Creación del Modelo

Diego Alonso, Telmo Aracama, Héctor García, Ignacio Gutierrez, Mario López, Carlos Mantilla, Pablo Rodríguez

23 de febrero de 2025

Índice

1. Reprocesamiento de los Datos: Corrección de Errores y Mejora del Preprocesamiento

Tras la identificación de varios errores durante el preprocesamiento de los datos en la primera entrega del proyecto, se ha decidido realizar una nueva versión de este proceso, con el objetivo de corregir dichos errores y mejorar la calidad de los datos para el análisis posterior. Este reprocesamiento se ha estructurado en cuatro pasos clave, cada uno de los cuales aborda aspectos específicos del flujo de trabajo de datos para asegurar una mayor precisión y consistencia en los resultados finales.

El reprocesamiento implica una revisión exhaustiva de las fases iniciales del análisis de datos, asegurando que se gestionen adecuadamente las variables relevantes, se eliminen posibles inconsistencias y se incorporen las transformaciones necesarias para optimizar la calidad de los datos. Este proceso tiene como objetivo no solo subsanar los errores previos, sino también ofrecer una base más sólida sobre la cual desarrollar modelos predictivos más precisos y fiables.

Cada uno de los pasos del reprocesamiento ha sido diseñado para abordar específicamente las áreas en las que se encontraron deficiencias, y se han implementado nuevas validaciones y controles de calidad para evitar la recurrencia de los errores anteriores.

El principal problema que surgió durante el preprocesamiento fue la enorme cantidad de datos que había que manejar. Para abordar esta dificultad, se optó por un enfoque progresivo: en lugar de procesar todos los datos de una sola vez, se fueron generando columnas de forma escalonada. Se priorizó la obtención inicial de aquellas columnas que permitieran filtrar grandes volúmenes de datos, reduciendo así progresivamente el tamaño del conjunto. Una vez completado el filtrado principal, se procedió a calcular el resto de las columnas necesarias. Este enfoque nos ha permitido reducir considerablemente el tiempo de procesado de los datos y utilizar pandas en todo el proceso.

1.1. Primer Paso: Decodificación y Extracción de Variables Básicas

En esta primera fase se realiza la decodificación de los mensajes originales recibidos en formato Base64, transformándolos a representación hexadecimal. Una vez decodificados, se aplican filtros de validación:

- Se calcula el CRC (Cyclic Redundancy Check) de cada mensaje para verificar su integridad. Solo se conservan aquellos mensajes cuyo CRC es igual a 0.
- Se analiza el *typecode* de cada mensaje. Los mensajes con *typecode* = -1, que no contienen información útil, son descartados.

Después de este filtrado inicial, se extraen campos relevantes para el análisis posterior:

- Identificador de aeronave (*icao*)
- Indicador de paridad (*oe_flag*)
- Altitud (*altitude*)

- Velocidad horizontal (*speed*), ángulo de desplazamiento (*angle*) y tasa de ascenso o descenso (*vertical_rate*)

Para optimizar el almacenamiento y el procesamiento posterior, se realiza la conversión de tipos de datos:

- Las fechas se transforman a formato `datetime`.
- Variables categóricas y booleanas se ajustan a tipos específicos (`category`, `boolean`) y los valores numéricos se reducen (`unsigned integers`).

Este procesamiento se realiza de manera paralela utilizando todos los núcleos disponibles del sistema, debido al volumen elevado de datos diarios. Finalmente, los datos decodificados y procesados se almacenan en formato `.parquet` comprimido, facilitando su acceso rápido y eficiente en los siguientes pasos del proyecto.

1.2. Segundo Paso: Filtrado de Mensajes y Construcción de Vuelos

Una vez realizada la decodificación y la eliminación de mensajes inválidos, se procede a un proceso de filtrado y estructuración de los datos, cuyo objetivo es depurar aún más la información y construir trayectorias de vuelo coherentes asociadas a operaciones de despegue y aterrizaje. Este procesamiento se realiza día a día, para capturar de forma precisa todas las operaciones realizadas por cada aeronave durante una jornada completa.

En primer lugar, se calcula la posición geográfica de cada mensaje decodificado. Para ello, se utiliza el valor del campo `oe_flag`, que indica el tipo de mensaje ADS-B (Even/Odd) necesario para aplicar el algoritmo de decodificación de posiciones conocido como *Compact Position Reporting* (CPR). A partir de esta reconstrucción, se obtiene para cada mensaje una latitud y longitud estimada. Además, se calcula si el avión asociado al mensaje se encuentra en la pista o no, basándose en su ubicación respecto a las zonas de operación aeroportuaria.

A continuación, se realiza un primer filtrado espacial, eliminando todos aquellos mensajes cuya posición esté considerablemente alejada del aeropuerto de interés (en este caso, Adolfo Suárez Madrid-Barajas). Esto permite reducir notablemente la cantidad de datos que deben ser procesados en etapas posteriores, enfocando el análisis en operaciones realmente relevantes.

Posteriormente, se lleva a cabo un filtrado más estricto a nivel de aeronaves: se eliminan todos los mensajes correspondientes a aviones que no atraviesan ninguna pista en ningún momento del día. Este criterio permite eliminar trayectorias de aeronaves que, si bien pueden estar próximas al aeropuerto, no realizan maniobras de despegue ni aterrizaje en él (por ejemplo, sobrevuelos o vuelos de proximidad).

Una vez realizada esta depuración, se introduce la noción de *flight_id*, una columna adicional que permite identificar de forma única cada evento de despegue o aterrizaje realizado por una aeronave durante el día. Esto es necesario porque una misma aeronave puede realizar múltiples operaciones en un mismo día (por ejemplo, varios vuelos de

llegada o salida).

Con los eventos de vuelo estructurados, se realiza una clasificación específica: se filtran y conservan únicamente aquellos eventos que correspondan a despegues. La distinción se basa en varios criterios, entre ellos:

- La evolución de la altitud a lo largo de la trayectoria (mayor que 0 para despegues).
- La velocidad relativa respecto a la pista. (en algún moomento debe ser 0)
- La diferencia de altitud entre mensajes consecutivos.

Estos criterios permiten identificar trayectorias ascendentes que comienzan en el área de pista, propias de maniobras de despegue, y eliminar eventos de aterrizaje o movimientos que no culminan en despegues.

Una vez obtenido el subconjunto de despegues, se enriquece la información asociada a cada vuelo añadiendo nuevos atributos relevantes, tales como:

- Si la aeronave se encuentra en un *holding point* (puntos de espera).
- Hora de llegada del avión al punto de espera
- El tiempo que lleva el avión esperando en el punto de espera.
- El momento temporal (hora exacta) y la posición geográfica (latitud y longitud) donde ocurre el despegue (en el caso de trayectorias de despegue complementarias).
- La distancia desde el lugar donde se encuentra el avión y el lugar donde se ha detectado que comienza el despegue.
- Otras columnas adicionales que caracterizan mejor el estado de la aeronave y su interacción con las infraestructuras aeroportuarias.

