Predicción de la Cantidad de Aeronaves en el Espacio Aéreo Cubano

Informe Técnico Final

Equipo de Desarrollo

8 de enero de 2026

Resumen

Este documento detalla el desarrollo de un sistema de predicción de tráfico aéreo para la FIR Habana. Se aborda la problemática de gestionar eficientemente el flujo de aeronaves mediante el uso de modelos de aprendizaje automático (XGBoost). El informe describe la metodología completa, desde la recolección y limpieza de datos (incluyendo la corrección crítica de conteo de vuelos únicos), hasta la ingeniería de características y la evaluación de resultados. Se discuten las implicaciones de los hallazgos y se proponen líneas de trabajo futuro.

Índice

1. Introducción

1.1. Antecedentes y Contexto

Breve historia del control de tráfico aéreo en la región y la necesidad de herramientas predictivas modernas.

1.2. Definición del Problema

Descripción formal del problema de forecasting de series temporales en este dominio.

1.3. Objetivos

1.3.1. Objetivo General

1.3.2. Objetivos Específicos

- Implementar pipeline de procesamiento de datos ATFM.
- Desarrollar modelos de regresión robustos.
- Validar resultados con métricas cuantitativas y cualitativas.

1.4. Alcance y Limitaciones

2. Marco Teórico

El presente capítulo establece los fundamentos conceptuales necesarios para comprender el alcance y la metodología del estudio. Se describe el contexto de la gestión de tráfico aéreo, las particularidades de las series temporales en este dominio y se detallan los algoritmos seleccionados para el desarrollo de los modelos predictivos.

2.1. Gestión de Afluencia del Tráfico Aéreo (ATFM)

La Gestión de Afluencia del Tráfico Aéreo (ATFM) es un servicio establecido para asegurar un flujo de tránsito aéreo seguro, ordenado y expedito. Su objetivo principal es asegurar que la capacidad de control de tránsito aéreo (ATC) sea utilizada al máximo posible y que el volumen de tráfico no exceda las capacidades declaradas por la autoridad competente. En el contexto de este proyecto, la región de estudio es la Región de Información de Vuelo (FIR) de La Habana, un corredor estratégico para el tráfico norte-sur en las Américas.

Un componente crítico en la medición del volumen de tráfico es el Identificador Único Global de Vuelo (GUFI). A diferencia de los conteos tradicionales por sector que pueden inflar las cifras debido a que un mismo vuelo atraviesa múltiples sectores, el uso del GUFI permite contabilizar aeronaves únicas en el sistema. Esta distinción es fundamental para obtener una estimación precisa de la carga de trabajo real y la demanda del espacio aéreo [1].

2.2. Análisis de Series Temporales en Aviación

El tráfico aéreo se caracteriza por exhibir patrones temporales altamente definidos. Se observa una fuerte estacionalidad a múltiples niveles: ciclos diarios determinados por los horarios comerciales de las aerolíneas, ciclos semanales con variaciones entre días laborables y fines de semana, y ciclos anuales influenciados por temporadas turísticas y festividades. El problema de predicción se aborda analizando estas componentes (tendencia, estacionalidad y ruido) en la serie histórica para proyectar valores futuros en un horizonte definido [2].

2.3. Algoritmos de Predicción Utilizados

En este estudio se han implementado y comparado diversos enfoques de modelado, abarcando desde estadísticas clásicas hasta aprendizaje profundo.

2.3.1. Modelos Estadísticos y Aditivos

Los modelos ARIMA (AutoRegressive Integrated Moving Average) y su variante estacional SARIMA constituyen la línea base clásica para la predicción de series temporales. Estos modelos asumen que los valores futuros son una función lineal de los valores pasados y los errores pasados [3]. Por otro lado, Prophet es un modelo aditivo desarrollado por Facebook, diseñado para manejar series temporales con fuertes efectos estacionales y días festivos. Descompone la serie en componentes de tendencia, estacionalidad y eventos, siendo particularmente robusto frente a valores atípicos y cambios en la tendencia [4].

