Entrenamiento y evaluación de Machine Learning usando PySpark

Omar Gaston[1]
[1] # *Departamento de Ingeniería en Sistemas, Pontificia Universidad Javeriana*
11 de noviembre de 2025, Bogotá, Colombia.
[1] oy-gastonc@javeriana.edu.do

1 Introducción

Este documento presenta una visión general de los tres notebooks desarrollados como parte del flujo completo de un proyecto de Machine Learning. Cada notebook aborda una fase fundamental del ciclo de vida del modelo: preprocesamiento de datos, modelado supervisado y modelado no supervisado. El objetivo principal es ilustrar el proceso completo, desde la carga y transformación inicial de los datos hasta la evaluación de modelos y la visualización de resultados.

2 Notebook: Preprocesamiento de Datos

http://10.43.100.120:8008/notebooks/talleres-omar/ML_Preprocesamiento_Omar.ipynb

El primer notebook, **ML_Preprocesamiento_Omar.ipynb**, se centra en preparar el conjunto de datos para su uso en tareas de aprendizaje automático. Esta fase es crucial, ya que la calidad del preprocesamiento influye directamente en el desempeño de modelos posteriores. En este notebook se realiza:

- Exploración inicial del dataset.
- Limpieza de datos: manejo de valores nulos, duplicados y ruido.
- Codificación de variables categóricas.
- Normalización y estandarización de características numéricas.
- División del dataset en subconjuntos de entrenamiento y prueba.
- Visualización de distribuciones y correlaciones relevantes.

El resultado final de este notebook es un dataset limpio, consistente y estructurado, listo para alimentar los modelos supervisados y no supervisados.

3 Notebook: Aprendizaje Supervisado

http://10.43.100.120:8008/notebooks/talleres-omar/ML_Supervisado_Omar.ipynb

El segundo notebook, **ML_Supervisado_Omar.ipynb**, aborda la construcción de modelos basados en aprendizaje supervisado con PySpark. El enfoque está en predecir correctamente la variable objetivo utilizando algoritmos clásicos adaptados a entornos distribuidos. En este notebook se realiza:

- Carga del dataset preprocesado desde el notebook anterior.
- Entrenamiento de varios modelos supervisados (por ejemplo: regresión logística, árboles de decisión, random forest).

- Evaluación del modelo mediante métricas como:
  - *Accuracy*
  - *Precision*
  - *Recall*
  - *F1-score*
- Construcción y visualización de la **matriz de confusión**.
- Interpretación de resultados y comparación entre modelos.

Este notebook proporciona una visión clara del rendimiento predictivo de distintos algoritmos y permite seleccionar el modelo más adecuado para el problema.

4 Notebook: Aprendizaje No Supervisado

http://10.43.100.120:8008/notebooks/talleres-omar/ML_NoSupervisado_Omar.ipynb

El tercer notebook, **ML_NoSupervisado_Omar.ipynb**, explora técnicas de aprendizaje no supervisado con el objetivo de descubrir patrones ocultos en los datos sin utilizar etiquetas. En este notebook se realiza:

- Carga y preparación del dataset.
- Aplicación de algoritmos de clustering como K-Means.
- Reducción de dimensionalidad para visualización (PCA).
- Construcción de gráficos 2D que representan los clusters descubiertos.
- Evaluación mediante métricas como *Silhouette Score*.
- Comparación visual y analítica de los grupos generados.

Este notebook permite comprender la estructura interna de los datos y evaluar la coherencia de los clusters identificados.

5 Conclusiones

Los tres notebooks conforman un flujo completo de trabajo en Machine Learning, permitiendo entender el dataset, transformarlo adecuadamente, probar modelos de clasificación supervisada, explorar patrones mediante aprendizaje no supervisado.

Algunos puntos importantes fueron:

- En datasets pequeños, algunas operaciones pueden parecer más lentas o innecesariamente costosas. Por esta razón, fue necesario ajustar la configuración de la sesión Spark.

- Los notebooks pueden volverse inestables si se ejecutan simultáneamente. Spark utiliza un solo SparkSession activa por proceso; cuando múltiples celdas intentan crear sesiones o lanzar cargas pesadas al mismo tiempo.

- Se utilizaron librerías nativas o compatibles con PySpark. También se ajustó la configuración para mostrar correctamente gráficos 2D dentro del entorno notebook.

Este proyecto puede servir como base para desarrollos más complejos, análisis avanzados o implementación de pipelines productivos con PySpark.