De esta manera, al finalizar este segundo paso, se dispone de un conjunto de vuelos depurado, donde cada trayectoria corresponde a un despegue correctamente detectado y estructurado, incluyendo información detallada sobre su posición, dinámica, y estado operativo en relación al aeropuerto.

1.3. Tercer Paso: Filtrado de Aviones que Paran en un Punto de Espera

Tras haber identificado y filtrado las trayectorias de aterrizaje en pasos anteriores, se realiza un procesamiento adicional orientado a seleccionar únicamente aquellas aeronaves que, en su aproximación final, pasan y se detienen en un punto de espera (*holding point*) antes de proceder a su operación. Esta selección resulta crítica, ya que los tiempos de espera y la ocupación de los puntos de espera son variables relevantes para analizar la dinámica operativa en el aeropuerto. Además, los despegues seleccionados en esta fase son los que usaremos posteriormente para el entrenamiento de modelos.

El procedimiento se estructura de la siguiente forma:

1.3.1. Identificación de aeronaves detenidas en puntos de espera

A partir de los mensajes procesados previamente, se filtran únicamente aquellos registros donde:

- La aeronave tenga asignado un `holding_point_id` (indicando su paso por un punto de espera conocido).
- La velocidad registrada (`speed`) sea igual a cero, lo que evidencia que la aeronave está detenida en ese lugar.

Este filtrado asegura que sólo se consideren aquellas situaciones en las que la aeronave realmente ha parado en el punto de espera, descartando trayectorias donde únicamente haya pasado por las inmediaciones sin detenerse.

1.3.2. Extracción de información adicional meteorológica y operacional

Para enriquecer los registros filtrados, se extraen características adicionales directamente a partir de los mensajes ADS-B, decodificando la información disponible en el mensaje hexadecimal (`msg_hex`). En concreto, se obtiene:

- Velocidad del viento (`wind_speed`) y dirección del viento (`wind_dir`), empleando la decodificación de mensajes BDS 4,4.
- Información sobre `wake_vortex_category` (`wake_vortex`), temperatura (`temp`) y `wind_shear` (`wind_shear`), empleando la decodificación de mensajes BDS 4,5.

La extracción de estos datos permite disponer de un conjunto de variables adicionales que pueden influir en la operación de despegue o aterrizaje de una aeronave, especialmente bajo condiciones meteorológicas adversas.

1.3.3. Normalización y optimización del conjunto de datos

Una vez enriquecido el conjunto de datos, se lleva a cabo una transformación destinada a:

- Convertir campos temporales (`timestamp`) a formatos de fecha y hora.
- Añadir nuevas columnas derivadas de la fecha, como mes, día de la semana, día del mes y hora decimal del día.
- Convertir diversas columnas categóricas (`icao`, `holding_point_id`, `runway`, `wake_vortex`, etc.) a tipos optimizados para reducir el consumo de memoria.
- Asignar tipos de datos compactos (`float32`, `uint8`, `category`) a las variables numéricas y categóricas.

Este proceso de normalización permite optimizar el almacenamiento y la eficiencia de las etapas de modelado y análisis posteriores.

1.3.4. Procesamiento masivo diario

El procesamiento se organiza de manera secuencial día a día, siguiendo ventanas temporales configuradas por rangos de días (`init`, `end`). Para cada archivo diario, se realiza la carga de los datos, la aplicación del filtrado descrito y la extracción de variables adicionales. Finalmente, los resultados se consolidan en un archivo único en formato parquet, dividido por rangos de fechas para facilitar su gestión posterior.

De esta manera, al finalizar el tercer paso, se dispone de un conjunto de aeronaves que no solo han realizado una operación de despegue, sino que también han pasado y se han detenido en un punto de espera, complementadas con información ambiental y operacional extraída directamente del mensaje ADS-B.

1.4. Cuarto Paso: Procesamiento y análisis de vuelos mediante la incorporación de características adicionales sobre el tráfico aéreo

En este paso, se realiza un análisis detallado de los vuelos en función de su relación con los despegues previos, el tráfico cercano, y las condiciones de la pista de despegue. La información se obtiene a partir de la base de datos de vuelos de despegues obtenida como resultado de la fase 2 y se complementa con características temporales y geoespaciales. A continuación, se describen las principales funciones y actividades que se llevan a cabo:

1.4.1. Obtención de información sobre despegues previos

Se analizan los despegues previos al vuelo actual en un rango de tiempo determinado (por ejemplo, 5, 10, 20, 30, 45, y 60 minutos) para capturar el número de despegues y calcular la media de la diferencia de tiempo entre ellos. Esto ayuda a comprender la dinámica del tráfico aéreo en un período cercano al vuelo en cuestión.

1.4.2. Identificación de la última aeronave que haya despegado en la misma pista

Utilizando los mensajes ADS-B hexadecimales (`msg_hex`), se determina el tipo de aeronave que realizó el último despegue antes del vuelo actual. Esto es crucial para modelar los efectos del vórtice de estela, que afecta la seguridad y la eficiencia de los vuelos cercanos. También se obtiene hace cuanto tiempo se realizó ese último despegue.

1.4.3. Cálculo de distancias y tiempos de separación

Se implementa la fórmula de Haversine para calcular la distancia geográfica entre las aeronaves, basada en las coordenadas de latitud y longitud. Esto se utiliza para determinar la proximidad de otros aviones con respecto a la pista de despegue, ayudando a identificar posibles riesgos o congestión en el área de despegue.

1.4.4. Evaluación del tráfico cercano

Se analiza el tráfico cercano que podría influir en el despegue del avión en cuestión. Este análisis se realiza considerando tres categorías de aeronaves:

- **Aviones en punto de espera (Holding):** Se identifican aquellos que están esperando para despegar en el área de holding cerca de la pista.
- **Aviones en la pista de despegue:** Se determina si otros aviones ya están utilizando la pista de despegue y si están dentro del rango temporal crítico.
- **Aviones en ruta hacia la pista de despegue:** Se evalúa si hay aeronaves que están en camino hacia la pista de despegue antes que el avión en cuestión, lo cual podría afectar su tiempo de despegue.

1.4.5. Procesamiento de los datos por vuelo

Para cada vuelo, se agrupan los datos por identificador de vuelo y se procesan de manera individual. Se realiza un análisis exhaustivo para cada vuelo, utilizando funciones aplicadas a cada grupo de datos, lo que permite una evaluación de los vuelos en función de las variables de interés (tiempo de despegue, tráfico cercano, tipo de aeronave, etc.).

1.4.6. Generación de archivos de salida

Después de procesar los vuelos y obtener las nuevas características, los resultados se almacenan en archivos `.parquet`, lo que permite un acceso eficiente a los datos procesados para su posterior análisis.

Este proceso permite incorporar una gran cantidad de variables dinámicas que podrían afectar la seguridad y la eficiencia de las operaciones de despegue, como la proximidad de otros aviones, la congestión de la pista y los efectos de los vórtices de estela. Estos datos son fundamentales para desarrollar modelos predictivos que mejoren la toma de decisiones en la gestión del tráfico aéreo y la optimización de las operaciones de despegue.

