MLflow Github Actions

Javier Callejas Cardozo

Universidad Ean

## Resumen del experimento

En este ejercicio se utilizó el conjunto de datos flight\_delays, disponible dentro del entorno de Azure Machine Learning Studio. Este dataset contiene información histórica de vuelos, incluyendo variables como el año, mes, día de la semana, aerolínea, aeropuertos de origen y destino, horarios programados y retrasos registrados.

El objetivo del experimento fue predecir la probabilidad de que un vuelo se retrase en su salida, utilizando técnicas de Machine Learning supervisado. Para ello, se aplicó un flujo completo de procesamiento y modelado dentro del Diseñador de Azure, el cual permite construir visualmente un pipeline de aprendizaje automático.

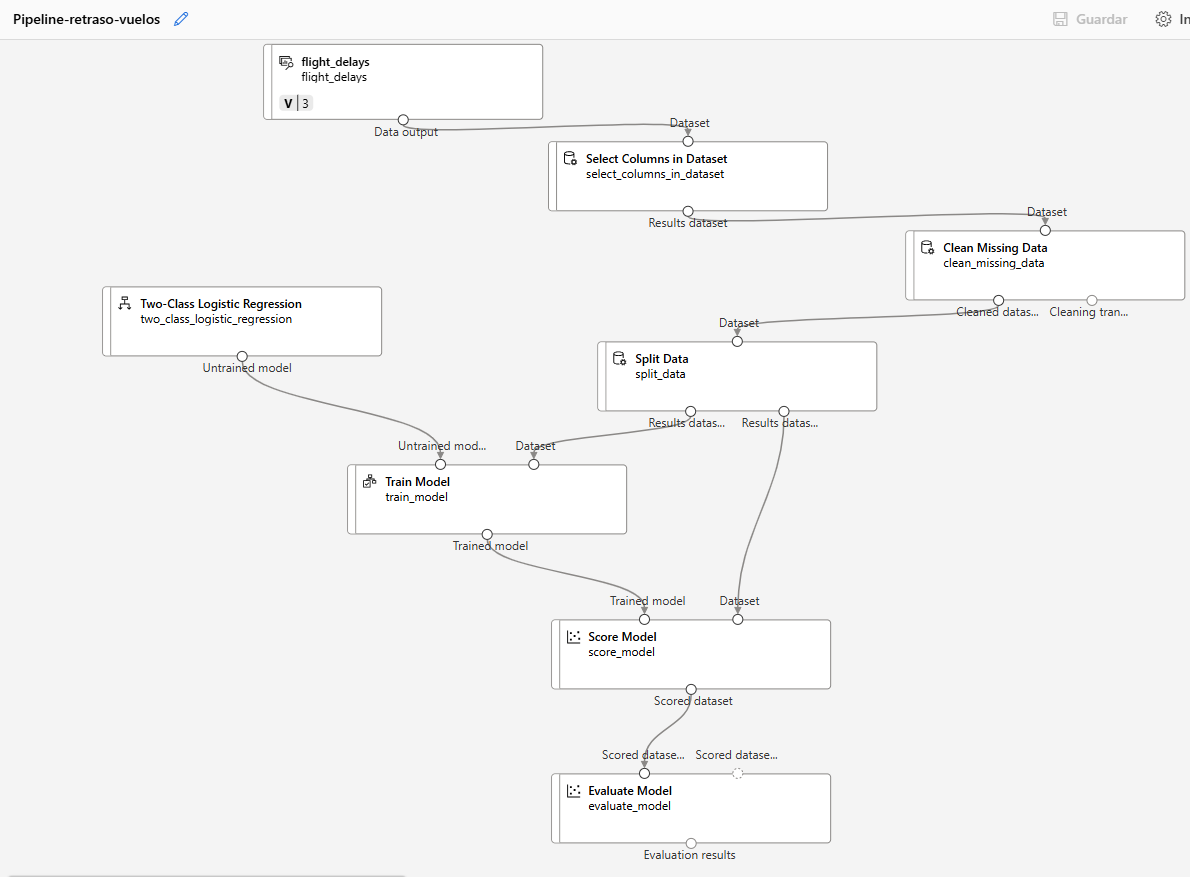
### Configuración y flujo del pipeline

El flujo de trabajo incluyó los siguientes pasos:

1. Selección del dataset: Se utilizó el conjunto de datos flight\_delays como fuente de información.
2. Limpieza de datos: Se empleó el módulo Clean Missing Data para eliminar o reemplazar valores nulos, garantizando la consistencia del conjunto de datos.
3. División del dataset: Los datos fueron divididos en dos subconjuntos, 70% para entrenamiento y 30% para prueba, utilizando el módulo Split Data.
4. Selección del algoritmo: Se aplicó el modelo Two-Class Logistic Regression, que es adecuado para problemas de clasificación binaria (en este caso: vuelo retrasado = 1, vuelo a tiempo = 0).
5. Entrenamiento y evaluación: El modelo fue entrenado usando el módulo Train Model, tomando como variable objetivo la columna DepDel15, que indica si un vuelo se retrasó más de 15 minutos.  
   Posteriormente, se evaluó el desempeño mediante los módulos Score Model y Evaluate Model, comparando las predicciones con los valores reales.

### Conclusiones

El pipeline permitió construir un modelo predictivo capaz de detectar retrasos en vuelos a partir de datos históricos. La regresión logística resultó una buena opción, ya que su salida probabilística permite estimar la probabilidad de retraso de manera clara y eficiente.



A continuación, se muestran dos imágenes del pipeline ejecutado:

