
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

```
INFO:sagemaker.image_uris:Same images used for training and inference. Defaulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker.image_uris:Same images used for training and inference. Defaulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker:Creating training-job with name: linear-learner-2025-09-19-00-54-17-555
2025-09-19 00:54:18 Starting - Starting the training job...
2025-09-19 00:54:43 Starting - Preparing the instances for training...
2025-09-19 00:55:11 Downloading - Downloading input data...
2025-09-19 00:55:36 Downloading - Downloading the training image.....
2025-09-19 00:56:43 Training - Training image download completed. Training in progress.....
2025-09-19 01:00:44 Uploading - Uploading generated training model...
2025-09-19 01:00:57 Completed - Training job completed
..Training seconds: 345
Billable seconds: 345
```

logs del job

Trabajos de entrenamiento

Información

🔄

Acciones ▾

Crear trabajo de entrenamiento

🔍

Buscar trabajos de entrenamiento

Nombre ▾

Hora de creación ▾

Duración

Estado del trabajo ▾

Estado secundario del trabajo

Estado del grupo en caliente

Tiempo restante

Actualmente no hay ningún recurso.

(no me aparecio nunca en los trabajos de entrenamiento)


CloudWatch

Overview información

1h 3h 12h 1d 1sem. Personalizado
Zona horaria UTC


Información general
Filtrar por grupo de recursos
Información
Acción

Empezar a usar CloudWatch [Ir a la página de inicio](#)
No tienes alarmas, métricas ni panel predeterminado. Una vez que los configure, se mostrarán aquí.




Configure alarmas en cualquiera de sus métricas para recibir una notificación cuando su métrica exceda el límite especificado.

[Crear alarmas](#)




Cree y asigne un nombre a cualquier panel de CloudWatch **CloudWatch-Default** para mostrarlo aquí.

[Crear un panel predeterminado](#)



Lleve a cabo una monitorización utilizando sus archivos de registro personalizados, de aplicación y de sistema existentes.


[Ver los registros](#)



Escriba reglas para indicar los eventos de interés para la aplicación y las acciones automatizadas que se deben desencadenar.

[Ver los eventos](#)

Comience con las soluciones de observabilidad [Explore soluciones de observabilidad](#)
CloudWatch observability solutions out-of-the-box observability for AWS services and popular workloads. These ready-to-use, customizable solutions are designed to get you up and running quickly with monitoring at AWS.



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

```
WARNING:sagemaker.analytics:Warning: No metrics called test:objective_loss found
WARNING:sagemaker.analytics:Warning: No metrics called test:binary_f_beta found
WARNING:sagemaker.analytics:Warning: No metrics called test:precision found
WARNING:sagemaker.analytics:Warning: No metrics called test:recall found
```

```
import io
#bucket='<LabBucketName>'
bucket='c174660a4519475l11672966t1w637423359651-labbucket-oqbw1d5aisic'
prefix='flight-linear'
train_file='flight_train.csv'
test_file='flight_test.csv'
validate_file='flight_validate.csv'
whole_file='flight.csv'
s3_resource = boto3.Session().resource('s3')

def upload_s3_csv(filename, folder, dataframe):
    csv_buffer = io.StringIO()
    dataframe.to_csv(csv_buffer, header=False, index=False)
    s3_resource.Bucket(bucket).Object(os.path.join(prefix, folder, filename)).put(Body=csv_buffer.getv

INFO:botocore.credentials:Found credentials from IAM Role: BaseNotebookInstanceEc2InstanceRole

def batch_linear_predict(test_data, estimator):
```

(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

```
import xgboost as xgb

# Drop the 'fecha' column from the original X_train and X_test DataFrames
X_train_processed = X_train.drop('fecha', axis=1)
X_test_processed = X_test.drop('fecha', axis=1)

# Re-apply the One-Hot Encoding on the processed data
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', one_hot_encoder, categorical_features)],
    remainder='passthrough')

X_train_encoded = preprocessor.fit_transform(X_train_processed)
X_test_encoded = preprocessor.transform(X_test_processed)

# Instantiate and train the XGBoost classifier
model = xgb.XGBClassifier(objective='binary:logistic', use_label_encoder=False, eval_metric=
model.fit(X_train_encoded, y_train)

print("XGBoost model trained successfully.")
```