1.5. Resultados del procesado

Como ya se ha comentado todo el pipeline de procesado se utilizó exclusivamente pandas. En todas las fases del procesado se paralelizó el proceso para que usase las 20 CPUs del ordenador con la biblioteca de python multiprocessing. En la primera fase, la decodificación, la paralelización se realiza a nivel de dia, es decir, dentro de un mismo dia cada worker procesa una hora distinto.

En las fases 2 y 4 —filtrado de mensajes y construcción de despegues, y análisis de vuelos mediante la incorporación de características adicionales sobre el tráfico aéreo y despegues previos— el procesamiento se realiza de forma independiente para cada `icao`. De este modo, la paralelización se lleva a cabo a nivel de `icao`: cada *worker* trabaja sobre un identificador distinto y, al finalizar el procesamiento, se concatenan todos los resultados obtenidos.

Los resultados en tiempo de ejecución de todo este pipeline de procesado han sido bastantes buenos. Tras todo el proceso se ha obtenido un conjunto de datos de 160.000 filas y 48 columnas que contienen información de 5895 vuelos. Pueden parecer pocos pero cabe destacar que en 3º paso, donde se filtran los aviones que no paran en el punto de espera, se pierden una gran cantidad de vuelos. Más concretamente, antes de realizar ese filtrado, teníamos un total de 46276 vuelos, pero un gran número de esos vuelos no

pasan por los puntos de esperea.

En tiempo de ejecución, los resultados del procesamiento de los 3 meses de datos y 130GB han sido los siguientes. Se han necesitado un total de **4 horas**

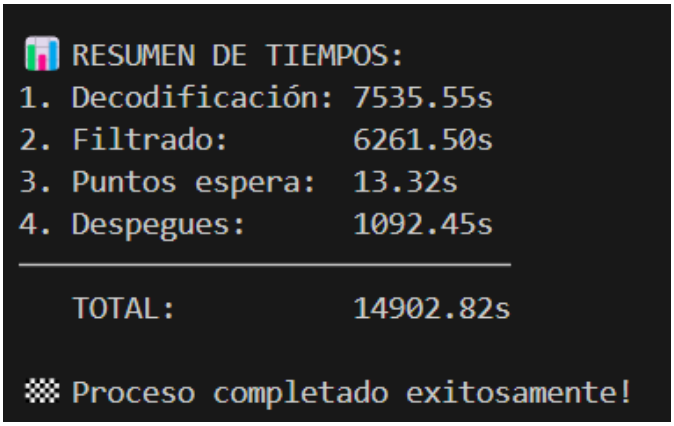

Figura 1: Tiempo de ejecución de cada fase del procesamiento.

2. Análisis Exploratorio de Datos

Ahora realizaremos un análisis de las variables qué consideramos más relevantes para el modelo.

2.1. Variables categóricas

Empezaremos con un breve análisis de algunas variables categóricas del conjunto de datos como la pista donde despega el avión, que holding point se está ocupando y el tipo del avión.

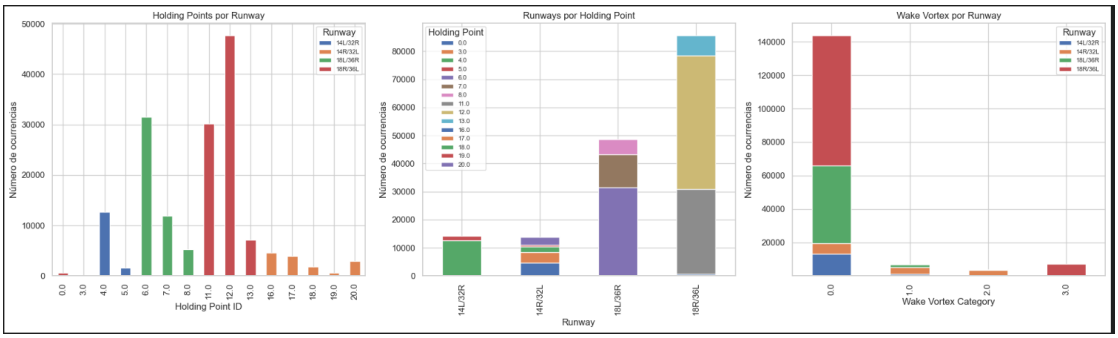

Figura 2: Diagramas de barras de variables categóricas

En el primer gráfico podemos ver que puntos de espera son más usados. Hay un claro desbalance en los datos y hay puntos de espera que no se usan prácticamente. Además podemos ver que cada punto de espera es específico para una pista de aterrizaje concreta

y que dentro de los puntos de espera para una misma pista, hay algunos que se utilizan considerablemente más que los otros.

En la segunda figura observamos un diagrama de barras que muestra cuantas veces se utiliza cada pista y cuantas veces es utilizada para cada punto de espera. Podemos ver un claro desbalance en estos datos ya que la pista **18R/36L** se utiliza casi 8 veces más que 14L/32R y 14R/32L, y casi el doble que 18L/36L. Esto puede suponer un problema a la hora de realizar el modelo ya que no dispondremos de suficientes datos para 3 de las pistas y crearña un claro sesgo hacia la pista 18R/36L

Por último vemos el número de aviones de cada tipo según tamaño, 0-3 en orden de tamaño, y en que pistas despegan cada uno. De nuevo vemos un claro sesgo de la población ya que la gran mayoría de los datos son de tipo 0 (tamaño pequeño).

2.2. Variables temporales

Para tratar de analizar el motivo de la descompensación de usos de pistas, hemos hecho una análisis de las operaciones realizadas a lo largo de distintos días y ver si sucede algo anómalo.

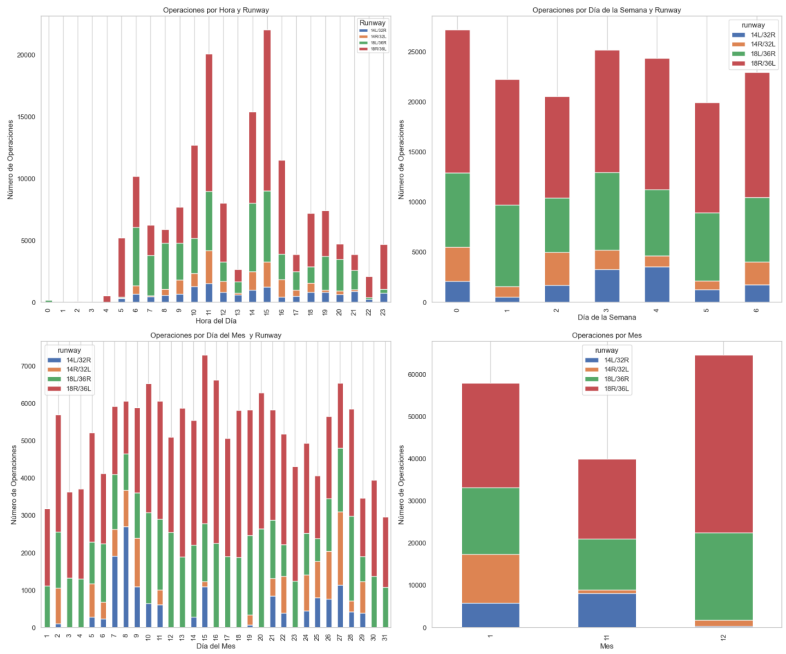

Figura 3: Análisis temporal de los despegues según pista de uso

Se observa en el gráfico que el uso de pistas según el día es más o menos igual. Sin embargo, en diciembre no se utilizó en ningún momento la pista 14L/32R y casi tampoco la 14R/32L.

En cuanto a horas valle y pico, se puede observar que prácticamente no se realizan operaciones entre las y 4 de la mañana. A partir de las 5 comienza a haber actividad en el aeropuerto hasta llegar a un primer pico sobre las 11 de la mañana. Posteriormente

observamos un valle entre las 12 y la 13 que podríamos asociar a la hora de comer y descanso de los trabajadores. Tras este valle se produce una nueva hora pico de 14 a 16. Posteriormente el número de operaciones realizadas por hota desciende gradualmente hasta media noche.

2.3. Análisis tráfico aéreo

Ahora estudiaremos como es el tráfico de aviones en las pistas, es decir, cuando hay pistas o holding points ocupados o cuando hay aviones de camino a la pista.

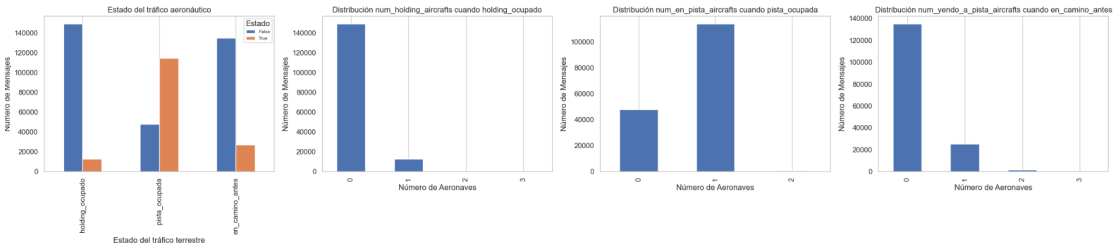

Figura 4: Estudio del tráfico de aeronaves en el aeropuerto

En la primera figura observamos el numero de mensajes enviados desde aviones en los puntos de espera mientras hay tráfico aéreo (punto de espera ocupado, pista ocupada o avión de camino a pista). Observamos que los casos en los que hay otros puntos de espera ocupados o aviones de camino a la pista de despegue son mínimo. En cambio, en la gran mayoría de casos hay algún avión ocupando la pista mientras otro espera en un punto de espera.

En las siguientes 3 gráficas observamos cuantos aviones hay en cada caso (ocupando punto espera, ocupando pista o de camino a la pista). Observamos que, en caso de ser True, lo normal es que solo haya un avión más en algún punto de espera y un solo avión de camino a la pista. Por otro lado, lo normal es que haya 1 o 0 aviones ocupando la pista.

A continuación observamos distintas distribuciones de columnas asociadas al tráfico aéreo. Tenemos tiempos-holding-x, que representa el tiempo que lleva esperando en el punto de espera el avión número X encontrado que está ocupando un punto de espera a la vez. Tenemos distancia-holding-X, distancia-en-pista-x y distancia-yendo-a-pista-X, que representan las distancias hasta el punto donde se suele despegar de los aviones que cumplen las condiciones de ocupando holding point, ocupando o pista o de camino a las pista.

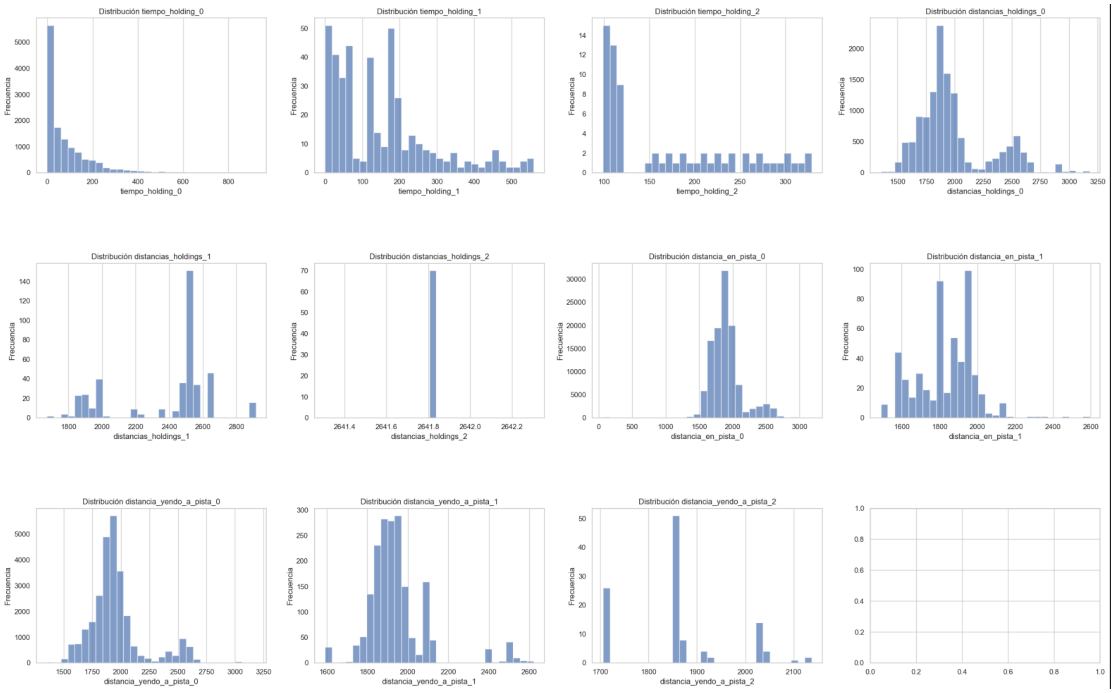

Figura 5: Estudio del tráfico de aeronaves en el aeropuerto

Podemos ver que muchas de las distribuciones no tienen suficiente cantidad de datos para poder ver la población real. Esto se debe a que estas columnas solo toman valor si su situación se cumple y son pocas veces cuando se cumple.

2.3.1. Historial de despegues

En esta sección analizaremos el historial de despegues en la pista de despegue asignada. Veremos tanto el número de despegues en los últimos minutos tanto la media de frecuencia de vuelos en los últimos minutos.

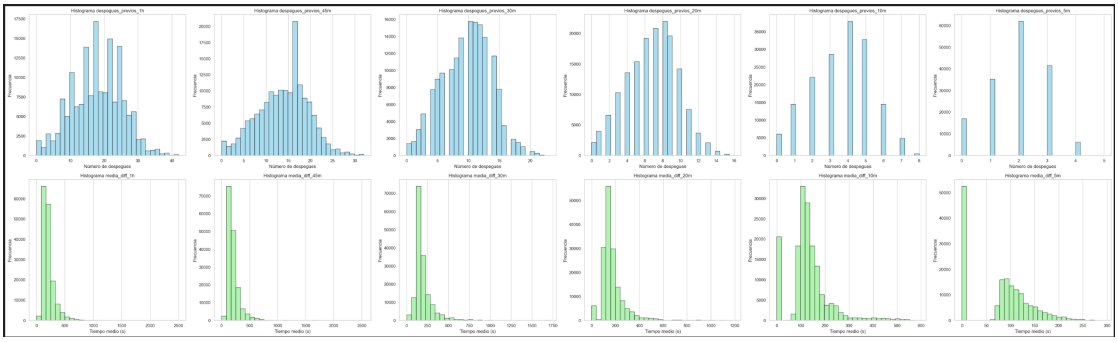

Figura 6: Histogramas sobre el historial de despegues de la pista objetivo

En la Figura 6 se muestran los histogramas de *despegues_previos_Xm* para diferentes valores de X. En todos los casos, la distribución presenta una moda decreciente conforme

se reduce la ventana de observación:

- Ventana 1 h ($X = 60$ min): pico alrededor de 15–20 despegues.
- Ventana 45 min: pico cercano a 12–15 despegues.
- Ventana 30 min: pico en 8–12 despegues.
- Ventana 20 min: pico en 6–9 despegues.
- Ventana 10 min: pico en 3–6 despegues.
- Ventana 5 min: pico en 1–3 despegues.

Esto sugiere que el tráfico se comporta de manera aproximadamente proporcional al tamaño de la ventana, con una cola derecha moderada que indica episodios de alta congestión.

La Figura 6 muestra los histogramas de *media_diff_Xm*, la media de los intervalos en segundos entre despegues en las mismas ventanas. Observamos:

- Para ventanas largas (1 h, 45 min), la mayoría de los intervalos se concentran por debajo de 300 s, con una cola larga que llega hasta 2 000 s, reflejando posibles pausas en horas de baja actividad.
- A medida que la ventana disminuye (30 min, 20 min, 10 min, 5 min), la distribución se desplaza hacia abajo (medias más cortas), concentrándose entre 50 s y 200 s.
- En ventanas cortas se puede observar un pico en -1. Esto se debe a escenarios en los que solo ha habido 0 o 1 despegues en la ventana de tiempo por lo que no se ha podido calcular la frecuencia de despegue en la pista.

2.4. Tiempo en espera y tiempo desde último despegue

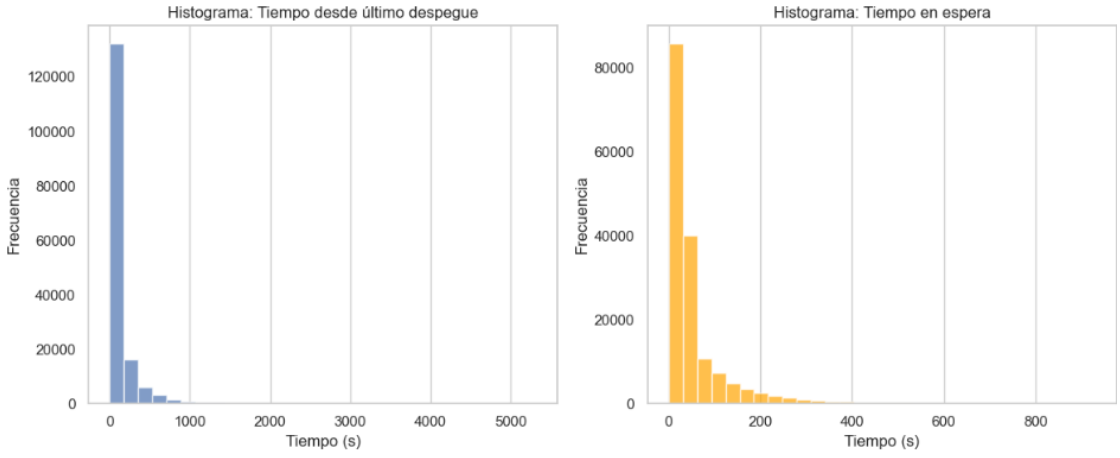

Figura 7: Histogramas de tiempo en espera y tiempo desde último despegue

Vemos las distribuciones de otras dos columnas que consideramos muy relevantes para el modelo como son, el tiempo que lleva esperando el avión el el punto de espera y cuanto tiempo ha pasado desde el último despegue (-1 es mayor a 90 minutos).
A partir de los histogramas de la Figura 7, destacamos:

- Ambas variables presentan una **distribución sesgada a la derecha**.
  - *Tiempo desde último despegue*: la práctica totalidad de los intervalos se concentra por debajo de los 200s, con una cola que alcanza hasta 5000s, reflejando periodos de muy baja actividad o paradas prolongadas.
  - *Tiempo en espera*: la mayor frecuencia se sitúa entre 20s y 40s, casi siempre por debajo de los 150s, y con muy pocos valores que superen los 600s.
- El comportamiento operativo es, en general, **fluido y eficiente**: los despegues se suceden con intervalos cortos y las esperas en rampa son breves en la mayoría de los casos.
- Existen **valores atípicos** (colas largas) que corresponden a momentos de baja demanda o incidencias (cierres de pista, meteorología adversa, etc.). Para análisis o modelado predictivo, se sugiere:
  - Tratamiento de outliers mediante recorte o transformaciones (por ejemplo, logarítmica).
  - Posible segmentación del análisis por rangos de tráfico para separar condiciones de operación normales de eventos extremos.

