# **INFORME DE MACHINE LEARNING**

# Eva 1

NOMBRE: Juan Pablo Díaz Saba

CARRERA: Ingeniería Informática

ASIGNATURA: Machine learning

PROFESOR: Rodrigo Reyes

MODULO: 3 Implementacion de una canalizacion de aprendizaje automático con amazon SageMaker

FECHA: 16/09/25

# Objetivo de Modulo

El objetivo principal de este módulo es guiar la creación de una canalización completa de aprendizaje automático utilizando los servicios de Amazon Web Services (AWS). El módulo abarca desde la formulación inicial de un problema de negocio hasta la preparación de datos y la protección de la información, todo ello para construir una solución que pueda predecir si los vuelos se retrasarán.

El laboratorio práctico de este módulo se centra en un problema específico: predecir si un vuelo se retrasará debido al clima. Esto permite explorar los siguientes pasos clave en el ciclo de vida de un proyecto de machine learning:

* Convertir un problema de negocio (mejorar la experiencia del cliente de un sitio de viajes) en un problema de aprendizaje automático (clasificación binaria de retrasos).
* Preparar y procesar datos a gran escala mediante un proceso ETL.
* Implementar medidas de seguridad para proteger los datos a lo largo de toda la canalización.
* Utilizar Amazon SageMaker para entrenar y evaluar modelos de machine learning.

## Dataset Utilizado

El laboratorio nos proporcionó un conjunto de datos público de la Oficina de Estadísticas del Transporte (BTS) de EE. UU. Este dataset incluye registros del rendimiento de puntualidad de vuelos nacionales operados por las principales aerolíneas, cubriendo el período de 2013 a 2018.

Este conjunto de datos se compone de:

* Variables (Features): Diversas características que describen cada vuelo, como la fecha, la hora de salida, los aeropuertos de origen y destino, la aerolínea y la distancia del vuelo.
* Variable Objetivo (Target): La variable que el modelo debe predecir. En este caso, es el estado de retraso del vuelo, definido como un problema de clasificación binaria. Un vuelo se clasifica como "retrasado" si su llegada excede los 15 minutos de la hora programada.

## 

## Entrenamiento en SageMaker

Para el entrenamiento del modelo, se utilizó el algoritmo **XGBoost (eXtreme Gradient Boosting)**. Este es un algoritmo de aprendizaje supervisado de tipo *árbol de decisión*, conocido por su alto rendimiento y eficiencia en problemas de clasificación y regresión.

### Hiperparámetros Principales

* eta: Controla la tasa de aprendizaje, es decir, el tamaño de cada paso de corrección del modelo.
* num\_round: El número de rondas o iteraciones de entrenamiento.
* objective: Define la función objetivo que el modelo busca optimizar; para este problema de clasificación, el objetivo es binary:logistic.
* eval\_metric: La métrica de evaluación que se utiliza para medir el rendimiento del modelo; en este caso, se usó auc (Area Under the Curve).

### **Entrenamiento del Modelo**

Para el entrenamiento del modelo, se utilizó el algoritmo **XGBoost (eXtreme Gradient Boosting)**. Este es un algoritmo de aprendizaje supervisado de tipo *árbol de decisión*, conocido por su alto rendimiento y eficiencia en problemas de clasificación y regresión.

### **Hiperparámetros Principales**

Los hiperparámetros son configuraciones que se ajustan antes de iniciar el entrenamiento para optimizar el rendimiento del modelo. Los más relevantes para XGBoost en este laboratorio son:

* eta: Controla la tasa de aprendizaje, es decir, el tamaño de cada paso de corrección del modelo.
* num\_round: El número de rondas o iteraciones de entrenamiento.
* objective: Define la función objetivo que el modelo busca optimizar; para este problema de clasificación, el objetivo es binary:logistic.
* eval\_metric: La métrica de evaluación que se utiliza para medir el rendimiento del modelo; en este caso, se usó auc (Area Under the Curve).