2.3.2. Aprendizaje Automático Basado en Árboles

Los métodos de ensamble basados en árboles de decisión han demostrado un rendimiento superior en datos tabulares estructurados. Random Forest construye múltiples árboles de decisión durante el entrenamiento y genera la media de las predicciones de los árboles individuales, reduciendo la varianza y el riesgo de sobreajuste [5].

XGBoost (Extreme Gradient Boosting), el modelo central de este estudio, es una implementación optimizada del algoritmo de Gradient Boosting. A diferencia de Random Forest que construye árboles independientes, XGBoost construye árboles de forma secuencial, donde cada nuevo árbol intenta corregir los errores de los anteriores. Utiliza una función objetivo regularizada que controla la complejidad del modelo, haciéndolo altamente eficiente y preciso para capturar patrones no lineales en grandes conjuntos de datos [6].

2.3.3. Redes Neuronales Recurrentes

Las redes LSTM (Long Short-Term Memory) son una arquitectura de redes neuronales recurrentes diseñada para aprender dependencias a largo plazo. A través de mecanismos de compuertas (entrada, olvido y salida), las LSTM pueden retener información relevante sobre secuencias temporales extensas mitigando el problema del gradiente desvaneciente, lo que las hace idóneas para modelar dinámicas temporales complejas [7].

2.4. Métricas de Evaluación

Para cuantificar el desempeño de los modelos se utilizan las siguientes métricas estándar:

El Error Absoluto Medio (MAE) mide la magnitud promedio de los errores en un conjunto de predicciones, sin considerar su dirección:

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

La Raíz del Error Cuadrático Medio (RMSE) es una medida cuadrática que penaliza más severamente los errores grandes, siendo útil para detectar predicciones con desviaciones significativas:

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

El Coeficiente de Determinación (R^2) indica la proporción de la varianza en la variable dependiente que es predecible a partir de las variables independientes, proporcionando una medida de la bondad de ajuste del modelo:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

3. Metodología Experimental

Este capítulo describe el diseño experimental adoptado para desarrollar y validar los modelos de predicción. Se detallan los procesos de recolección de datos, las estrategias de integración de fuentes heterogéneas y las técnicas de ingeniería de características aplicadas para construir un dataset robusto para el entrenamiento.

3.1. Adquisición y Procesamiento de Datos

El sistema se alimenta de tres fuentes de datos primarias: registros meteorológicos históricos, noticias de eventos relevantes y registros de operaciones aéreas. A continuación, se especifica el flujo de procesamiento para cada una de estas fuentes.

3.1.1. Datos Meteorológicos

Para la gestión de los voluminosos datos climáticos históricos, se desarrolló la herramienta `db-tool`. Esta utilidad implementa un ecosistema de contenedores Docker orquestados para la carga paralela de respaldos de bases de datos MSSQL. Utiliza un enfoque multihilo (`ThreadPoolExecutor`) para parsear y ejecutar scripts SQL de manera eficiente, monitorizando el progreso en tiempo real. Posteriormente, el módulo `etl-tool` se encarga de la extracción, transformación y limpieza de las variables de interés (visibilidad, velocidad del viento, temperatura) que impactan directamente en la operatividad aeroportuaria.

3.1.2. Noticias y Eventos

Se implementó un sistema de extracción de información (`Event_extractor`) sobre un corpus de 13,022 noticias obtenidas del medio *Cubadebate*. El pipeline de procesamiento de lenguaje natural (NLP) incluye:

1. **Extracción de Entidades y Fechas:** Utilizando la librería `spaCy` para identificar referencias temporales y entidades nombradas relevantes.

2. **Clasificación de Eventos:** Se evaluaron múltiples algoritmos (Naive Bayes, Random Forest, Gradient Boosting), seleccionándose **LinearSVC** por obtener el mejor desempeño con una exactitud del 95.0 % en la categorización de noticias (deportivas, políticas, culturales, etc.).