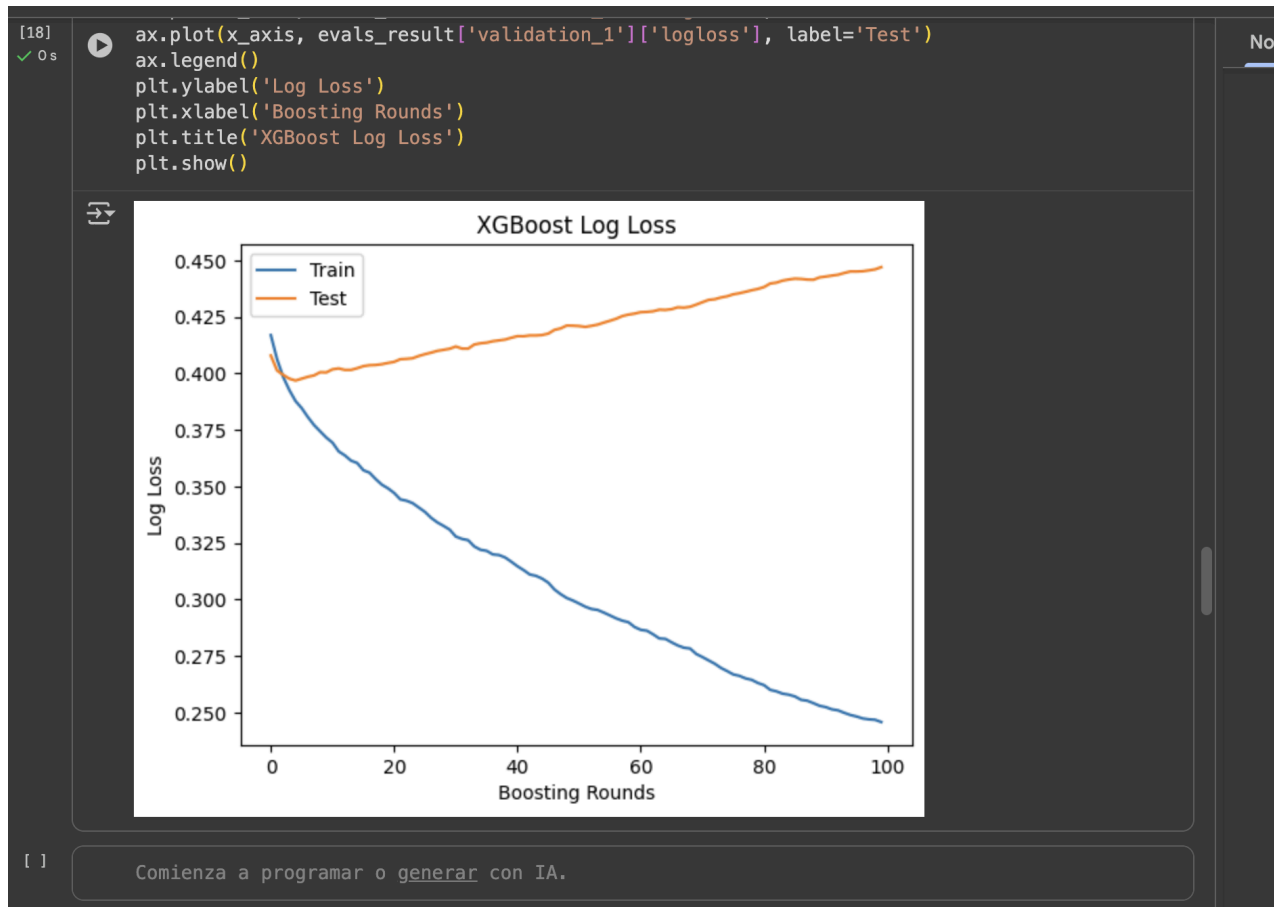
XGBoost model trained successfully.
/usr/local/lib/python3.12/dist-packages/xgboost/training.py:183: UserWarning: [01:32:13] WARN
Parameters: { "use_label_encoder" } are not used.

```
bst.update(dtrain, iteration=i, fobj=obj)
```

Pytho

(abarcare desde las imagenes de evidencia)

GRAFICAS DE METRICAS



EVIDENCIA DE EVALUACION DEL MODELO

```
from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score

y_pred = model.predict(X_test_encoded)
y_pred_proba = model.predict_proba(X_test_encoded)[:, 1]

precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_pred_proba)

print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")
print(f"AUC: {auc:.4f}")
```

→ Precision: 0.2462
Recall: 0.0539
F1-Score: 0.0884
AUC: 0.6330

TABLA CON EJEMPLOS REALES APLICADOS AL MODELO ENTRENADO

Example Predictions:

	fecha	hora_salida	aerolinea	origen	destino	distancia	Predicted Probability	Predicted Class	Actual Target
6252	2025-01-20	05:49:04.745864	4	Airline B	City B	City E	653.654483	0.013765	0
4684	2025-06-29	02:56:43.669623	12	Airline A	City B	City E	3351.784792	0.059754	0
1731	2025-05-04	18:50:47.725574	5	Airline C	City A	City F	328.696954	0.001310	0
4742	2024-10-02	01:17:22.150951	14	Airline D	City E	City A	2023.507742	0.147630	0
4521	2025-04-17	17:03:52.942463	3	Airline D	City B	City F	1455.062707	0.003318	0

```
# Predict the delay probability and class for the selected samples
predicted_prob = model.predict_proba(sample_processed_encoded)[: , 1]
predicted_class = model.predict(sample_processed_encoded)

# Get the actual target values for the samples
actual_targets = y_test.iloc[sample_indices].values

# Create a DataFrame to display the results
results_df = sample_original_features.copy()
results_df['Predicted Probability'] = predicted_prob
results_df['Predicted Class'] = predicted_class
results_df['Actual Target'] = actual_targets

print("Example Predictions:")
display(results_df)
```

Example Predictions:

	fecha	hora_salida	aerolinea	origen	destino	distancia	Predicted Probability	Predicted Class
6252	2025-01-20	05:49:04.745864	4	Airline B	City B	City E	653.654483	0.013765
4684	2025-06-29	02:56:43.669623	12	Airline A	City B	City E	3351.784792	0.059754
1731	2025-05-04	18:50:47.725574	5	Airline C	City A	City F	328.696954	0.001310
4742	2024-10-02	01:17:22.150951	14	Airline D	City E	City A	2023.507742	0.147630
4521	2025-04-17	17:03:52.942463	3	Airline D	City B	City F	1455.062707	0.003318

(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.