2.5. Análisis de la Variable `tiempo_hasta_despegue`

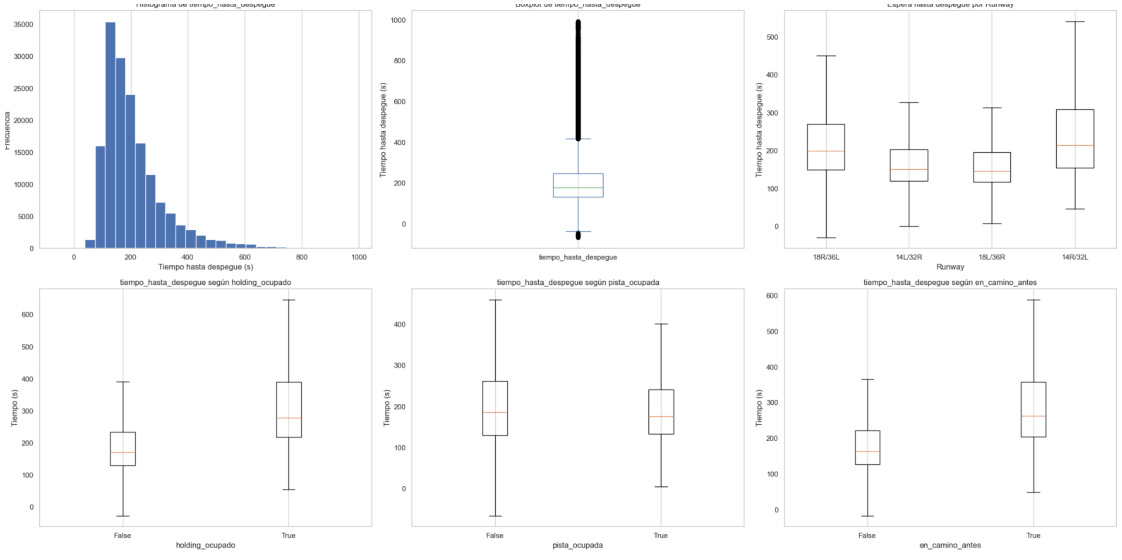

Figura 8: Análisis del tiempo hasta el despegue

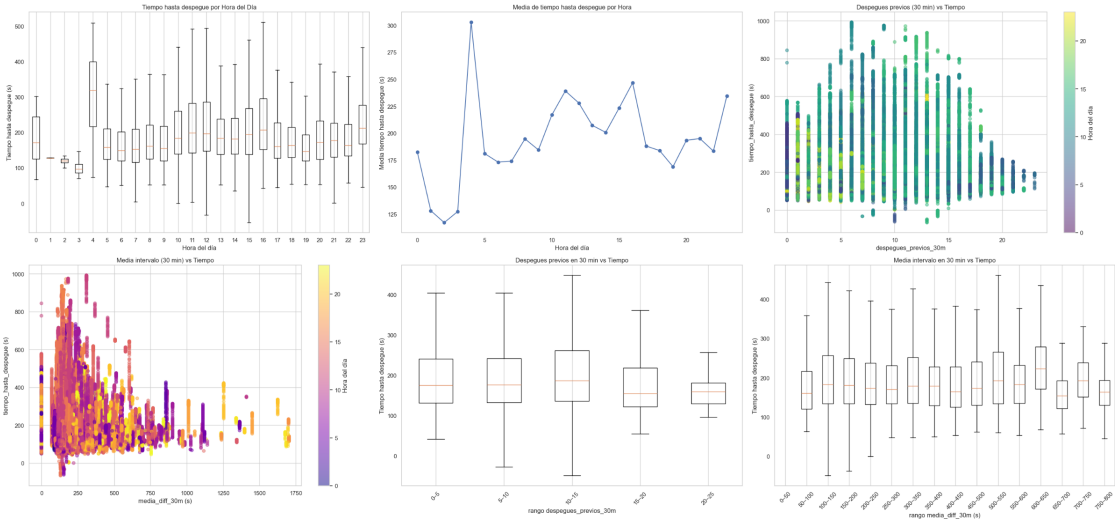

Figura 9: Análisis del tiempo hasta el despegue

Distribución General

- **Histograma (Imagen 1, superior izquierda):** Muestra la frecuencia de los diferentes tiempos hasta el despegue. La distribución está fuertemente **sesgada hacia la derecha** (asimetría positiva). Esto significa que la mayoría de los vuelos tienen tiempos de espera relativamente cortos, pero hay una cola larga de vuelos que experimentan tiempos considerablemente más largos.
- **Moda:** El tiempo más frecuente (la barra más alta del histograma) se sitúa alrededor de los $150 - 200$ segundos.
- **Boxplot General (Imagen 1, superior centro):** Confirma la asimetría derecha.
  - **Mediana:** La línea central de la caja (mediana o percentil 50) parece estar en torno a los $180 - 200$ segundos. Esto indica que la mitad de los vuelos despegan en menos de este tiempo.
  - **Rango Intercuartílico (IQR):** La caja (que contiene el 50 % central de los datos) es relativamente compacta, quizás abarcando desde $\approx 130s$ hasta $\approx 250s$.
  - **Outliers (Valores Atípicos):** Hay una cantidad muy significativa de outliers por encima del bigote superior. Estos puntos representan los vuelos con tiempos de espera excepcionalmente largos, llegando hasta $900s$ o más según el eje Y. Esto subraya la importancia de la cola derecha de la distribución.
- **Media vs. Mediana:** Debido a la asimetría derecha y la presencia de outliers altos, la **media** del `tiempo_hasta_despegue` será **mayor** que la mediana. El gráfico de línea de la media por hora (Imagen 2, superior centro) muestra valores medios que superan los $200 - 250$ segundos en horas punta, confirmando esto.

Factores que Influyen en `tiempo_hasta_despegue`

- **Hora del Día (Imagen 2, superior izquierda y centro):** Este es un factor muy influyente.

- Se observa un patrón cíclico claro a lo largo de las 24 horas.
- Los tiempos son mínimos durante la noche y la madrugada (aprox. 00:00 - 06:00).
- Hay picos pronunciados durante las horas de mayor tráfico: uno por la mañana (aprox. 07:00 - 10:00) y otro por la tarde/noche (aprox. 15:00 - 19:00).
- Tanto la mediana como la variabilidad (altura de las cajas y longitud de los bigotes en el boxplot por hora) aumentan considerablemente durante estas horas punta.

- **Demanda de Despegues y Congestión (Imagen 2, sup. der. e inf. izq y centro):** Observando los gráficos de la segunda imágen no se puede apreciar una relación clara entre el número de despegues que ha habido previamente y el tiempo hasta el despegue del avión. Si que es verdad que para momentos con un número de despegues previos superior a 15, el tiempo hasta el despegue suele ser menor como se aprecia tanto en el gráfico de dispersión como en el gráfico de boxplots donde los dos últimos boxplot se extienden menos y su mediana es menor en comparación con los otros 3.

  Por otro lado, viendo el gráfico de dispersión de la segunda imágen (inferior izquierda) que representa la media_diff_30m frente tiempo_hasta_despegue, se puede apreciar que los tiempos de espera altos (¿600 segundos) ocurren únicamente para momentos con una frecuencia de despegues baja. No se aprecia correlación clara entre las variables.

- **Runway (Pista) (Imagen 1, superior derecha):** Existen diferencias en la distribución del `tiempo_hasta_despegue` entre las distintas pistas. Para las pistas **14L/32R y 18L/36R**, el tiempo hasta el despegue suele ser menor.

4. Conclusiones Clave

- El `tiempo_hasta_despegue` es una variable muy variable, caracterizada por una distribución asimétrica con muchos vuelos rápidos pero una cola significativa de vuelos con retrasos considerables (outliers).
- Los principales impulsores del `tiempo_hasta_despegue` son la **hora del día** y la **congestión/demanda**.
- Las **condiciones operacionales** inmediatas (pista/holding ocupados) y la **pista asignada** también juegan un papel.