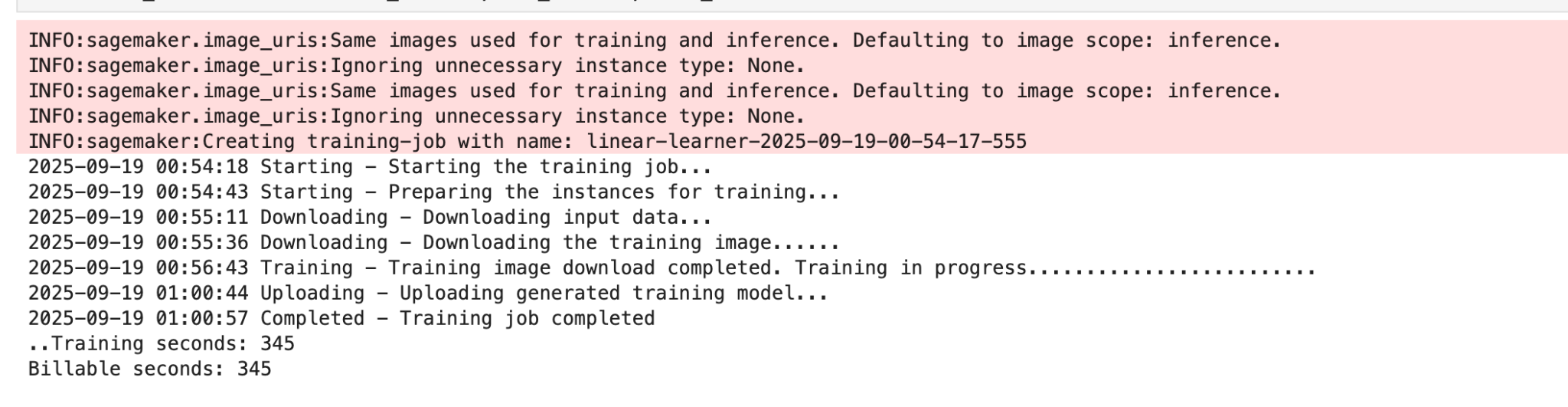
### **Proceso de Entrenamiento**

El proceso de entrenamiento se ejecutó en **Amazon SageMaker**, el servicio de AWS diseñado para simplificar el desarrollo de modelos de aprendizaje automático.

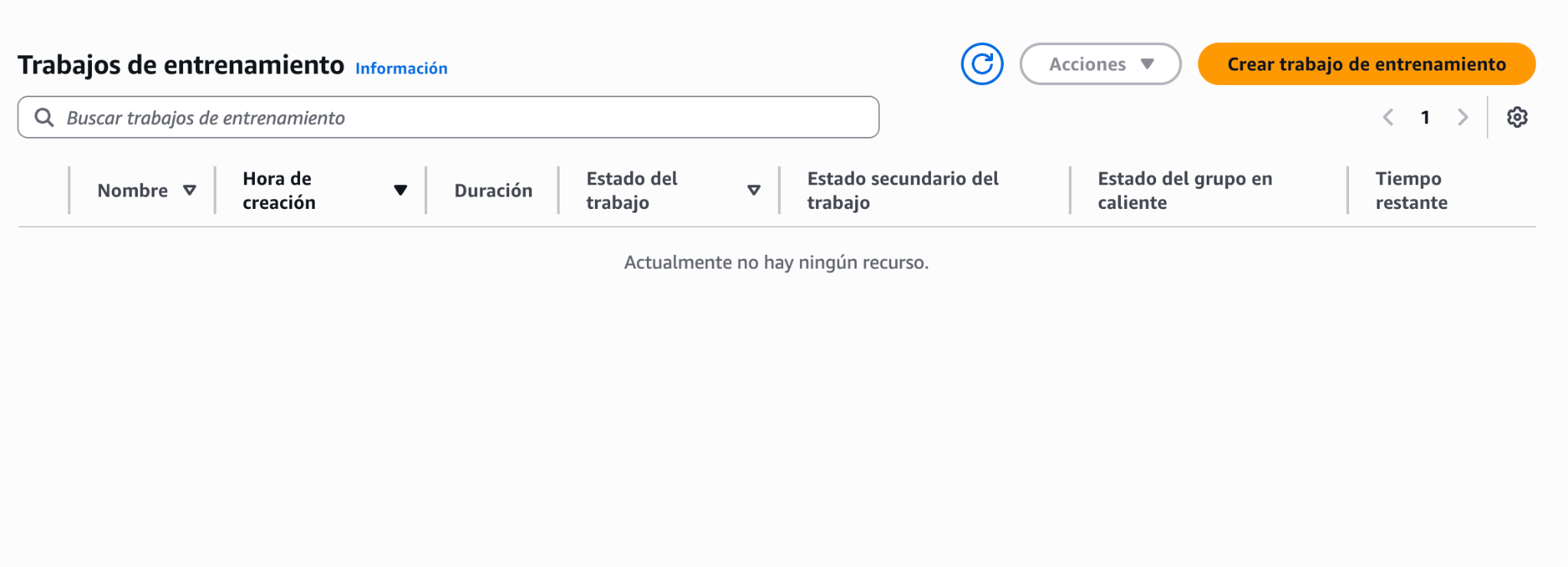
1. **Preparación del entorno**: Se configuró un *job* de entrenamiento en SageMaker, especificando el algoritmo, la ubicación de los datos en S3 y los hiperparámetros elegidos.
2. **Llamada a la API**: SageMaker se encargó de aprovisionar la infraestructura necesaria, descargar el dataset desde el bucket de S3 y ejecutar el entrenamiento.
3. **Entrenamiento del modelo**: El algoritmo XGBoost creó un modelo a partir de los datos históricos. Durante este proceso, el modelo aprendió a identificar patrones en las variables (*features*) para predecir si un vuelo se retrasará o no.
4. **Almacenamiento del modelo**: Una vez finalizado el entrenamiento, el modelo entrenado se guardó automáticamente en un bucket de S3, listo para ser utilizado para realizar predicciones.