3. **Análisis de Sentimiento:** Dado que los modelos pre-entrenados en redes sociales (como TASS-2019) mostraron un rendimiento deficiente en el dominio periodístico, se optó por un enfoque basado en diccionarios léxicos de polaridad (`KeywordSentimentClassifier`), que demostró mayor robustez para textos formales.

3.1.3. Datos de Operaciones Aéreas

Los datos de tráfico provienen de archivos ATFM y reportes ATC. Durante la fase de análisis exploratorio, se identificó una anomalía crítica en la agregación de datos horarios: la suma simple de conteos por sector (AOI) duplicaba las aeronaves que transitaban por múltiples sectores en una misma hora. Se corrigió esta lógica implementando un conteo basado en Identificadores Únicos Globales de Vuelo (GUFI), lo que permitió obtener el volumen real de aeronaves únicas en el sistema, reduciendo el error de magnitud en un 47 %.

3.2. Fusión e Integración de Datos

El reto principal consistió en la unificación de fuentes con diferentes frecuencias y naturalezas. Se estableció una frecuencia base horaria, alineando los registros meteorológicos mediante promedios horarios y proyectando el impacto de los eventos noticiosos en ventanas temporales específicas asociadas a sus fechas de ocurrencia.

3.3. Ingeniería de Características

Para capturar la dinámica del tráfico aéreo, se generaron las siguientes variables predictoras:

- **Variables Temporales Cíclicas:** Transformación de hora, día y mes mediante funciones seno y coseno para preservar la continuidad cíclica (e.g., la similitud entre las 23:00 y las 00:00).

- **Calendario y Festivos:** Variables binarias indicadoras de festivos nacionales cubanos e indicadores de proximidad (días previos y posteriores).

- **Ajuste COVID-19:** Incorporación de una variable de impacto para modelar la caída estructural y recuperación del tráfico durante los períodos de restricción pandémica.
- **Lags y Ventanas Móviles:** Creación de variables de retardo (t-1, t-24, t-168) y estadísticas móviles (media, máx, mín de las últimas 24 horas) para capturar la autocorrelación de la serie.

3.4. Estrategia de Validación

Para la evaluación de los modelos, se adoptó una partición de datos respetando la secuencia temporal, reservando los últimos 7 días del conjunto de datos exclusivamente para la prueba (testing). Esta estrategia .ºut-of-sample"simula un escenario real de pronóstico operativo. Las métricas seleccionadas para la comparación de modelos incluyen MAE, RMSE, R^2 y MAPE, priorizando la robustez ante valores atípicos y la precisión en la tendencia.

4. Desarrollo e Implementación

Este capítulo profundiza en la materialización de la solución propuesta, describiendo la arquitectura técnica del sistema, la implementación específica de los algoritmos de predicción y los procesos de optimización aplicados para maximizar su desempeño.

4.1. Arquitectura del Sistema

El sistema se ha diseñado siguiendo una arquitectura modular basada en tuberías (pipelines) de datos, lo que facilita la escalabilidad y mantenibilidad. El flujo de información comienza con los módulos de ingesta (`db-tool` y `Event_extractor`), cuyos datos convergen en un preprocesador centralizado. Este componente normaliza las escalas temporales y fusiona las características exógenas con el dataset principal. Posteriormente, el módulo de ingeniería de características genera las variables sintéticas descritas anteriormente antes de alimentar el registro de modelos, donde se gestionan el entrenamiento, validación y almacenamiento de las versiones de los modelos.

4.2. Modelos Evaluados

Se ha implementado un abanico de modelos predictivos, definidos en el módulo `models/model.py`, para evaluar distintas aproximaciones al problema de forecasting.

- **ARIMA/SARIMA:** Se estableció como línea base estadística. La implementación utiliza la biblioteca `statsmodels`, configurando los órdenes $(p, d, q)(P, D, Q)_s$ para capturar la estacionalidad diaria ($s = 24$) [3].
- **Prophet:** Se utilizó para descomponer explícitamente la estacionalidad y evaluar el impacto de los festivos como regresores adicionales. Su naturaleza aditiva permitió una interpretación clara de los componentes de tendencia [4].

- **Random Forest:** Implementado con `scikit-learn`, este modelo permitió capturar interacciones no lineales entre características sin requerir un escalado estricto de los datos. Se exploró la importancia de las características basada en la reducción de impureza [5].
- **XGBoost:** Implementado mediante la librería `xgboost`, este modelo de gradient boosting demostró ser altamente eficaz. Se configuró con una función objetivo de regresión cuadrática y métricas de evaluación RMSE para el proceso de boosting [6].
- **LSTM:** Se diseñó una red neuronal recurrente utilizando `TensorFlow`, con una arquitectura de capas ocultas apiladas para modelar dependencias temporales complejas a largo plazo [7].
- **Ensemble:** Se construyó un meta-modelo que combina las predicciones de los modelos individuales mediante un promedio ponderado, asignando mayores pesos a los modelos con menor error de validación (MAE) histórico.

4.3. Optimización de Hiperparámetros

Para mejorar el rendimiento de los modelos base, se llevó a cabo un proceso sistemático de ajuste de hiperparámetros (*hyperparameter tuning*). Se utilizó una estrategia de Búsqueda de Cuadrícula (*Grid Search*) con validación cruzada temporal (*Time Series Cross-Validation*) para evitar la fuga de datos del futuro al pasado.

Para el modelo XGBoost, los parámetros clave optimizados incluyeron:

- **Tasa de aprendizaje (learning_rate):** Se exploraron valores entre 0.01 y 0.3 para controlar la contribución de cada árbol.
- **Profundidad máxima (max_depth):** Se probaron profundidades de 3 a 10 para balancear la capacidad de modelado y el riesgo de sobreajuste.
- **Número de estimadores (n_estimators):** Se ajustó el número de árboles entre 100 y 1000, utilizando *early stopping* para detener el entrenamiento cuando el error de validación dejaba de disminuir.

4.4. Configuración Final

Tras el proceso de experimentación, la configuración que ofreció el mejor compromiso entre precisión y estabilidad para el modelo XGBoost desplegado fue:

- `learning_rate`: 0.05
- `max_depth`: 6
- `n_estimators`: 500
- `subsample`: 0.8 (para reducir varianza mediante el submuestreo de filas)
- `colsample_bytree`: 0.8 (submuestreo de columnas por árbol)

Esta configuración permitió al modelo generalizar correctamente sobre los días de prueba, reaccionando adecuadamente a los patrones horarios y a la influencia de los días festivos.

5. Resultados

5.1. Desempeño Global del Modelo

Tabla comparativa de métricas en el conjunto de prueba.

5.2. Análisis Temporal de Predicciones

Gráficos de series temporales (Real vs Predicho) para el horizonte de prueba.

5.3. Importancia de Características

Análisis de qué variables influyen más en la predicción.

6. Discusión

6.1. Interpretación de los Resultados

¿El modelo captura correctamente los picos y valles? ¿Cómo se comporta en días atípicos?

6.2. Análisis de Errores

Identificación de momentos con mayor error y posibles causas (e.g., eventos no capturados por las features).

6.3. Impacto de la Calidad de los Datos

Discusión sobre la reducción de volumen observada tras corregir la lógica de agregación y la persistencia de volúmenes altos en picos.

6.4. Implicaciones Operativas

Utilidad práctica del modelo para los controladores y planificadores.

7. Conclusiones y Recomendaciones

7.1. Conclusiones

7.2. Trabajo Futuro

Referencias

[1] International Civil Aviation Organization, *Manual on Collaborative Air Traffic Flow Management (Doc 9971)*. ICAO, Montreal, Canada, 2014.

[2] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 5th ed., 2015.

[3] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. Melbourne, Australia: OTexts, 2nd ed., 2018.

[4] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[5] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2016.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

A. Anexos

A.1. Fragmentos de Código Relevantes