliminar y una vez obtenidos los datos deseados, simplemente se traspasó el código a la base de datos oficial.

Paralelamente a este trabajo, se llevaron a cabo investigaciones externas relacionadas con el plan de trabajo posterior para poder prevenir futuros riesgos. Principalmente, se diseñaron todos los filtros y fórmulas de cálculo de variables y se realizó un estudio de mercado sobre las posibles alternativas para los componentes hardware del proyecto.

También con el fin de coordinar las actividades con los plazos de entrega previstos y planificar los recursos temporales, se realizó una programación que asignara todas las tareas a realizar a sus respectivos intervalos de tiempo de tal forma que cada una de ellas tuviese un plazo de finalización que diese paso al inicio de la siguiente tarea en la cadena. Dicho esquema se incluye en el anexo 3 como un diagrama Gantt.

En definitiva, tras unos imprevistos iniciales relacionados con la adquisición de los datos de entrenamiento, el diagrama de planificación Gantt fue un soporte clave para dar salida a

todas las tareas del proyecto. Gracias al mismo se puedo garantizar el desarrollo eficiente del proyecto y el alcance de resultados a tiempo.

Capítulo 5. SISTEMA/MODELO DESARROLLADO

Este capítulo describe el desarrollo de todo el producto llevado a cabo. Se describe el diseño del sistema global, así como la división, montaje y posterior integración de todos los subsistemas.

5.1 ANÁLISIS DEL SISTEMA

Crear un sistema capaz de identificar aeronaves y predecir sus tiempos respectivos requiere una combinación compleja de conjuntos de datos. En el núcleo del diseño, se encuentra un conjunto de datos integral que se fusiona a partir de fuentes diversas y confiables, abarcando características de aeronaves, patrones meteorológicos y una adquisición de datos de hardware robusta. El enfoque es metódico y ha conllevado numerosos problemas y continuara suponiendo desafíos para futuras investigaciones. A lo largo del proceso, también se han integrado estrategias de mitigación de riesgos, específicamente en áreas de filtrado y recolección de datos. Estas medidas también se sincronizan con el desarrollo de un modelo de aprendizaje automático para refinar la precisión de las predicciones de tiempos de aterrizaje. Esta sección profundiza en la arquitectura del sistema destacando el marco colaborativo de los diversos componentes en juego para el diseño y montaje del sistema final.

5.2 DISEÑO

A un alto nivel, el sistema está diseñado para predecir los tiempos de llegada de las aeronaves a través de un conjunto de datos integral. Ese conjunto de datos incluye datos meteorológicos, información sobre el tipo de aeronave y recolección confiable de datos de hardware. El diseño incorpora dos experimentos principales de reducción de riesgos que involucran el filtrado del conjunto de datos y la reducción del riesgo de recolección de datos.

Estos trabajan en conjunto con un robusto modelo de aprendizaje automático que se desarrollará para optimizar la gestión de los vuelos en aeropuertos de destino, con el fin de optimizar específicamente las capacidades de las pistas. Para la parte de filtrado del conjunto de datos, identifiqué las características importantes a extraer de la base de datos de NOAA y seleccioné las estaciones meteorológicas más cercanas a los principales aeropuertos objetivo. Fue importante centrarse solo en características que pueden jugar un papel potencial en los retrasos de vuelos como la velocidad del viento, la precipitación, etc. Después de aislar estas características, la base de datos meteorológica resultante se fusionó con la base de datos de aeronaves de Opensky para crear una nueva base de datos lista para entrenar. La base de datos resultante contenía toda la información necesaria para entrenar el modelo de aprendizaje automático de TensorFlow con precisión.

La interfaz GUI tiene los resultados del modelo en el backend y es capaz de mostrarlos en el frontend, que el usuario ve en tiempo real. Esta muestra los vuelos en movimiento, y al hacer clic en un avión específico, puede mostrar las predicciones hechas para los tiempos de aterrizaje, así como demás información importante.

Por último, para comprobar el rendimiento del modelo se utiliza un conjunto de test cuyos datos provienen de los registros capturados por el receptor ADS-B. EL rendimiento del modelo se apreciará de forma numérica mediante el calculo del error cuadrático medio (ecuación 1) entre los valores de la variable objetivo “segundos restantes para aterrizar” predichos por el modelo y aquellos reales, en el conjunto de test.

Todo el sistema descrito se observa en el diagrama de la figura 10.

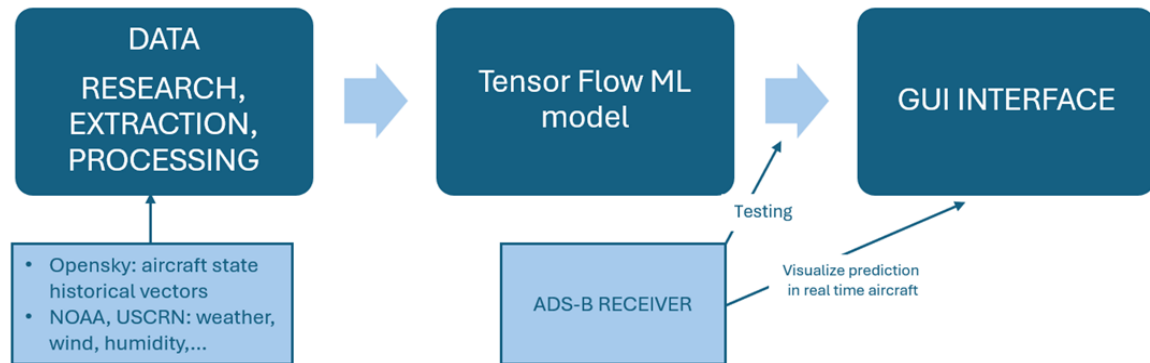


Figura 10: Esquema de la división y sucesión de actividades del proyecto

5.3 IMPLEMENTACIÓN

Al estar diseñado para una futura implementación en el sector, es esencial reconocer la necesidad de un enfoque estructurado, dividiendo el trabajo en subsistemas manejables. Cada una de estas subpartes juega un papel crucial en el éxito general del proyecto. Desde la investigación y extracción de datos hasta el entrenamiento del modelo, la visualización GUI y la configuración del hardware con el receptor ADS-B, cada aspecto requiere atención cuidadosa y experiencia. Estas tareas se desglosan metódicamente en una serie de etapas manejables de manera que el alcance no sea tan abrumador. Esto requiere el uso óptimo de los recursos, el cumplimiento de los plazos y la flexibilidad para adaptarse al proyecto a medida que demás subsistemas evolucionan.

Por otro lado, la integración de estas subpartes será primordial. La combinación exitosa de las tareas culminará en un sistema interconectado que cumpla con los objetivos del proyecto. Al coordinar los esfuerzos de cada subparte y alinearlos hacia un objetivo común, se logra una solución integral que cumpla con los requisitos del proyecto y supere las expectativas.

5.3.1 DIVISIÓN, DISEÑO Y MONTAJE DE SUBSISTEMAS

En esta sección va enfocada la división y especificación de cada subsistema. Este desglose estructurado mejora la manejabilidad y enfoca la experiencia en tareas especializadas, desde la recolección y procesamiento de datos hasta el entrenamiento del modelo y el desarrollo de la GUI. Al delinear el trabajo en subpartes, se asegura un enfoque cohesivo que se integra sin problemas para formar un sistema robusto y eficiente. Esta sección describe los roles, procesos y resultados de cada subparte, proporcionando una vista completa de cómo los esfuerzos particulares contribuyen a alcanzar los objetivos finales del proyecto.

5.3.1.1 Investigación, extracción y procesamiento de datos

La primera tarea fue la colección de datos relevantes de las fuentes apropiadas y la transformación de esos datos en un conjunto de datos utilizable para el aprendizaje automático. Por un lado, se accedió a la fuente de OpenSky y adquirieron registros históricos de aeronaves en los aeropuertos de interés. Por otro lado, se llevó a cabo una búsqueda de registros meteorológicos históricos en las fechas y destinos de interés, así como el procesamiento de estos datos externos. Por último, tras la preparación aislada de cada tipo de datos, se integraron todas las tablas en un solo conjunto de datos listo para el entrenamiento. Con respecto a la fusión con la base de datos de OpenSky, utilicé la fecha como variable key. Convertí el formato de fecha y hora de la base de datos de NOAA al mismo formato que la base de datos de OpenSky para que la fusión según la fecha fuera fluida. Por último, añadí una columna de aeropuerto que identificaba el aeropuerto correlacionado con cada estación meteorológica en la base de datos final fusionada para un análisis más eficiente en la parte de entrenamiento del modelo.

Para obtener datos de OpenSky, se solicitó acceso a la base de datos OpenSky Trino. Esta base de datos permitió consultar datos de vectores de estado históricos de OpenSky. Estos vectores de estado contienen datos sobre la posición y la velocidad de una aeronave en un momento específico. Para ello utilicé la biblioteca Python Traffic, una interfaz de Python diseñada para analizar datos de OpenSky; consultar datos de aeronaves entrantes para los

principales aeropuertos de EE. UU. La captura mostrada en la figura 11 presenta los atributos obtenidos para un vector aleatorio. La fila referente a la densidad del tráfico “traffic density” corresponde al número de aviones a una distancia igual o inferior a x millas náuticas del aeropuerto. Una vez que se obtuvieron los datos del vector de estado de la aeronave, estos se procesaron utilizando diferentes funciones y técnicas matemáticas, como la ecuación de distancia haversine, para obtener atributos físicos diversos como la velocidad, ángulo coseno, así como el tiempo de llegada para cada aeronave en preparación para el entrenamiento.