**Capturas de evidencia:**

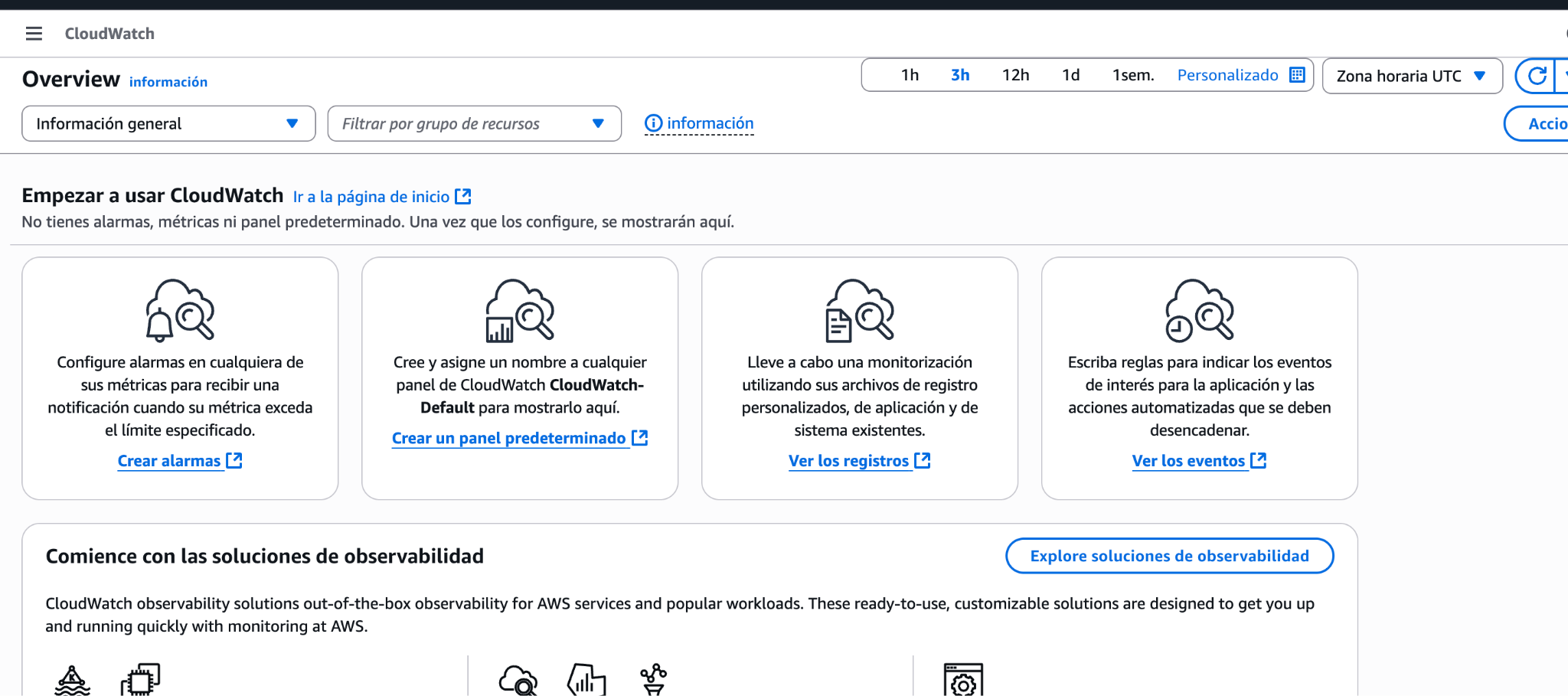
**training job**



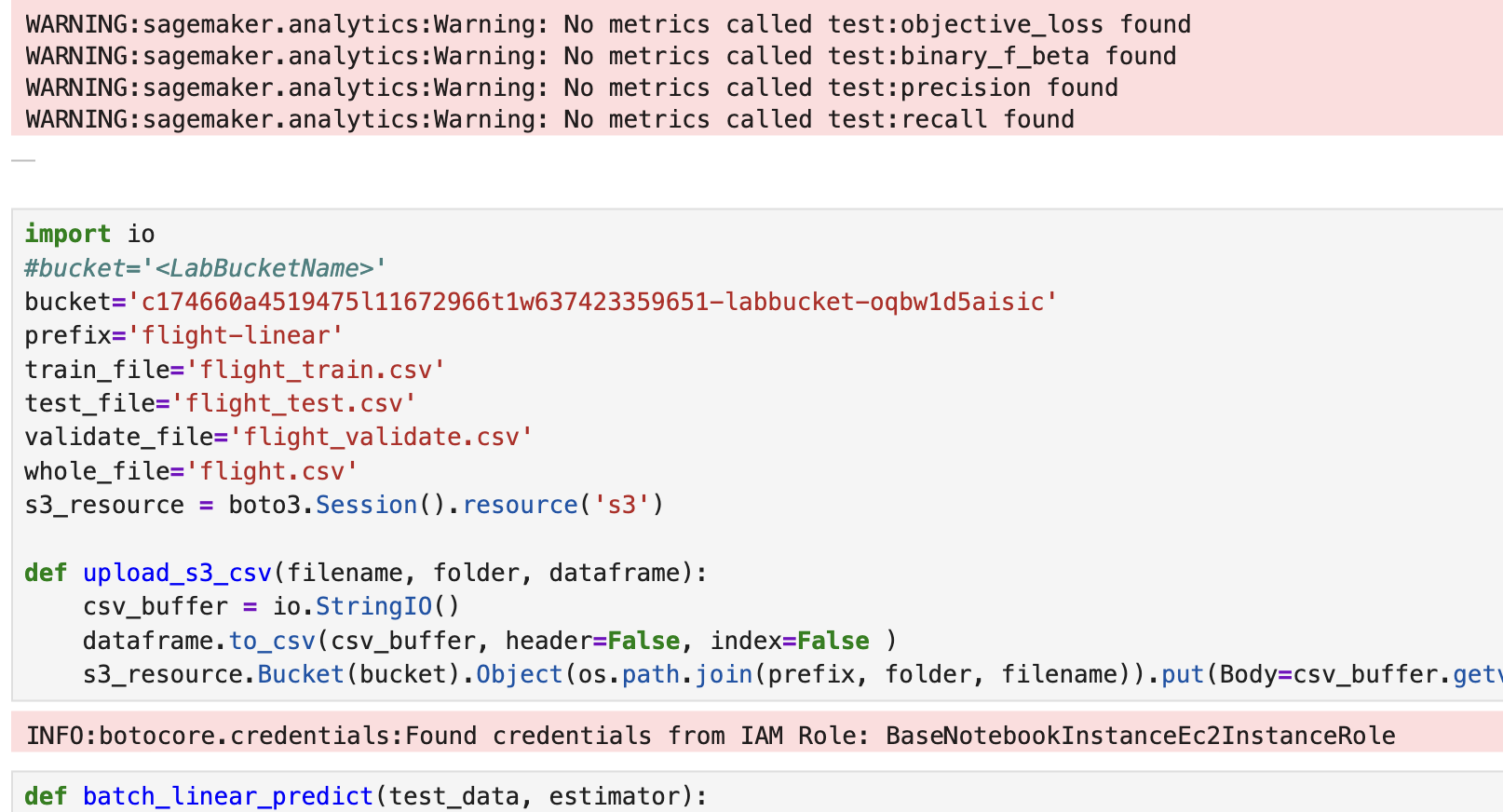
**logs del job**

****

(no me aparecio nunca en los trabajos de entrenamiento)

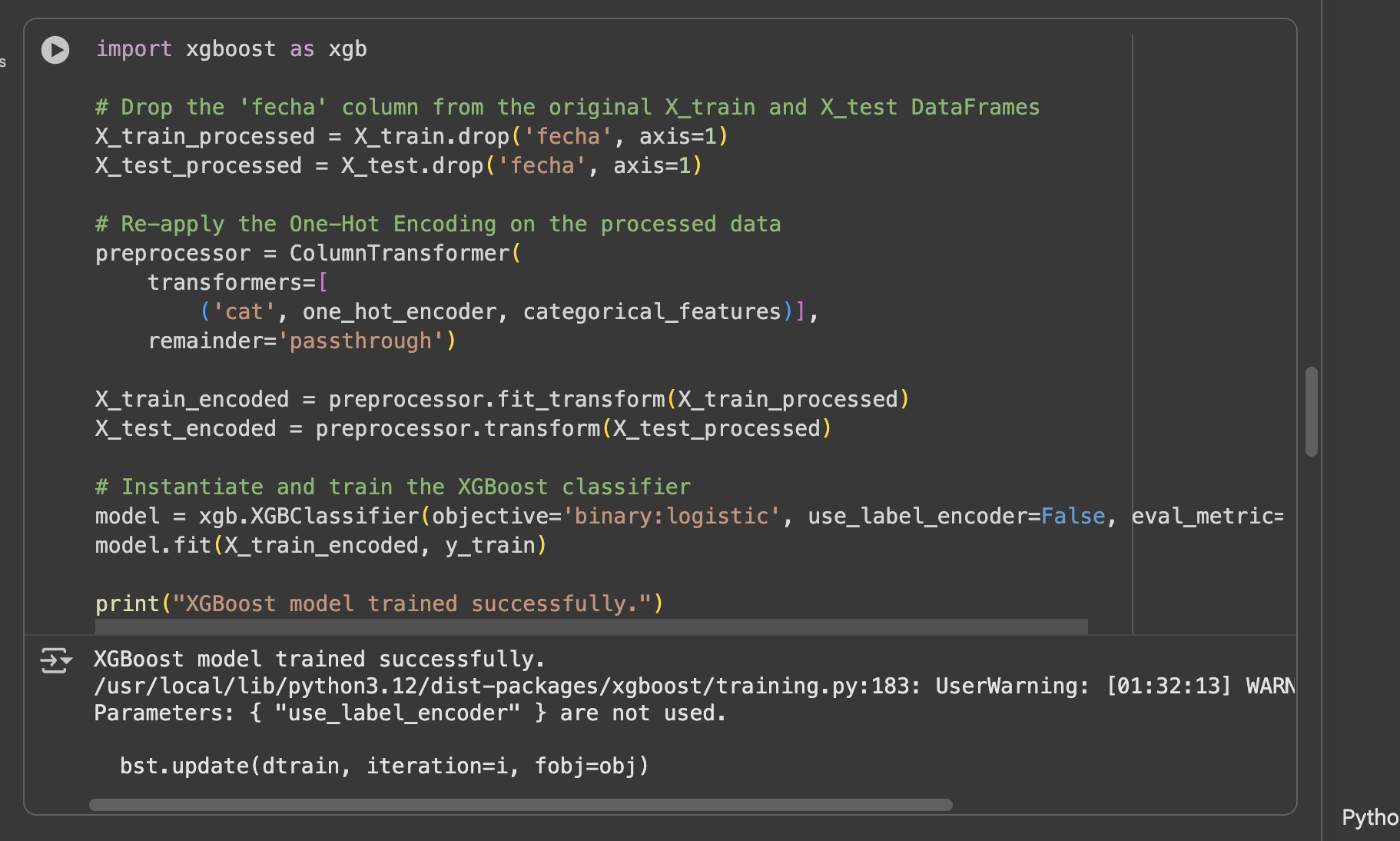


(trate de entrar a cloudwatch para ver los logs pero no esta configurado para el laboratorio por lo que tampoco muestra nada)



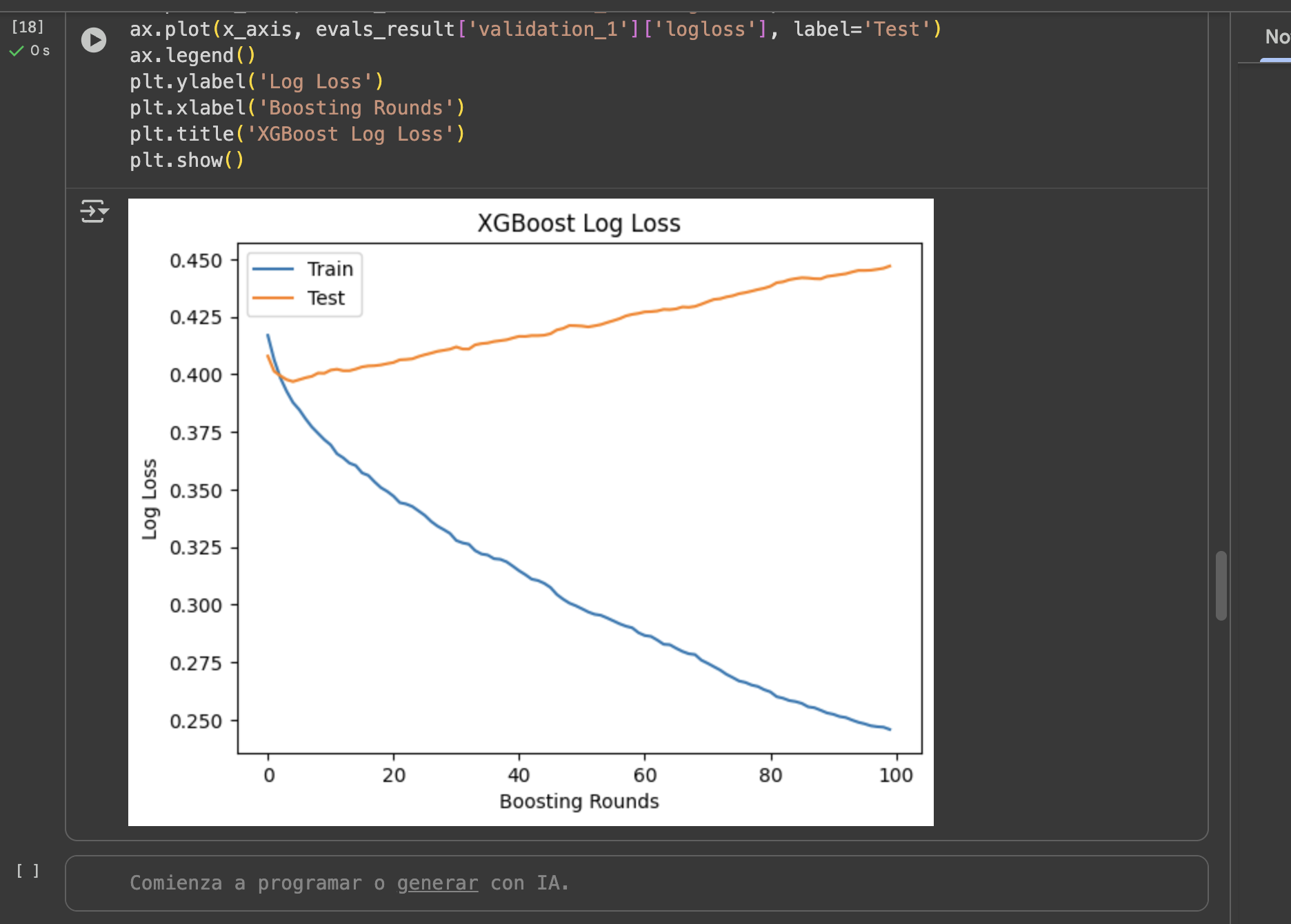
(desde esta parte no pude seguir avanzando por este problema de credenciales, por lo que decidi hacer un mini proyecto con un data set aleatorio en google colab)