3. Transformación previa al entrenamiento

Tras el análisis exploratorio de los datos hemos observado distintas distribuciones y tendencias en los datos que conviene correfir antes de entrenar los modelos. Antes de ajustar nuestro modelo, aplicamos una serie de pasos de preprocesamiento para asegurar que los datos sean limpios, coherentes y estén en un formato adecuado para el entrenamiento. A continuación describimos, de forma cercana y paso a paso, cómo hemos construido el *pipeline* de transformación:

1. **Carga y filtrado inicial de datos.**
   - Se leen los datos desde un fichero Parquet y se descartan aquellos registros que no contienen valor de `wake_vortex` o presentan tiempos negativos para la variable objetivo `tiempo_hasta_despegue`.
   - Además, se filtran los casos donde `tiempo_hasta_despegue` supera los 600 segundos para eliminar outliers extremos.
2. **Eliminación de columnas irrelevantes.**
   - Se eliminan identificadores que no aportan información predictiva (`icao`, `flight_id`) y la marca temporal original (`ts`) para simplificar el conjunto de características.
3. **Codificación de información cíclica.**
   - Para las variables temporales `hour`, `month` y `day_of_week`, generamos sus componentes seno y coseno, de manera que la naturaleza cíclica queda reflejada sin crear saltos bruscos al pasar, por ejemplo, de las 23:00 a las 00:00.
4. **Tratamiento de valores faltantes y valores especiales.**
   - Aquellas variables donde el valor `-1` indicaba ausencia de información fueron reemplazadas por `NaN`.
   - Para mitigar el efecto de valores extremos, hicimos un *clipping* al 1$^{\text{er}}$ y 99$^{\text{o}}$ percentil en todas las variables numéricas.
5. **Agrupación de variables y selección de transformaciones.**
   - Dividimos las variables en cuatro grupos según su distribución y naturaleza:
     - *Variables a transformar con logaritmo*: aquellas con distribuciones muy sesgadas (`tiempo_en_espera`, `tiempo_desde_ultimo_despegue`, diffs de medias y tiempos de holding).
     - *Variables a escalar con* ***`RobustScaler`***: distancias y recuentos, para reducir el impacto de valores atípicos.
     - *Variables a escalar con* ***`StandardScaler`***: datos meteorológicos y contadores de despegues previos, que se aproximan a una distribución más normal.
     - *Variables categóricas*: se imputan con la moda y se codifican mediante `OrdinalEncoder` para no imponer un orden artificial.
6. **Construcción del *ColumnTransformer*.**
   - Cada uno de los grupos anteriores se procesa con un `Pipeline` específico que incluye:
     - Imputación de valores faltantes (`SimpleImputer`).
     - Transformación correspondiente (`FunctionTransformer` + logaritmo, `RobustScaler`, `StandardScaler`, `OneHotEncoder`).
   - Finalmente, se combinan todos con `ColumnTransformer`, descartando el resto de columnas que no estén en ninguno de los grupos.

7. **División en entrenamiento y prueba.**
   - Separamos los datos de forma temporal: utilizamos los meses de noviembre y diciembre, así como los primeros 15 días de otros meses, para el conjunto de entrenamiento; el resto va al conjunto de prueba.
8. **Ajuste y aplicación del preprocesador.**
   - Ajustamos el *ColumnTransformer* con los datos de entrenamiento.
   - Transformamos ambos conjuntos (entrenamiento y prueba), reconstruimos los `DataFrame` resultantes, reintroducimos la columna temporal `ts` y la variable objetivo `tiempo_hasta_despegue`, y guardamos los resultados en formato Parquet para alimentar el modelo.

Con este *pipeline* bien definido, garantizamos que todas las características entren al modelo normalizadas, libres de valores extremos, correctamente codificadas y con la información temporal tratada de forma consistente y humana.

4. Entrenamiento y evaluación de modelos

A continuación describimos de manera cercana y paso a paso cómo llevamos a cabo el proceso de entrenamiento y evaluación de nuestros modelos:

1. **Carga de datos y partición inicial.**
   - Leemos los conjuntos preprocesados (`train.parquet`, `test.parquet`) y descartamos filas con valores faltantes.
   - Separamos características (`X`) y variable objetivo (`y` = `tiempo_hasta_despegue`) en ambos conjuntos.
2. **Split entrenamiento–validación basado en fecha.**
   - Usamos la fecha (`ts`) para dividir el conjunto de entrenamiento en:
     - *Train*: días del mes ≤ 25.
     - *Validation*: días del mes > 25.
   - Eliminamos la columna `ts` antes de entrenar los modelos.
3. **Transformación logarítmica del target.**
   - En el análisis exploratorio de los datos vimos una distribución de la variable respuesta con una gran cola a la derecha por lo que en este punto le aplicamos $\log(1 + y)$ a `y_train` y `y_val` para estabilizar varianzas.
   - Mantenemos `y_test` en escala original para evaluar resultados finales.
4. **Selección de características con *Mutual Information*.**
   - Calculamos puntajes de *mutual information* entre cada variable de entrada y el target log-transformado.
   - Definimos un umbral (e.g. 0.005) y seleccionamos las variables que lo superan.
   - Tras este proceso filtramos de 46 columnas a 30 columnas: ['distancia$_y endo_{ap} ista'_0$,' $num_y endo_{ap} is$

5. **Preparación de conjuntos finales.**
   - Reducimos `X_train`, `X_val` y `X_test` a las columnas *seleccionadas*.
6. **Función de evaluación común.**
   - Definimos `evaluate_model()`, que:
     - Predice sobre el test set en escala log.
     - Invierte la transformación ($\exp -1$) y fuerza predicciones no negativas.
     - Calcula MAE, RMSE y R^2, además del tiempo de predicción.
7. **Entrenamiento y optimización de modelos.**
   - Empleamos `GridSearchCV` con validación cruzada (`cv=3`) usando como métrica *neg_mean_absolute_error* (MAE).
   - Modelos optimizados:
     - **Random Forest:** exploramos número de árboles, profundidad máxima y criterios de división.
     - **Gradient Boosting:** variamos estimadores, profundidad, tasa de aprendizaje y submuestreo.
     - **SVR (RBF):** ajustamos C y ϵ.
     - **KNN:** probamos distintos vecinos, esquemas de pesos y distancias (Manhattan vs. Euclidiana).
8. **Comparación y selección del mejor modelo.**
   - Recopilamos resultados de MAE, RMSE, R^2 y tiempo de predicción en un `DataFrame`.
   - Visualizamos métricas en gráficos de barras para identificar rápidamente el más preciso y eficiente.
   - Seleccionamos el modelo con menor MAE para análisis detallado.
9. **Resultados**

Cuadro 1: Comparación Final de Modelos

Modelo	**MAE**	**RMSE**	R^2	**Tiempo Predicción (s)**
Gradient Boosting	67.642	104.077	0.166	0.0137
Random Forest	68.482	104.919	0.152	0.1359
SVR	72.761	116.293	−0,041	31.2965
KNN	80.728	117.787	−0,068	2.3240

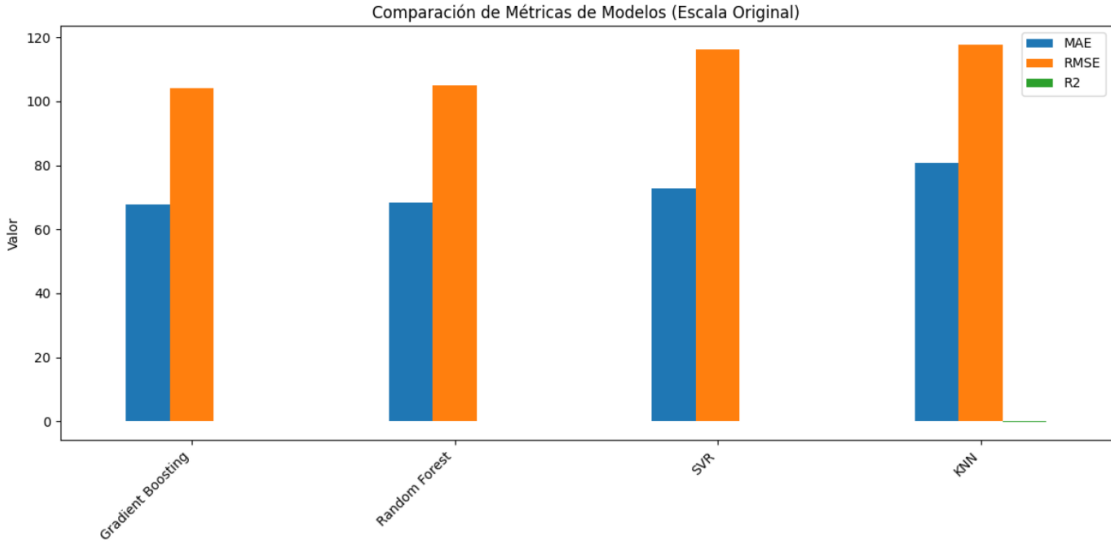

Figura 10: Resultados entrenamiento de modelos

Podemos observar que el mejor modelo es el **Gradient Boosting** con un MAE en los datos de test de **67.642 segundos**

10. **Análisis detallado del mejor modelo.**

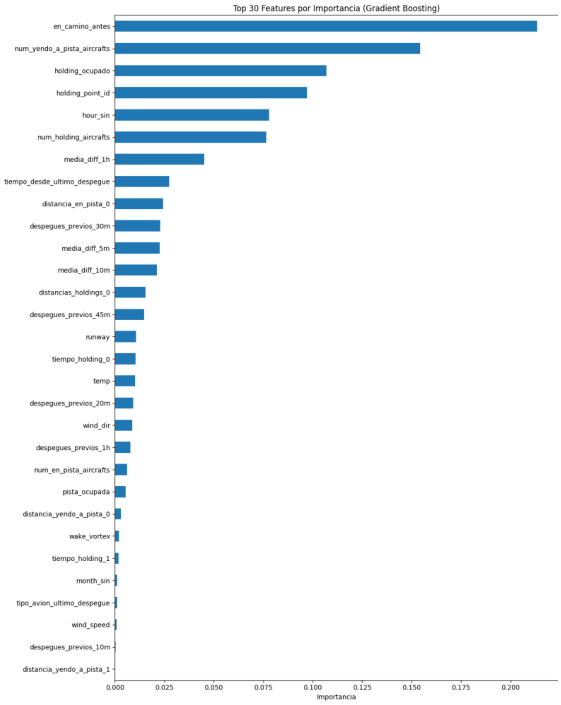

Figura 11: Importancia de columnas en el modelo

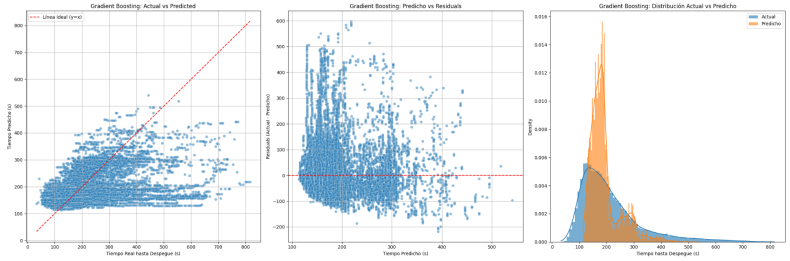

Figura 12: Análisis detallado de las predicciones

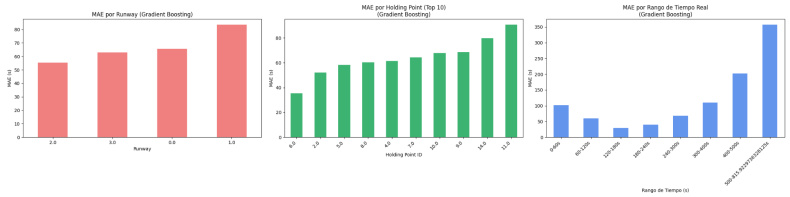

Figura 13: Análisis de las predicciones segmentadas por pista, punto de espera y por rango de tiempo real

Analizando los resultados observamos que las predicciones no son demasiado buenas. Como se aprecia en la Figura 12, el gráfico de valores reales versus predichos muestra una mayor concentración de datos en el rango inferior (0-300s), con una dispersión que aumenta notablemente para valores superiores. La línea diagonal de referencia (y=x) pone de manifiesto una tendencia del modelo a subestimar sistemáticamente los tiempos más elevados, fenómeno que podría atribuirse a la menor representación de estos casos en los datos de entrenamiento. Probablemente esto se deba al recorte de outliers con un tiempo hasta despegue superior a 600 segundos.

Los residuos presentan un comportamiento heterocedástico, con varianza no constante a lo largo del rango de predicciones. En el 3 gráfico de la figura 12 se puede ver la comparación de distribuciones entre valores reales y predichos, la cual confirma cierto sesgo, con la distribución de predicciones mostrando menor dispersión que los valores reales. El modelo predice mayoritariamente valores entre 100 y 300 segundos y no consigue distinguir casos extremos donde el tiempo hasta despegue es muy pequeño o muy grande.

Como consecuencia del hecho de que el modelo no generalize bien para datos con tiempos hasta el despegue muy elevados, podemos observar en el 3º gráfico de la figura 13 que el MAE para los tiempos de espera más prolongados, particularmente en el rango superior es muy superior al resto de rangos. Este comportamiento confirma la dificultad inherente en la predicción de eventos atípicos o situaciones de congestión extrema.

El análisis segmentado revela información valiosa sobre el rendimiento contextual del modelo. El error absoluto medio (MAE) varía significativamente entre pistas de despegue, con diferencias superiores al 20 % entre la pista más predecible (2.0 - 18L/36R) y la menos predecible (1.0 - 14R/32L). Probablemente esto se deba a que contamos con muy pocos datos sobre las pista 14R/32L. Esto también podría

indicar variaciones operativas o factores externos no capturados completamente por las variables del modelo.

5. Reparto de Trabajo

Cada integrante del grupo ha participado en distintas etapas del proyecto. A continuación, se presenta el desglose de trabajo realizado por cada miembro:

Nombre	**Tarea**	**Porcentaje**
Carlos Mantilla	Preprocesamiento	100 %
Héctor García	Preprocesamiento	100 %
Diego Alonso	Entrenamiento	100 %
Telmo Aracama	Limpieza y Entrenamiento	100 %
Mario López	Entrenamiento y Evaluación	100 %
Ignacio Gutierrez	Preprocesamiento	100 %
Pablo Rodríguez	Entrenamiento y Evaluación	100 %