Category	Feature	Description
Trajectory	latitude	33.39499263440148
	longitude	-97.70735760124364
	geopotential altitude	19300.0
	ground speed	326.0
	track	126.2727223048272
	vertical rate	-1664.0
	cos.angle	-0.5916294208773368
Traffic Density	15 min before	5
	15 min after	6
	15 min BA	11
	30 min before	5
	30 min after	5
	30 min BA	10
	60 min before	2
	60 min after	4
	60 min BA	6

Figura 11: Tabla de los atributos

Para la adquisición de datos meteorológicos, los aeropuertos no suelen publicar variaciones en sus registros meteorológicos a intervalos cortos. Por lo tanto, accedí a la fuente pública NOAA (Administración Nacional Oceánica y Atmosférica) ([2]) que recopila una gran cantidad de información meteorológica en diversas áreas del territorio nacional. Específicamente, me centré en la USCRN (Red de Referencia Climática de EE. UU.),

asociada con NOAA. Esta web constituye una red de estaciones de monitoreo climático sistemático localizadas por todo el territorio estadounidense. Estas estaciones utilizan instrumentos de alta calidad para medir la temperatura, la precipitación, la velocidad del viento, las condiciones del suelo y otras características relevantes. A diferencia del resto de datos meteorológicos, los registros tomados por USCRN resultan muy útiles para nuestro caso de uso ya que se toman en intervalos de 5 minutos, lo que añade variaciones sutiles en los patrones meteorológicos y permite obtener valores más precisos en el modelo.

Como se observa en la figura 12, estas estaciones están distribuidas uniformemente en varios puntos del territorio, por lo que no están ubicadas precisamente en los aeropuertos de interés. Por lo tanto, para asignar estos datos a cada aeropuerto, realicé un estudio meteorológico entre cada aeropuerto y sus estaciones de medida más cercanas. A través de este proceso, se asignó a cada aeropuerto su estación USCRN más cercana, siempre que no estuviera afectada por patrones meteorológicos significativamente diferentes de los esperados en el aeropuerto. Por ejemplo, se evitaron diferencias de altitud entre la estación y el aeropuerto, así como estaciones ubicadas en la costa para aeropuertos más interiores. En la tabla 3 se incluye la asignación de estaciones a cada aeropuerto.

<i>AEROPUERTO</i>	<i>ESTACIÓN USCRN</i>
<i>Aeropuerto Internacional de Dallas Fort Worth</i>	<i>Sensor de Austin Texas</i>
<i>Aeropuerto Internacional de Miami</i>	<i>Sensor del aeropuerto de Titusville</i>
<i>Aeropuerto JFK de Nueva York</i>	<i>Sensor de Kingston</i>
<i>Aeropuerto Internacional O'Hare de Chicago</i>	<i>Sensor de Shabbona</i>
<i>Aeropuerto Internacional de Denver</i>	<i>Sensor de Nunn</i>

<i>Aeropuerto Internacional de Los Ángeles</i>	<i>Sensor de Fallbrook</i>
<i>Aeropuerto Internacional de Atlanta</i>	<i>Sensor de Watkinsville</i>

Tabla 3: Asignación de estaciones a cada aeropuerto

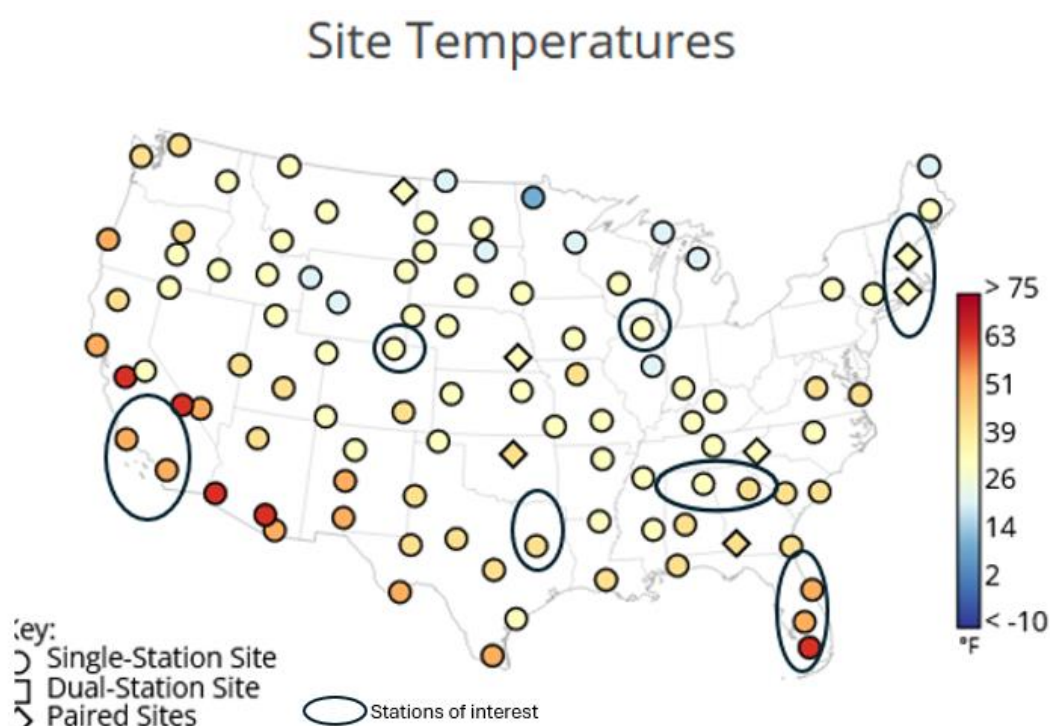


Figura 12: Mapa de estaciones USCRN en todo el territorio nacional

Cada estación recoge registros a intervalos de 5 minutos medidos en la superficie. Entre todas las variables recopiladas, limpié todos los conjuntos de datos y me centré en las siguientes:

- WBANNO: El número WBAN de la estación.
- UTC_DATE: La fecha UTC de la observación.
- UTC_TIME: La hora UTC al final del período de observación de 5 minutos.
- AIR_TEMPERATURE: Temperatura promedio, en grados Celsius.
- PRECIPITATION: Cantidad total de precipitación, en milímetros.

- **SOLAR_RADIATION:** Radiación solar promedio, en vatios por metro cuadrado.
- **WIND_SPEED:** Velocidad promedio del viento, en metros por segundo.
- **WIND_DIRECTION:** Dirección promedio del viento en grados desde el norte.

Para la combinación de ambos conjuntos de datos, aeronaves y meteorología, en cada aeropuerto, se ha utilizado la variable temporal como “key”. De esta manera, cada registro de aeronave se asocia con las características temporales que lo afectan en ese momento cerca del aeropuerto. Cabe destacar que las líneas de las bases de datos de aeronaves tienen una precisión temporal de 1 segundo (cada segundo se reciben registros de aeronaves). Sin embargo, los datos meteorológicos se envían cada 5 minutos. Por esta razón, se realizó una fusión aproximada. Se creó una variable auxiliar para asignar a cada tiempo en la tabla de aeronaves su valor de tiempo más cercano divisible por 5 minutos. De esta manera, los valores obtenidos en esta variable auxiliar coinciden con los valores temporales de las bases de datos meteorológicas, y la fusión de ambas bases de datos se puede realizar utilizando estos valores. La figura 13 muestra el esquema que se siguió para hacer el join de las dos bases de datos y La figura 14 muestra una porción de la “mastertabla” que incluye todos los atributos que afectan al tiempo de vuelo.

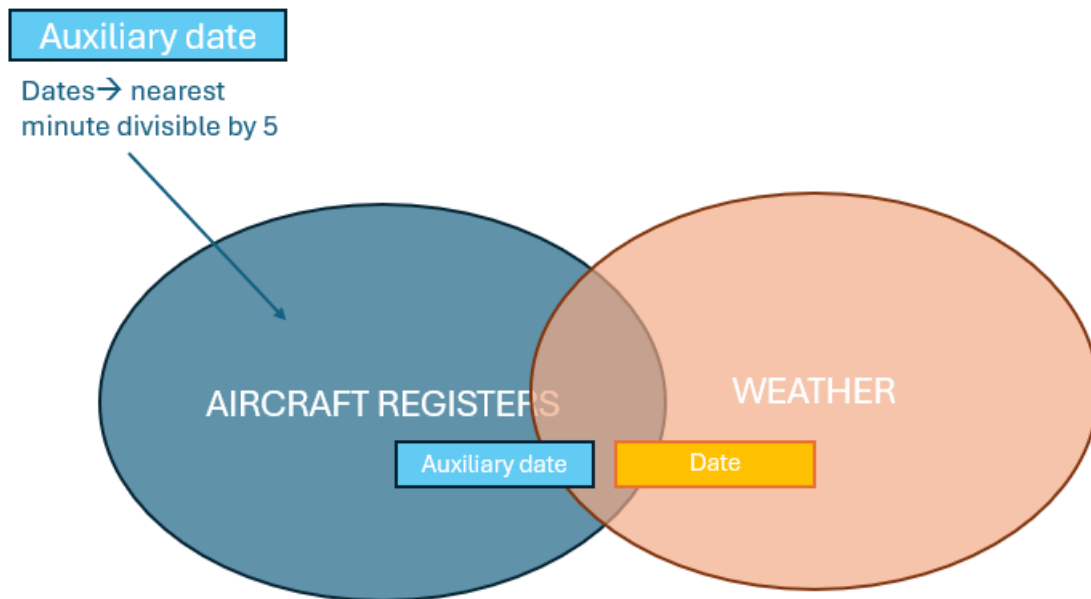


Figura 13: Esquema del join entre las tablas de registros aéreos y meteorológicos

Unnamed: 0	timestamp_x	icao24_x	latitude	longitude	groundspeed	track	vertical_rate	callsign	altitude	...	LATITUDE	AIR_TEMPERATURE	PRECIPITATION	RELATIVE HUMIDITY	WIND_1_5	V
0	0	2020-01-01 00:00:01+00:00	0xa224f9	40.763560	-89.343754	433.0	42.193630	-1856.0	UAL474	26875.0	...	41.84	-2.9	0.0	76	4.37
1	1	2020-01-01 00:00:01+00:00	0xa224f9	40.763560	-89.343754	433.0	42.193630	-1856.0	UAL474	26875.0	...	41.84	-0.3	0.0	82	3.04
2	2	2020-01-01 00:00:01+00:00	0xa224f9	40.763560	-89.343754	433.0	42.193630	-1856.0	UAL474	26875.0	...	41.84	0.0	0.0	80	1.98
3	3	2020-01-01 00:00:01+00:00	0xa224f9	40.763560	-89.343754	433.0	42.193630	-1856.0	UAL474	26875.0	...	41.84	-0.7	0.0	65	0.49

Figura 14: Ejemplo de registros en la tabla final "mastertabla"

5.3.1.2 Entrenamiento del modelo

Antes de entrenar el modelo, el conjunto de datos de entrada necesitaba algunas modificaciones que mantuvieran cierta coherencia de formato y requisitos propios al modelo para obtener resultados congruentes y exitosos. Para ello, en primer lugar, se escalaron uniformemente todas las variables para que tomaran valores proporcionales comprendidos entre 0 y 100. Esto permite evitar desajustes, es decir que todas las variables sean consideradas con el mismo peso de entrada por el modelo de tal forma que sus coeficientes asignados en salida no dependiesen de su magnitud sino de su importancia real para la predicción del objetivo. En segundo lugar, teniendo en cuenta que se trataba de un modelo

de regresión, el target debía ser de tipo numérico. Por ello, se codificó el instante temporal de llegada como tiempo restante en segundos para que el modelo pueda predecir una salida continua en lugar de un objeto de fecha y hora. Además, utilizando Sci-kit learn, una biblioteca de aprendizaje automático para Python, se agruparon los datos geográficos utilizando clustering de k-means y se agregaron como una característica. Por último, se realizó una limpieza exhaustiva y sucesivos cálculos para la obtención de nuevas variables necesarias para calcular con precisión el tiempo de llegada restante en cada vuelo. El procedimiento seguido se detalla a continuación.

Primero, se estableció un filtro basado en la distancia de los aviones al aeropuerto, asegurando que solo se consideraran los aviones en fase de aproximación. Además, se verificó la trayectoria del avión cuando la distancia era menor a 1 milla náutica, confirmando que el avión estaba en una ruta de aproximación adecuada. Para definir el tiempo de llegada, se consideró el momento en que la distancia era menor a 1 milla náutica y la trayectoria del avión era aproximadamente igual al ángulo de la trayectoria esperada. Esta condición ayudó a determinar el momento exacto en que el avión estaba a punto de aterrizar.

Este enfoque resolvió el problema de los vuelos que podrían quedar fuera del rango de muestreo. Al eliminar los vuelos que no se acercaban al ángulo de trayectoria correcto y que no estaban lo suficientemente cerca del aeropuerto, se mejoró la precisión de los datos utilizados.

Para enriquecer el modelo de predicción, se añadieron características adicionales al conjunto de datos. Estas incluyen las distancias en los ejes y y x desde el aeropuerto (y_distance y x_distance), las velocidades en los ejes y y x (y_velocity y x_velocity), el coseno de la trayectoria (útil para cálculos vectoriales relacionados con la dirección del avión), la distancia total desde el aeropuerto, y el tiempo hasta la llegada codificado en segundos restantes como se menciona más arriba.

Este enfoque meticuloso en la limpieza y preparación de los datos garantiza que el modelo de predicción tenga la información más relevante y precisa posible para calcular los tiempos de llegada de los aviones con alta exactitud.

El entrenamiento real del modelo se realizó utilizando XGBoost, una biblioteca de árboles de decisión potenciados por gradiente. Sci-kit learn se utilizó nuevamente para medir métricas del modelo como el error cuadrático medio.

5.3.1.3 Hardware y receptor ADS-B

La configuración de hardware ADS-B es crucial para recopilar datos ADS-B en vivo de las aeronaves, lo cual permite un seguimiento preciso de los aviones. A continuación, se describe cómo funciona esta configuración, esquematizada en la figura 15.

- **Transmisión de Paquetes ADS-B por las Aeronaves**
 - Las aeronaves transmiten paquetes de datos ADS-B en la frecuencia de 1090 MHz.
- **Recepción de la Señal por la Antena**
 - Una antena A3 Jetvision, colocada en una posición elevada para maximizar el rango de recepción, capta los paquetes ADS-B transmitidos por las aeronaves.
- **Transmisión de la Señal al Receptor**
 - La señal captada por la antena se transmite al receptor RTL-SDR V3 R860 mediante un cable coaxial de baja pérdida de 15 metros. Este cable asegura una mínima degradación de la señal durante la transmisión.
- **Procesamiento de la Señal por el Receptor**
 - El receptor, conectado a un ordenador a través de una conexión USB, recibe y procesa los datos ADS-B.
- **Envío de los Datos al Ordenador**
 - El receptor transfiere los datos procesados al ordenador.
- **Subida de los Datos a la Nube**
 - El ordenador, conectado a Internet, envía los datos ADS-B a un bucket de AWS S3, donde se almacenan para su procesamiento y análisis futuro.

Los dos elementos de software utilizados para el procesamiento de los datos son rtl1090 para la recepción en vivo y PlanePlotter para la transformación de datos en archivos ejecutables. A continuación, se detallan sus especificaciones

- **rtl1090**
 - Este software se utiliza para inicializar el receptor RTL-SDR y comenzar a captar los datos ADS-B.
- **PlanePlotter**
 - PlanePlotter captura los datos ADS-B en archivos de registro. Estos archivos contienen información detallada sobre las aeronaves, como velocidad en tierra, altitud, ICAO, etc.

Para más detalle la figura 15, muestra un diagrama del flujo productivo del sistema hardware, recepción, captura, transformación y envió a la nube de los datos ADS-B. La figura 16 es una captura de pantalla de lo que se observa en el “laptop” una vez se reciben los datos.

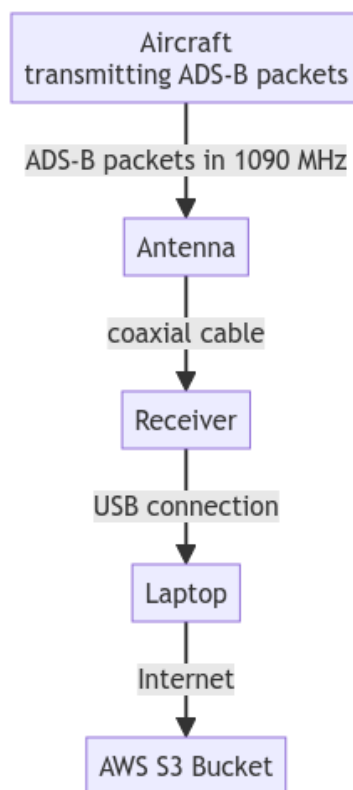


Figura 15: Diagrama del flujo de datos desde la detección por la antena hasta el almacenamiento en la nube

ICAO	C/S	ALT..MCP	V/S	GS	TT	SSR	G*456^	SG	MSGs
ACD1F6		F230		362	009			4	3
A5725A		F360						5	3
A03397								3	1
A20A81		F248	+23	366	014			6	13
A6B202								4	1
A64E0B	N505RS	F018	- 2	102	026	2445	*	13	33
A57412								4	2
AD517D								5	1
AB6C26	AAL1083	F080	- 9	289	154	1501	*	12	100
AB8F0D		F360		405	244		*	13	72
AB5628	N83BP	F048	+ 3	124	276			17	24

Figura 16: Captura de los registros observados en la herramienta de visualización en el ordenador

5.3.1.4 Visualizador GUI

Comprender el visualizador GUI implica primero comprender cómo recibe datos, así como el flujo de datos en los componentes circundantes. La interfaz de la canalización GUI se representa en la figura 17.

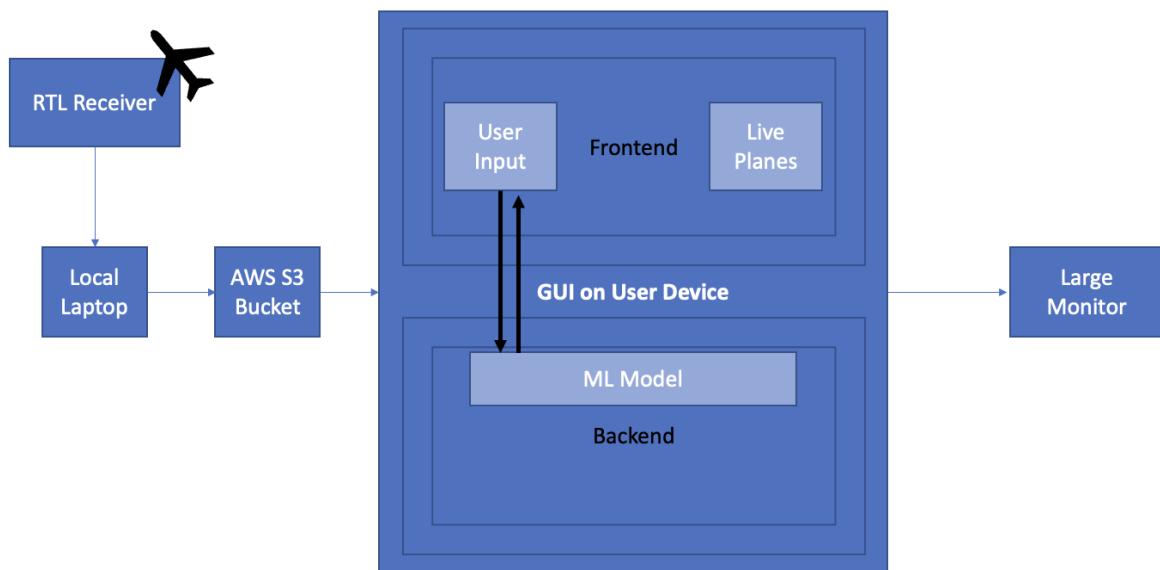


Figura 17: Diagrama del flujo de datos con la interfaz GUI

Como se ve en el diagrama anterior, la GUI recibe las solicitudes de datos del usuario desde el frontend, que tiene el papel de mostrar aviones en vivo en el mapa basado en datos ADS-B, así como formularios que permiten al usuario filtrar esos datos en vivo para mostrar solo lo que se desea. El receptor RTL envía los datos a un host local que ejecuta un script para conectar la laptop a un bucket AWS S3 que finalmente almacena los datos. El backend de la interfaz GUI se conecta al bucket AWS S3 del que extrae los datos correspondientes al vuelo consultado por el usuario, ejecuta el algoritmo de aprendizaje automático y extrae los resultados, tiempo de llegada estimado. El frontend y el backend funcionan en paralelo de manera eficiente para proporcionar información en tiempo real al usuario.

Los detalles de la propia interfaz gráfica, mostrada en la figura 17, se especifican a continuación.

Esta proporciona una manera interactiva de obtener información útil. La GUI está diseñada para facilitar que los usuarios vean exactamente lo que desean y reducir los datos de entrada al modelo de predicción, restringiéndolos únicamente a aquellos vuelos solicitados por el usuario. Esto además de ser un aspecto técnico del propio proyecto, favorece la eficiencia del funcionamiento en la nube pues, en lugar de hacerlo de manera continua, el modelo se ejecuta únicamente con las entradas demandadas por medio de esta interfaz.

El propósito principal de esta interfaz es mostrar datos en tiempo real de vuelos, permitiendo a los usuarios hacer clic en un avión específico para ver su tiempo de aterrizaje. La interfaz está diseñada con un backend en Python y un frontend en React, utilizando Flask para la comunicación entre el frontend y el backend. React es una biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario. Se utiliza principalmente para desarrollar aplicaciones de una sola página (SPA), donde el contenido de la página se actualiza dinámicamente sin recargar toda la página. Permite crear una interfaz de usuario interactiva y dinámica, como la que muestra el mapa. Flask es un microframework para aplicaciones web escrito en Python. Proporciona solo lo esencial para construir aplicaciones web, y es altamente extensible con diversas bibliotecas y plugins. Flask recibe las solicitudes del frontend, procesa los datos (como el número de vuelo y destino ingresados por el usuario), y devuelve la información necesaria para ser mostrada en la interfaz de usuario.

La figura 18 muestra cómo se ve la interfaz por el usuario del sistema. En la parte superior de la interfaz, hay campos donde los usuarios pueden ingresar el número de vuelo, el destino y el número de cola del avión. Al hacer clic en "SUBMIT", la interfaz procesa estos datos para mostrar resultados en tiempo real. En la sección de resultados, se muestra el tiempo restante hasta el aterrizaje y la hora de aterrizaje prevista.

Debajo de esta sección, hay un mapa que muestra la ubicación de varios aviones en el área alrededor del Aeropuerto Internacional de Dallas/Fort Worth. En el mapa, cada avión está

representado por un icono. Al hacer clic en uno de estos iconos, se muestra información detallada sobre el vuelo, incluyendo el número de vuelo y el tiempo de llegada previsto.

Por ejemplo, en la imagen proporcionada, el vuelo 10197 tiene una hora de llegada estimada a las 12:00 AM y está representado por un icono en el mapa. La información adicional como el tiempo restante hasta el aterrizaje se muestra también, proporcionando a los usuarios una vista clara y concisa de los datos de vuelo en tiempo real.

En resumen, esta interfaz gráfica de usuario facilita la interacción con el modelo de predicción de tiempos de llegada de aviones, ofreciendo una herramienta poderosa y accesible para usuarios sin necesidad de conocimientos técnicos avanzados.

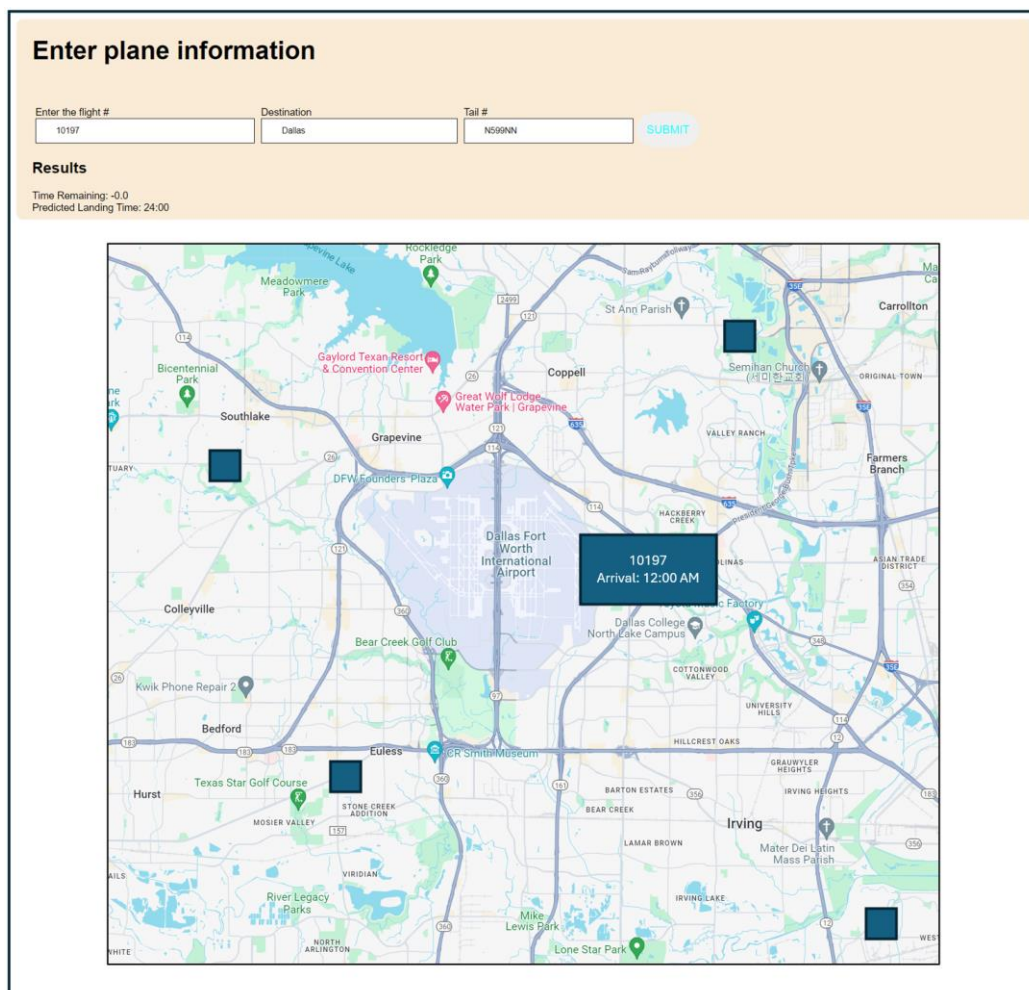


Figura 18: Vista de la interfaz por el usuario

5.3.2 INTEGRACIÓN DE SUBSISTEMAS

Para garantizar un manejo y accesibilidad eficientes de los datos ADS-B en vivo, el proyecto integra soluciones de almacenamiento en la nube utilizando buckets AWS S3. Como queda explicado en el capítulo 2: “Descripción de las tecnologías”, el bucket S3 AWS es un servicio de almacenamiento de objetos en la nube proporcionado por Amazon Web Services (AWS), donde se puede almacenar y recuperar cualquier cantidad de datos en cualquier momento. Específicamente, se eligió AWS S3 por su escalabilidad para almacenar desde archivos pequeños hasta grandes archivos de datos. Con un archivo de registro continuamente actualizado lleno de miles de líneas de datos ads-b, la escalabilidad del bucket S3 permite diversas necesidades de almacenamiento. También se eligió S3 por su accesibilidad, ya que los datos pueden ser accesibles para cualquier persona, lo cual es crucial para este proyecto cuando se necesita distribuir los datos ADS-B en vivo.

Además, por su durabilidad y disponibilidad, el bucket S3 ofrece la capacidad de almacenar datos incluso cuando no está operativo. Esto es crucial para el almacenamiento de datos en comparación con una instancia EC2, ya que esta última se elimina una vez que la instancia se detiene o termina.

El uso del bucket AWS S3 requiere una configuración inicial que incluye la descarga de las herramientas CLI (Command Line Interface) para interactuar con el bucket S3 a través del símbolo del sistema o la terminal, permitiendo el acceso directo a la vista de los datos. La configuración también implica la creación del nuevo bucket S3 a través de la consola de administración de AWS, lo que incluye establecer el nombre del bucket, su dirección, y configurar los permisos correctos para asegurar la privacidad y seguridad de los datos.

Los datos ADS-B en vivo capturados por el receptor se registran a través del software PlanePlotter mediante la inicialización de rtl1090. Un script luego convierte estos archivos de registro en archivos JSON para luego ser enviados al bucket S3 y asegurar que los datos estén estructurados y sean fácilmente accesibles para procesamiento futuro. El script también

maneja el envío de datos JSON en vivo al bucket S3, que mantiene estos registros ADS-B en la nube en intervalos predefinidos asegurando la disponibilidad de datos en tiempo real.

Otro script extrae los datos más recientemente modificados del bucket S3, con los archivos JSON almacenados que se recuperan periódicamente del bucket S3 y carga los datos en variables locales de Python a partir de los objetos JSON. Estos datos del bucket S3 se inyectan posteriormente como entradas al modelo de ML para permitir actualizaciones dinámicas basadas en datos en vivo actuales. Finalmente, los resultados de predicción obtenidos por el modelo se muestran en un visualizador GUI para permitir a los usuarios ver las aeronaves y su aterrizaje previsto. La figura 19 ilustra la arquitectura del sistema mencionada y cómo la nube AWS es el elemento focal que reúne todos los subsistemas del proyecto.

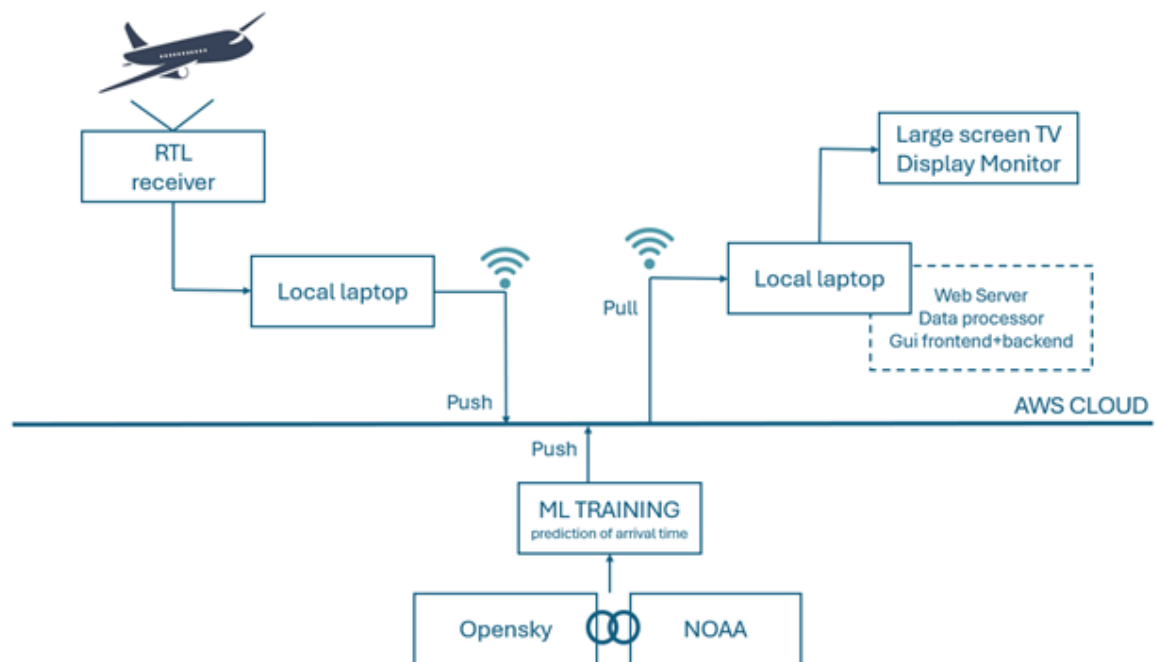


Figura 19: Arquitectura del sistema global y flujo de datos integrado

5.3.3 PRUEBAS DE EVALUACIÓN

El modelo se evaluó utilizando Sci-kit learn. El conjunto de datos de entrenamiento se dividió en conjuntos de entrenamiento y prueba. Una vez que el modelo terminó de entrenarse en el conjunto de datos de entrenamiento, se validó ejecutando inferencias en el conjunto de prueba y midiendo el error con el error cuadrático medio.

Las pruebas en los componentes de hardware se llevaron a cabo utilizando software como el administrador rtl1090 y PlanePlotter. Estas herramientas fueron necesarias para facilitar las pruebas porque, al utilizarlas, el probador podría monitorear la actividad y visualizar si la antena y el receptor estarían capturando paquetes ADS-B de la antena y el receptor y procesándolos a través de las respectivas herramientas de software. Si la antena y el receptor estaban funcionando correctamente, habría columnas respectivas que mostrarían información ADS-B destacando la aeronave respectiva, como ICAO, velocidad en tierra, registro de vuelo, latitud y longitud, etc. Se realizaron más pruebas inspeccionando los archivos de registro para correlacionar los parámetros generados por PlanePlotter con sus valores respectivos generados en estos archivos de registro.

La integración del bucket S3 con el hardware se probó asegurando que, una vez configurado el bucket S3 en Amazon Web Services (AWS), se utilizaban las herramientas de la Interfaz de Línea de Comandos (CLI) para configurar el bucket S3 en el portal. Las pruebas del bucket S3 se realizaron enviando archivos al bucket S3 y recuperando esos mismos archivos en la misma máquina local; tan pronto como se confirmó esto, se modificó el script del hardware para enviar estos archivos en sincronía con el temporizador de 10 minutos que convierte el archivo de registro a JSON.

Se agregó una declaración print en el proceso de depuración para mostrar si el archivo se había subido al bucket S3 y, si no, otra declaración print para mostrar que el archivo no se había subido al bucket S3.

También se utilizó un script de extracción durante las pruebas para asegurarse de que los archivos en el bucket S3 pudieran ser descargados a la máquina local conectando el cliente

S3 utilizando la clave de acceso y la clave de acceso secreta configuradas durante la configuración del bucket S3 y utilizando un subproceso para recorrer recursivamente todo el bucket S3 y descargar estos archivos a la máquina local ejecutando el comando S3 para copiar y pegar los archivos en una ruta de archivo deseada y haciendo esto repetidamente para todos los archivos existentes en el bucket S3.

Se realizaron pruebas adicionales marcando encabezados clave en este caso como: velocidad en tierra, latitud, longitud y altitud para asegurar que el registro de vuelo y el ICAO coincidieran con su vuelo respectivo durante la recolección de datos. Esto también se hizo para visualizar los datos ADS-B en vivo a lo largo del tiempo y comprender el comportamiento de las diferentes características que podrían influir en el tiempo de aterrizaje, como los cambios de altitud y los cambios de velocidad. La Figura 20 resalta las pruebas realizadas mientras se registraban los datos de vuelo en vivo y esto se hizo como se mencionó anteriormente para visualizar algunas de las propiedades de las aeronaves para ayudar a comprender el comportamiento de las aeronaves y usar estos datos para el análisis de los tiempos de aterrizaje previstos.

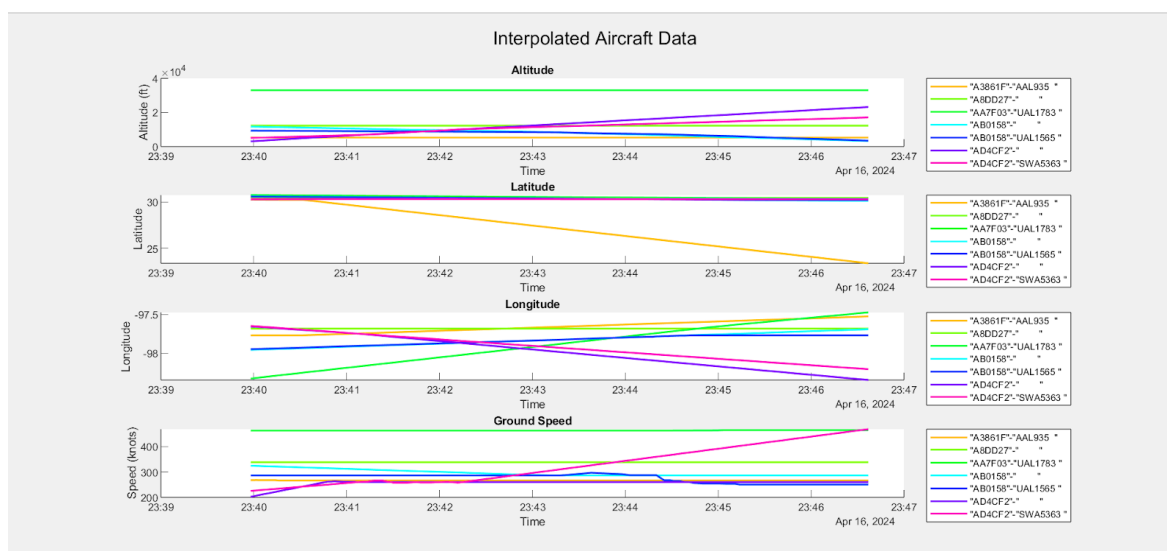


Figura 20: Gráficas del comportamiento de variables en los datos ADS-B recibidos

Capítulo 6. ANÁLISIS DE RESULTADOS

En el presente Trabajo de Fin de Grado (TFG) se ha desarrollado un modelo de predicción para estimar la hora de llegada de un avión a su aeropuerto de destino. El modelo se centra tanto en características intrínsecas del avión como en factores externos relacionados con decisiones humanas, tales como condiciones meteorológicas y la trayectoria al aterrizar. En este análisis de resultados, se detallan los logros parciales y totales del proyecto, destacando la obtención de un dataset adecuado, la integración de una plataforma en la nube y el rendimiento final del modelo del que un usuario pueda extraer información por medio de una interfaz gráfica.

6.1 RESULTADOS PARCIALES

6.1.1 OBTENCIÓN DE UN DATASET CORRECTO Y MANEJABLE

Uno de los primeros desafíos enfrentados en este proyecto fue la recopilación y preparación de un conjunto de datos robusto y manejable. Este dataset debía incluir no solo las características técnicas de los aviones, como su tipo, velocidad y capacidad de combustible, sino también datos externos como condiciones meteorológicas, control del tráfico aéreo y decisiones de los pilotos.

La selección y limpieza de datos implicaron la eliminación de valores atípicos y la imputación de datos faltantes, garantizando la calidad y la integridad del dataset. Se logró una base de datos comprensible y estructurada, lo cual fue fundamental para el desarrollo y la implementación del modelo predictivo. Luego, el objetivo parcial sobre la extracción y creación de una fuente de datos que incluya todas las características requeridas y que pueda entrenarse con un modelo de predicción queda completamente satisfecho.

6.1.2 INTEGRACIÓN DE UNA PLATAFORMA EFICIENTE EN LA NUBE

La gestión y procesamiento de grandes volúmenes de datos requerían una infraestructura tecnológica avanzada. Por ello, se optó por la integración de una plataforma en la nube que ofreciera las capacidades necesarias para almacenar, procesar y analizar los datos de manera eficiente.

Se implementaron servicios en la nube que permitieron la escalabilidad y la flexibilidad del proyecto. La plataforma facilitó la ejecución de algoritmos de aprendizaje automático en un entorno optimizado, reduciendo tiempos de procesamiento y mejorando la accesibilidad y colaboración entre los miembros del equipo.

6.1.3 OBTENCIÓN DE UNA INTERFAZ GUI

Además, se desarrolló una interfaz gráfica de usuario (GUI) que, aunque su diseño inicial no cumplió con todas las expectativas y funcionalidades deseadas, permite a los usuarios interactuar con el modelo de predicción. Esta GUI conecta el proyecto almacenado en la nube con los usuarios, proporcionando una manera accesible de consultar y visualizar las predicciones de llegada de los aviones. A pesar de sus limitaciones actuales, la interfaz es un paso importante hacia la integración completa del sistema, facilitando su uso práctico y demostrando la viabilidad del modelo predictivo en un entorno real.

6.2 RESULTADOS GENERALES

6.2.1 RENDIMIENTO DEL MODELO PREDICTIVO - ANÁLISIS DE LA IMPORTANCIA DE LAS VARIABLES

La gráfica de la figura 21 adjunta muestra la importancia de cada variable en el modelo de regresión lineal utilizado para la predicción. Cada barra en la gráfica representa el coeficiente asignado a cada característica, indicando su peso relativo en el modelo predictivo. Las

variables con coeficientes más altos tienen una mayor influencia en la predicción de la hora de llegada de los aviones.

El análisis de la importancia de las variables reveló que ciertas características intrínsecas, como la velocidad, y meteorológicas como las precipitaciones el viento, tienen un impacto significativo en las predicciones. Este desglose tecnológico proporciona una comprensión más profunda de cómo cada variable contribuye al rendimiento del modelo, permitiendo una mejor interpretación y futuras mejoras.

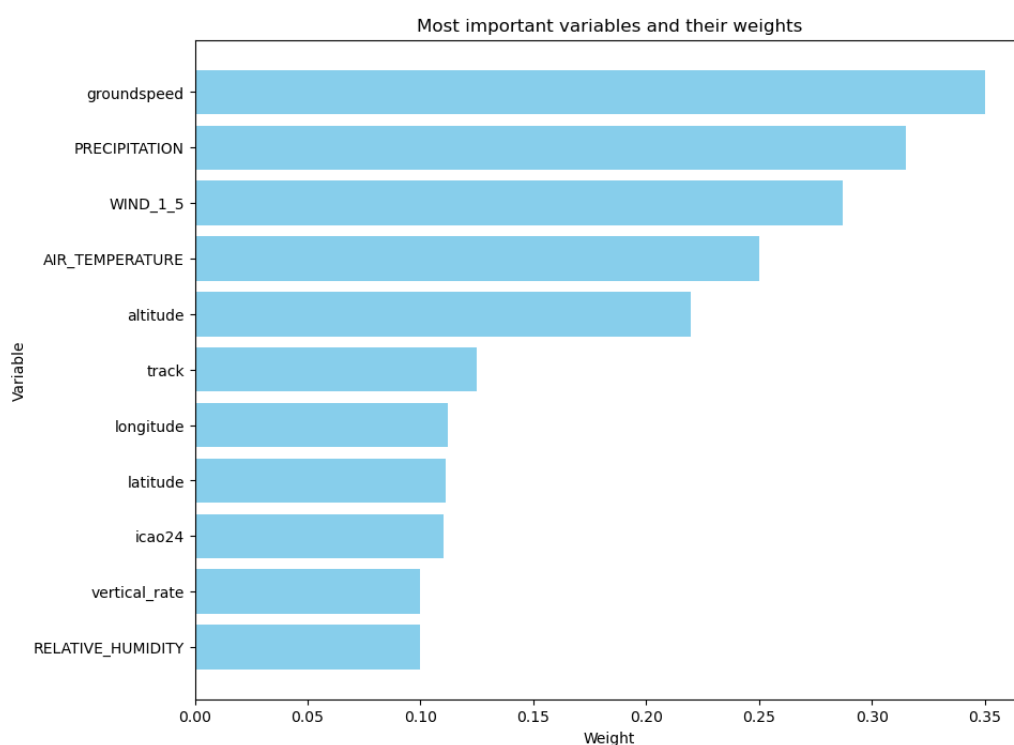


Figura 21: Gráfica de barras del peso de las variables más importantes en el modelo

6.2.2 RENDIMIENTO DEL MODELO PREDICTIVO – ERROR CUADRÁTICO MEDIO

El rendimiento del modelo se evaluó utilizando el error cuadrático medio (RMSE), que se calculó sucesivamente con conjuntos de test tomados en diferentes momentos. Los resultados del RMSE variaron entre 522 y 1350 segundos, dependiendo del conjunto de

datos utilizado para la prueba. Como se muestra en la captura de la figura 22, los valores obtenidos son los siguientes.

- **Conjunto de Test 1:** RMSE = 750 segundos
- **Conjunto de Test 2:** RMSE = 522 segundos
- **Conjunto de Test 3:** RMSE = 1350 segundos
- **Conjunto de Test 4:** RMSE = 1012 segundos

```
RMSE for test 1 is 750 seconds
RMSE for test 2 is 522 seconds
RMSE for test 3 is 1350 seconds
RMSE for test 4 is 1012 seconds
```

Figura 22: Salida del RMSE para los distintos conjuntos de test

Cabe destacar que los valores más bajos obtenidos son bastante positivos ya que la diferencia ronda los 10 minutos. Sin embargo, el caso no es ideal en los dos últimos tests. Esta disparidad en los valores del RMSE puede atribuirse a varios factores. En primer lugar, las diferencias en las condiciones meteorológicas y el volumen de tráfico aéreo en los distintos momentos de las pruebas pueden haber afectado la precisión del modelo. Por ejemplo, los lunes y viernes suelen ser los más congestionados en muchas regiones, debido a la combinación de vuelos de llegada y salida, especialmente en aeropuertos principales donde hay un alto volumen de vuelos nacionales e internacionales. Además, esto se acentúa especialmente en las primeras horas de la mañana y al final de la tarde, cuando los vuelos comerciales y de carga son más frecuentes. Luego dependiendo del momento de toma del conjunto de test, el tráfico aéreo, y en particular el tiempo de vuelo, es propenso a comportarse de manera más o menos predecible. Esto podría haber introducido variaciones en los datos debido a las diferencias en las condiciones locales, como el tráfico aéreo y las condiciones meteorológicas en comparación con otros aeropuertos o regiones.

En segundo lugar, esta ligera disparidad en los valores del error puede deberse a otro factor clave relacionado con la capacidad limitada de realización del propio proyecto. Me refiero a la localización única del receptor ADS-B. Los datos de prueba se recopilaron utilizando un receptor ADS-B ubicado en un único punto, específicamente en la ciudad de Austin. Lo ideal

habría sido colocar antenas y receptores en varias zonas del territorio, pero la realidad más sensata, dados los recursos disponibles, era montar un único receptor en el mismo lugar en donde se realizó el resto del trabajo. Estas diferencias geográficas y operativas entre la ubicación del receptor ADS-B y los aeropuertos utilizados para el entrenamiento pueden haber contribuido a la variabilidad observada en los resultados de RMSE entre los conjuntos de prueba.

En conclusión, el desarrollo y análisis del modelo de predicción para la estimación de la hora de llegada de aviones ha sido un proceso exhaustivo y fructífero. Se logró superar los desafíos iniciales de obtener y preparar un dataset completo, asegurando la inclusión de variables relevantes tanto intrínsecas como externas al avión. La integración de una plataforma en la nube proporcionó la infraestructura necesaria para el procesamiento eficiente de datos y la ejecución de algoritmos de aprendizaje automático, facilitando la escalabilidad y optimización del proyecto.

El análisis detallado de la importancia de las variables destacó cómo características como la velocidad del avión y las condiciones meteorológicas influyen significativamente en las predicciones. Asimismo, la evaluación del modelo mediante el error cuadrático medio (RMSE) reveló variaciones en la precisión predictiva, atribuibles a factores como las condiciones locales de tráfico aéreo y meteorológicas, así como a limitaciones geográficas en la recolección de datos de prueba.

Estos hallazgos proporcionan una base sólida para la continuación del trabajo, enfocándose en la refinación del modelo predictivo y la implementación de estrategias para mejorar la precisión y la robustez en diferentes escenarios operativos del tráfico aéreo.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

Comentar las conclusiones del proyecto, destacando lo que se ha hecho, dejando claros qué objetivos se han cubierto y cuáles son las aportaciones hechas.

7.1 CONCLUSIONES

En conclusión, este proyecto se diseñó con un enfoque sistemático para abordar la predicción precisa de tiempos de llegada de aeronaves mediante el uso de datos ADS-B en tiempo real y datos meteorológicos históricos. La estructura del proyecto se dividió en varios subsistemas manejables, cada uno de los cuales desempeñó un papel vital en el éxito del proyecto. Desde la recolección y procesamiento de datos hasta el entrenamiento del modelo y la configuración del hardware, se implementaron estrategias cuidadosas para garantizar la cohesión y la eficiencia en cada etapa del desarrollo. A través del análisis detallado de los resultados obtenidos, se han alcanzado varios hitos significativos que contribuyen al avance en la gestión y planificación de operaciones aéreas.