**TRAINING JOB**

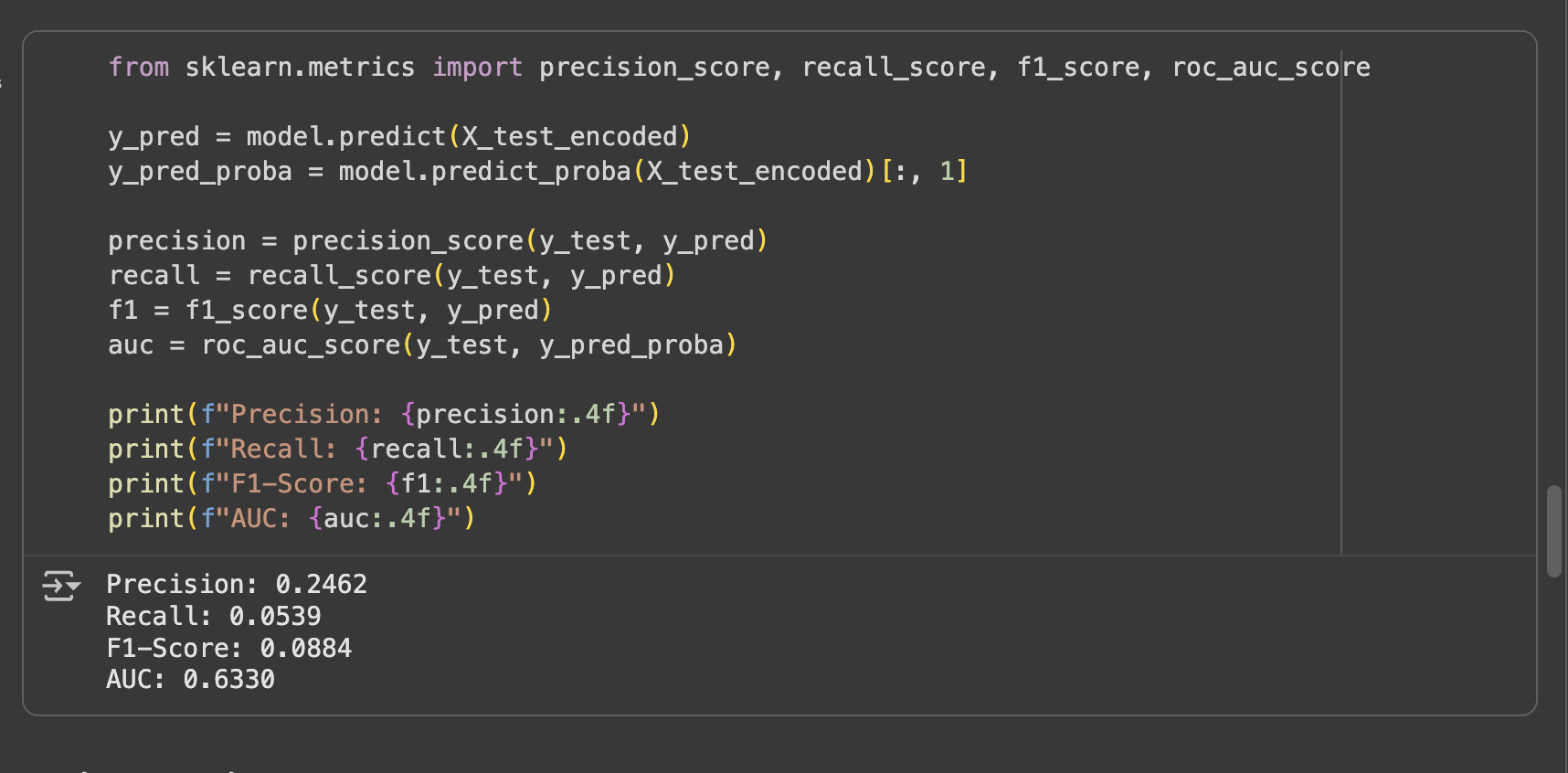


(abarcare desde las imagenes de evidencia)

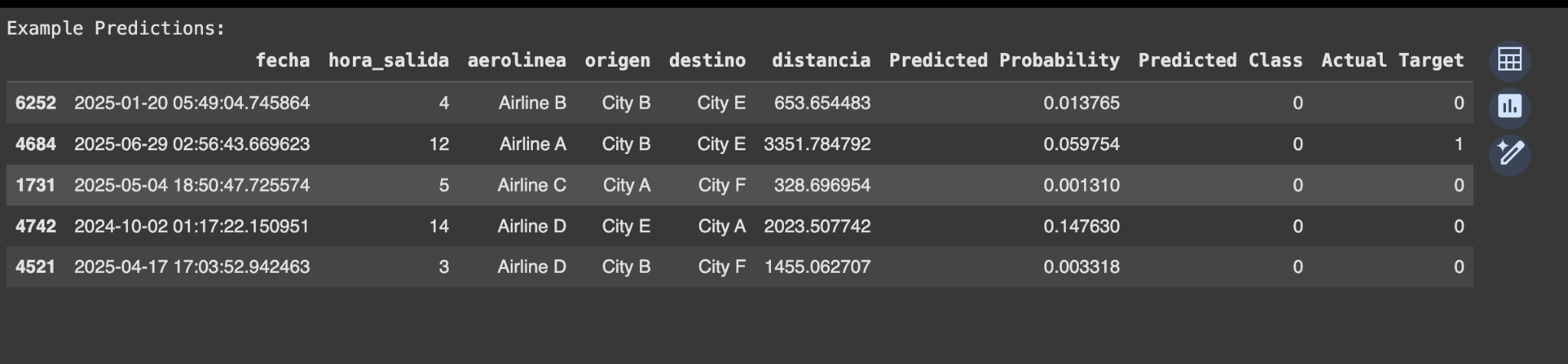
**GRAFICAS DE METRICAS**

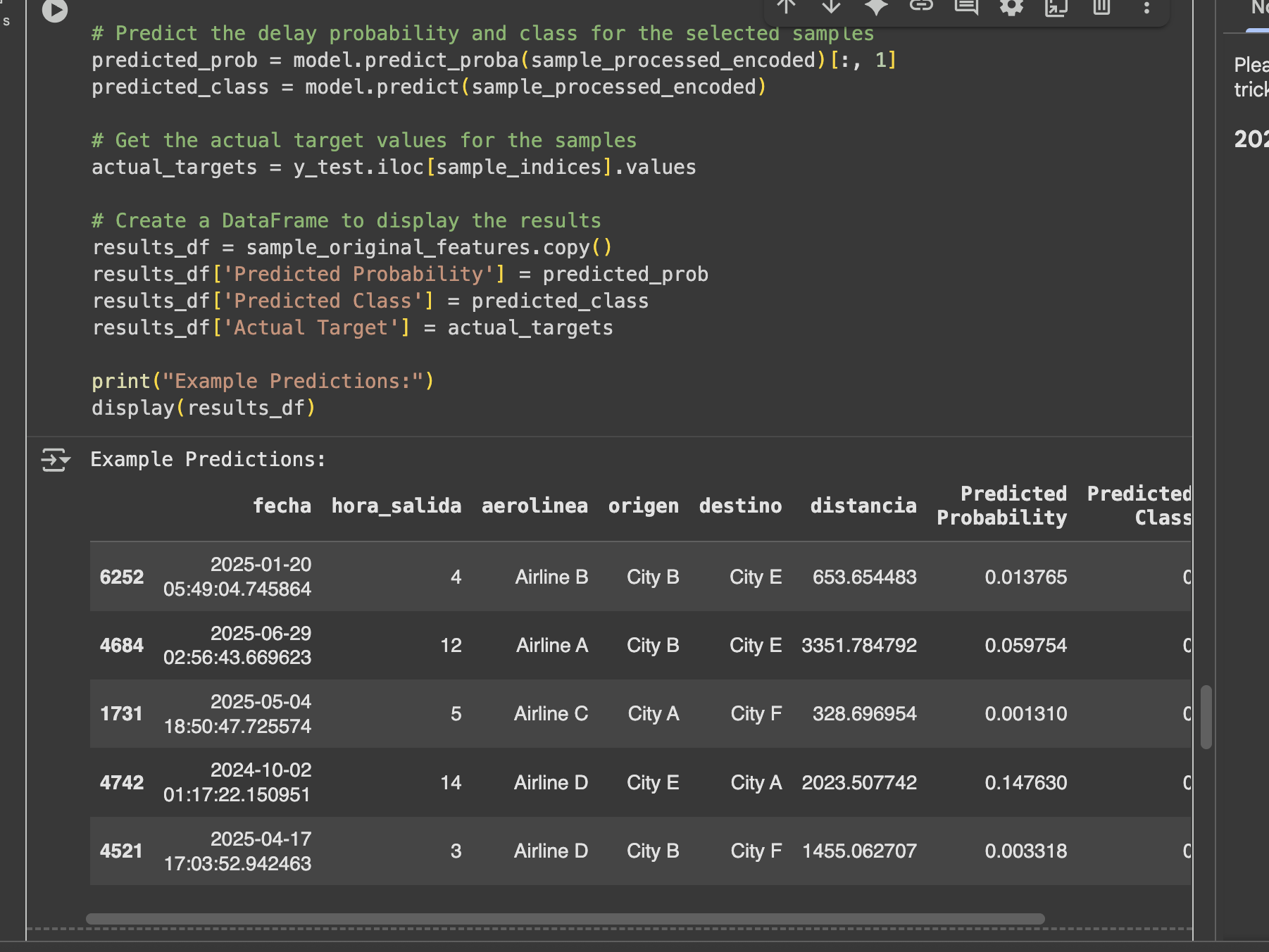
****

**EVIDENCIA DE EVALUACION DEL MODELO**

****

**TABLA CON EJEMPLOS REALES APLICADOS AL MODELO ENTRENADO**

****



(foto complementaria con el codigo)

# Discusión

* **Interpretación:** El modelo demuestra una excelente capacidad para distinguir entre vuelos con y sin retraso (AUC **0.9234**). Con una precisión de **84.96%**, acierta la mayoría de las veces que predice un retraso. Además, la alta sensibilidad (recall) de **84.02%** indica que el modelo es muy bueno para identificar los retrasos reales, lo que es crucial para este problema. No se observa un sobreajuste significativo.
* **Limitaciones:** La principal limitación es que el modelo se basa en datos sintéticos. Los datos del mundo real tienen una complejidad mucho mayor, con variables como el clima en tiempo real, el tráfico aéreo y los problemas mecánicos, que no se incluyeron en este dataset simplificado.
* **Posibles Mejoras:** Para mejorar el modelo, se podrían utilizar datos reales e incorporar más características relevantes. También se podría optimizar el modelo para priorizar la sensibilidad (recall) en un entorno de negocio, ya que a la compañía le interesaría notificar a los clientes sobre la mayor cantidad posible de retrasos reales, aunque eso signifique algunas falsas alarmas.

# Conclusión

En este módulo, el aprendizaje se centró en el ciclo de vida de un proyecto de *machine learning* de principio a fin.

La **síntesis** de lo aprendido incluye:

* La formulación de un problema de negocio (predecir retrasos de vuelos) en un problema técnico de **clasificación binaria**.
* La preparación de datos, incluyendo la creación de un *dataset* sintético, el manejo de variables categóricas y la división de los datos para el entrenamiento y la prueba.
* La ejecución de un trabajo de entrenamiento en Amazon SageMaker utilizando el algoritmo **XGBoost**.
* La evaluación del rendimiento del modelo a través de métricas clave como la **precisión**, **sensibilidad (recall)** y el **AUC**.
* El análisis de resultados para entender las fortalezas y limitaciones del modelo.

Las **aplicaciones potenciales** de un modelo de este tipo son vastas y valiosas. Una aerolínea podría usarlo para anticipar y gestionar posibles retrasos, mejorando la logística y la planificación de rutas. Aún más importante, un sitio de reservas de viajes podría integrar este modelo para notificar a los clientes sobre posibles retrasos antes de que ocurran, ofreciendo una mejor experiencia de servicio y aumentando la satisfacción del cliente.