Uno de los principales logros del proyecto fue la integración de datos de múltiples fuentes, como OpenSky y NOAA, y su transformación en un conjunto de datos utilizable para el aprendizaje automático. Este proceso incluyó la conversión de formatos de fecha, la asignación de estaciones meteorológicas a aeropuertos específicos y la sincronización temporal de los datos para facilitar la fusión de bases de datos. Este enfoque meticuloso permitió crear una base de datos robusta y precisa que fue fundamental para el entrenamiento del modelo de predicción y que puede reciclarse para futuras investigaciones en el ámbito de la predicción del sector aéreo.

Por otro lado, se incluyeron nuevas herramientas tecnológicas y de machine learning que supusieron un avance muy valorado por la empresa Boeing. El modelo de predicción se entrenó utilizando XGBoost, una biblioteca de árboles de decisión potenciados por gradiente, y se evaluó mediante métricas de error cuadrático medio utilizando Sci-kit learn.

Además, se añadieron características adicionales al conjunto de datos, como las distancias y velocidades en los ejes x e y, y el coseno de la trayectoria, lo que mejoró la precisión del modelo. Todo ello contribuyó a la efectividad del modelo, permitiendo predicciones continuas y precisas.

La configuración de hardware incluyó la implementación de un receptor ADS-B para la captura de datos en vivo de aeronaves. La antena A3 Jetvision y el receptor RTL-SDR V3 R860 se utilizaron para captar y procesar los datos, que luego se enviaron a un bucket AWS S3 para su almacenamiento y análisis. Esta infraestructura de hardware y almacenamiento en la nube permitió una captura y procesamiento eficientes de los datos en tiempo real.

La interfaz gráfica de usuario (GUI) desarrollada facilitó la interacción con el sistema, permitiendo a los usuarios visualizar datos de vuelos en tiempo real y obtener predicciones de tiempos de llegada. Esta GUI, desarrollada con React y Flask, ofreció una herramienta intuitiva y accesible para los usuarios, mostrando información relevante de manera clara y concisa.

Este proyecto no solo mejora las soluciones existentes en el mercado al proporcionar predicciones más precisas y en tiempo real, sino que también sirve como un broche final para la carrera de ingeniería de telecomunicaciones. Los aspectos tecnológicos y técnicos del proyecto, como la integración de múltiples fuentes de datos, el uso de técnicas avanzadas de aprendizaje automático y la implementación de infraestructura de hardware, demuestran un alto nivel de competencia técnica. Además, la gestión del proyecto involucró la coordinación de un equipo, la colaboración con una empresa externa y el cumplimiento de plazos, lo que subraya la capacidad de gestión y organización en proyectos de gran envergadura.

El proyecto también tiene un impacto social significativo, ya que mejora la eficiencia y la seguridad en la gestión del tráfico aéreo, contribuyendo al desarrollo sostenible y económico. La precisión en la predicción de tiempos de llegada puede optimizar las operaciones aeroportuarias, reducir el consumo de combustible y disminuir las emisiones de carbono, favoreciendo así un desarrollo más sostenible.

En resumen, este proyecto ha cumplido total o parcialmente con sus objetivos, proporcionando una solución integral y avanzada para la predicción de tiempos de llegada de aeronaves. La combinación de técnicas avanzadas, gestión eficaz y colaboración ha resultado en un producto que no solo mejora las soluciones actuales, sino que también demuestra la culminación exitosa de una carrera en ingeniería de telecomunicaciones, resaltando tanto las habilidades técnicas como las capacidades de gestión y colaboración.

7.2 TRABAJOS FUTUROS

Se identificaron desafíos significativos relacionados con las variaciones en las condiciones locales de tráfico aéreo y meteorológicas, así como limitaciones geográficas en la recolección de datos. Estos hallazgos subrayan la necesidad de estrategias específicas para mejorar la robustez y precisión del modelo en diferentes contextos operativos. Luego, para posibles trabajos futuros y mejoras del sistema de predicción, y considerando la naturaleza dinámica de los viajes aéreos, es fundamental refinar y expandir continuamente las capacidades del sistema. Aquí hay dos estrategias clave que pueden fortalecer significativamente la efectividad del sistema.

7.2.1 AMPLIACIÓN DEL ESTUDIO A NUEVOS AEROPUERTOS DESTINO

El proyecto llevado a cabo solo se enfoca en los aeropuertos más importantes de Estados Unidos, como requisito de la empresa colaboradora Boeing. Estos aeropuertos son:

- Aeropuerto Internacional de Dallas Fort Worth
- Aeropuerto Internacional de Miami
- Aeropuerto JFK de Nueva York
- Aeropuerto Internacional O'Hare de Chicago
- Aeropuerto Internacional de Denver
- Aeropuerto Internacional de Los Ángeles
- Aeropuerto Internacional de Atlanta

Tratándose de los aeropuertos que gestionan el mayor porcentaje de vuelos diarios en todo el país, dar con una solución para estos ya supondría un gran avance en la eficiencia del sector, razón del éxito del propio proyecto. Sin embargo, escalar el trabajo, generalizar los progresos a nuevos aeropuertos y permitir que otros actores compren la solución y/o hagan uso del modelo de predicción requiere de una ampliación de los datos de entrada incluyendo nuevos aeropuertos de estudio. En resumen, a medida que la gestión del tráfico aéreo evoluciona, incorporar datos de una variedad más amplia de aeropuertos se vuelve esencial. Cada aeropuerto presenta desafíos operativos únicos influenciados por factores como la ubicación geográfica, los patrones climáticos locales y el volumen de tráfico. Al expandir el conjunto de datos para incluir una gama más diversa de aeropuertos, el sistema de predicción puede adaptarse mejor a las complejidades de varios entornos operativos.

Incluir datos de aeropuertos adicionales enriquece la comprensión del sistema sobre las características operativas diversas. Entrenar el modelo con una gama más amplia de entornos aeroportuarios mejora su adaptabilidad, permitiéndole captar matices específicos de cada ubicación. Este conjunto de datos ampliado permite al modelo de predicción generar estimaciones más precisas en diferentes regiones y escenarios. Por ejemplo, los aeropuertos situados en regiones propensas a condiciones climáticas adversas o que experimentan altos volúmenes de tráfico pueden requerir algoritmos predictivos especializados adaptados a esas condiciones. Al incorporar datos de una variedad diversa de aeropuertos, el sistema está mejor equipado para manejar tales complejidades, mejorando en última instancia sus capacidades predictivas.

7.2.2 ESTABLECIMIENTO DE UNA RED DE RECEPTORES Y ANTENAS

El proyecto llevado a cabo tiene un componente de hardware muy importante para la obtención de resultados. Se trata de un receptor ADS-B que recibe registros en tiempo real de los aviones en vuelo. Por un lado, estos registros se almacenaron y tomaron como datos para evaluar la exactitud y precisión del modelo. Por otro lado, estos se insertan continuamente en la interfaz GUI para poder visualizar los aviones en vuelo en tiempo real a través de la pantalla. La disponibilidad de un único receptor ADS-B limita las capacidades

de estas dos últimas tareas pues solo puede recibir datos de una única ubicación en el tiempo. Por ello sería interesante crear una red de receptores y antenas colocadas cerca de los aeropuertos de estudio de tal forma que los datos de test provengan equitativamente de todos los aeropuertos y que los usuarios de la interfaz puedan acceder a vuelos en tiempo real en diversas localizaciones.

Resulta que los datos en tiempo real juegan un papel crucial en la predicción precisa de los tiempos de llegada de los vuelos. Para garantizar la disponibilidad de datos oportunos y completos, es fundamental establecer una red de receptores y antenas en múltiples aeropuertos. Esta red facilita una comunicación e intercambio de datos sin problemas, permitiendo que el sistema de predicción acceda a una gran cantidad de datos de vuelos en vivo desde diversas ubicaciones alrededor del mundo.

Desarrollar una red de receptores y antenas posicionadas estratégicamente en múltiples aeropuertos establece una infraestructura robusta para la comunicación global de datos. Al agregar datos en vivo de múltiples nodos, el sistema de predicción obtiene una visión integral y en tiempo real de las actividades de vuelo a nivel mundial. Este flujo de datos permite al sistema captar cambios repentinos en las trayectorias de vuelo, el rendimiento de las aeronaves y las condiciones atmosféricas, mejorando la precisión de las predicciones de tiempo de llegada. Además, en escenarios que involucran vuelos intercontinentales o configuraciones complejas del espacio aéreo, el sistema en red proporciona valiosos conocimientos sobre los patrones de vuelo globales, permitiendo que el modelo de predicción se adapte dinámicamente a las condiciones cambiantes. Al aumentar el modelo de predicción con datos provenientes de una infraestructura en red, la robustez y precisión del sistema se incrementan significativamente, allanando el camino para operaciones de transporte aéreo más eficientes a escala global.

Al implementar estas recomendaciones y explorar continuamente vías de mejora futura, el sistema de predicción puede evolucionar para satisfacer las demandas cambiantes de la

industria de la aviación, ofreciendo estimaciones confiables de los tiempos de llegada de los vuelos y contribuyendo a una mayor eficiencia y seguridad en el transporte aéreo.

Capítulo 8. BIBLIOGRAFÍA

- [1] “The OpenSky Network - Free ADS-B and Mode S data for Research”. The OpenSky Network - Free ADS-B and Mode S data for Research. Accedido el 2 de julio de 2024. [En línea]. Disponible: <https://opensky-network.org/>
- [2] NOAA, “National Centers for Environmental Information (NCEI),” Noaa.gov, Jun. 06, 2019. <https://www.ncei.noaa.gov/>
- [3] N. C. for E. I. NCEI), “Quality Controlled Datasets,” [www.ncei.noaa.gov](https://www.ncei.noaa.gov/access/crn/qcdatasets.html). <https://www.ncei.noaa.gov/access/crn/qcdatasets.html>
- [4] “Introducción a TensorFlow”. TensorFlow. Accedido el 2 de julio de 2024. [En línea]. Disponible: <https://www.tensorflow.org/learn?hl=es-419>
- [5] “How to Get your Aircraft Compliant with ADS-B Requirements”. PilotMall.com. Accedido el 2 de julio de 2024. [En línea]. Disponible: <https://www.pilotmall.com/blogs/news/what-do-you-need-to-do-to-get-your-aircraft-compliant-with-the-2020-ads-b-requirement>
- [6] Olive, Xavier and Morio, Jerome. “Trajectory clustering of air traffic ^ flows around airports”. Aerospace Science and Technology, 2019.
- [7] M. Shravan, R. Rakshit, P. Sanjana, B. K. Priya and N. Kumar, "RTL SDR ADS-B Data Analysis for Predicting Airports and ATS Routes," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-7, doi: 10.1109/INCET49848.2020.9154059.
- [8] G. Chen, J. Rosenow, M. Schultz, and O. Okhrin, “Using Open Source Data for Landing Time Prediction with Machine Learning Methods,” in 8th OpenSky Symposium 2020, in OpenSky. MDPI, Dec. 2020. doi: 10.3390/proceedings2020059005.
- [9] AeroTransport DataBank. “ATDB.aero - ICAO 24 bit addresses - Decode.” [aerotrtransport.org](https://www.aerotrtransport.org/html/ICAO_hex_decode.html). https://www.aerotrtransport.org/html/ICAO_hex_decode.html (Accessed Nov. 30, 2023).
- [10] “Get A Receiver,” [opensky-network.org](https://opensky-network.org/contribute/get-a-receiver). <https://opensky-network.org/contribute/get-a-receiver> (accessed Dec. 05, 2023).
- [11] Honeywell Aerospace. "AUTOMATIC DEPENDENT SURVEILLANCE-BROADCAST WILL TRANSFORM AIR TRAFFIC CONTROL: A white paper on DO-260 mandates, Honeywell solutions, and expected benefits."

- <https://aerospace.honeywell.com/content/dam/aerobt/en/documents/landing-pages/white-papers/DO-260-mandates-white-paper.pdf> (accessed November 3, 2023).
- [12] International Civil Aviation Organization. "Overview of Automatic Dependent Surveillance-Broadcast (ADS-B) Out." <https://www.icao.int/NACC/Documents/Meetings/2021/ADSB/P01-OverviewADSBOut-ENG.pdf> (accessed November 3, 2023).
- [13] "Title 47 Telecommunication Parts 0 to 19." Accessed: November 3, 2023. [Online]. Available: <https://www.govinfo.gov/content/pkg/CFR-2021-title47-vol1/pdf/CFR-2021-title47-vol1.pdf> (accessed November 3, 2023).
- [14] Bureau of Transportation Statistics, U.S. Department of Transportation. "Flight Delay Information." [Online]. Available: <https://www.bts.gov/topics/airlines-and-airports/flight-delay-information>
- [15] Bureau of Transportation Statistics, U.S. Department of Transportation. "Airline On-Time Statistics and Delay Causes." [Online]. Available: https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp
- [16] University of California, Berkeley. "Economic Impact of Flight Delays." [Online]. Available: <https://news.berkeley.edu/2023/06/15/economic-impact-of-flight-delays>

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

El desarrollo de un modelo de predicción sobre la hora de llegada de los aviones a los aeropuertos destino, utilizando características intrínsecas del avión, datos meteorológicos y rutas de aterrizaje seguidas por el piloto, se alinea significativamente con varios Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas. Este proyecto no solo representa un avance tecnológico, sino también un paso importante hacia un futuro más eficiente y sostenible en el sector de la aviación.

En primer lugar, el proyecto contribuye al ODS 9: Industria, Innovación e Infraestructura. Según la Asociación Internacional de Transporte Aéreo (IATA), los retrasos en los vuelos cuestan a la industria aérea aproximadamente 25 mil millones de dólares al año. Al mejorar la precisión en la predicción de la hora de llegada de los aviones, se optimiza la gestión del tráfico aéreo y se reducen los tiempos de espera, tanto en el aire como en tierra. Esta optimización mejora la eficiencia operativa de los aeropuertos y las aerolíneas, lo que se traduce en una utilización más racional y eficaz de las infraestructuras existentes. Además, al incorporar datos meteorológicos y de rutas de aterrizaje, el modelo promueve la innovación tecnológica y el uso de información avanzada para la toma de decisiones, lo que puede incentivar nuevas inversiones en tecnología y mejoras en la infraestructura aeroportuaria.

Asimismo, el proyecto está en línea con el ODS 13: Acción por el Clima. La aviación es responsable de aproximadamente el 2% de las emisiones globales de dióxido de carbono, y la reducción de los tiempos de vuelo y espera puede contribuir a disminuir el consumo de combustible y, por ende, las emisiones de CO₂. Estudios han demostrado que una mejora del 1% en la eficiencia del combustible puede ahorrar alrededor de 200 millones de galones de combustible de aviación anualmente. Al prever mejor las condiciones meteorológicas y

ajustar las rutas en consecuencia, el modelo puede ayudar a evitar rutas ineficientes y tiempos prolongados en el aire, reduciendo así la huella de carbono de cada vuelo y el malgasto de recursos energéticos no renovables.

El ODS 11: Ciudades y Comunidades Sostenibles también se ve reflejado en este proyecto. Una mejor gestión de las llegadas de aviones puede reducir el ruido y la congestión en los alrededores de los aeropuertos, mejorando la calidad de vida de las comunidades cercanas. Un estudio realizado por el Instituto de Investigación del Transporte en la Universidad de California encontró que los retrasos y las ineficiencias en el aterrizaje contribuyen significativamente al ruido ambiental. La precisión en los horarios de llegada puede facilitar la planificación del transporte terrestre, mejorando la conectividad y la eficiencia del transporte público y privado que se coordina con los horarios de los vuelos. Además, una mejor previsión de los tiempos de llegada puede ayudar a las autoridades locales a gestionar el tráfico vehicular en las áreas cercanas a los aeropuertos, reduciendo los embotellamientos y la contaminación.

Por último, el proyecto tiene un impacto social positivo en el ODS 8: Trabajo Decente y Crecimiento Económico. Al aumentar la eficiencia operativa de las aerolíneas y aeropuertos, se pueden generar ahorros significativos y mejorar la competitividad del sector. Un informe de la Organización de Aviación Civil Internacional (OACI) estima que cada 1% de aumento en la eficiencia operativa puede traducirse en un ahorro de aproximadamente 1.5 mil millones de dólares para la industria. Esto no solo fomenta el crecimiento económico, sino que también puede traducirse en mejores condiciones laborales para los empleados del sector, desde los controladores aéreos hasta el personal de tierra. La reducción del estrés operativo y la mejora en la previsión de tareas contribuyen a un entorno de trabajo más seguro y satisfactorio.

En conclusión, el desarrollo de un modelo de predicción sobre la hora de llegada de los aviones es un avance significativo que se alinea con varios Objetivos de Desarrollo Sostenible. Promueve la eficiencia y la innovación en la infraestructura de transporte aéreo, contribuye a la reducción de emisiones y a la lucha contra el cambio climático, mejora la

sostenibilidad de las ciudades y comunidades, y apoya el crecimiento económico y el empleo digno. Este proyecto, por tanto, no solo representa un avance tecnológico, sino que también tiene un impacto positivo en la sociedad y el medio ambiente, avanzando hacia un futuro más sostenible y equitativo.

ANEXO II: TABLA DE ASIGNACIÓN DE TAREAS EN EL GRUPO

Data team:

Name	Planned Contributions	Areas of Expertise	Course Work
Vikram Padmanaban	Finding appropriate datasets and data cleaning	Pandas, Data Engineering	Prior intern experience, ECE 461L, ECE 351K
Randy Tan	Help with data cleaning and data scraping	Pandas	ECE 460J
Amaia Rotaeché	Supervisor, ingestion, cleaning, and analysis	Data Pipeline Engineering, Pandas, Querying	Analytics Internships, Python, SQL, R, machine learning courses

Machine learning team:

Name	Planned Contributions	Areas of Expertise	Course Work
Mahek Kakkar	Help develop prediction lading model based on factors	Data science, neural networks, regression analysis	ECE 460J
Randy Tan	Help develop models	Data science, model tuning	ECE 460J
Dasol Ahn	Help develop models	Neural networks, data science	ECE 371Q
Amaia Rotaeché	Supervisor, Help develop prediction	Machine learning, regression models, neural networks	ECE 461P + Machine learning

	model and visualization of data	PowerBi/Python/Tableau (visualization tools)	course at my home university (ICAI-Madrid)
--	---------------------------------	--	--

Hardware team:

Name	Planned Contributions	Areas of Expertise	Course Work
Miguel González	Finding compatible ADS-B receivers, setuping the receiver, and collecting data from specific receiver	Cadence Virtuoso, LTSpice, Oscilloscopes	ECE 445L, ECE 438, ECE 460R
Amaia Rotaeché	Supervisor, assembly, design and integration	Circuits, digital systems, LTSpice, Oscilloscopes, Electronic engineering	Courses in microprocessors, electronics, electric circuits

ANEXO III: DIAGRAMA GANTT DE PLANIFICACIÓN TEMPORAL

