Documentación Técnica - MLOPs Data Bricks

Configuración del cluster de Databricks

El primer paso es configurar el cluster de Databricks, para ello se utilizó una máquina con las siguientes características:

- 1 Driver
- 8 GB Memory, 4 Cores

La sustentación del porqué se utilizaron esas características es debido a que el dataset es pequeño, y los modelos utilizados tampoco requieren mayor cantidad de recursos.

En las siguientes gráficas podemos observar el uso de recursos del cluster de Databricks.

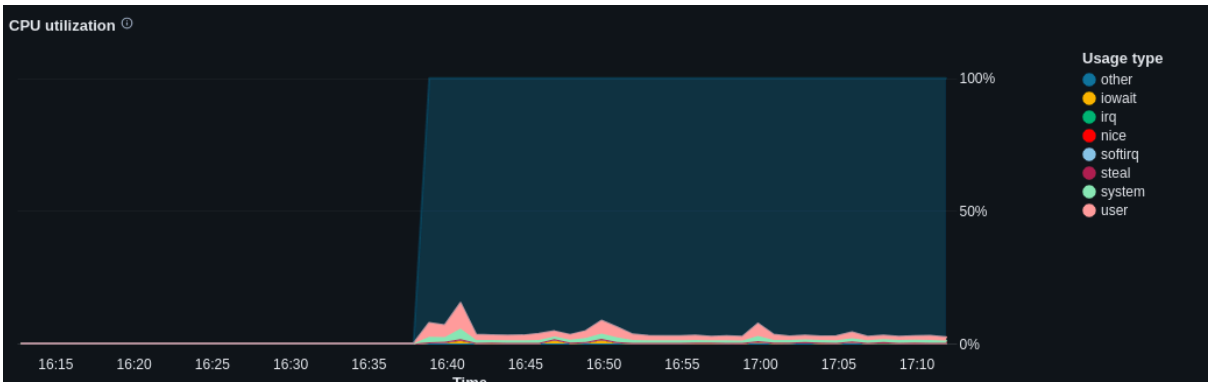

Figura 1: Uso de CPU

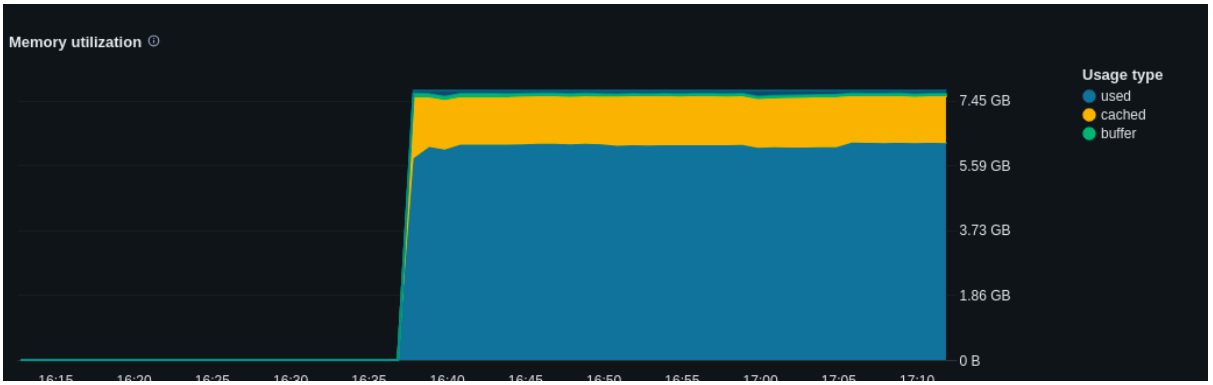

Figura 2: Uso de Memoria

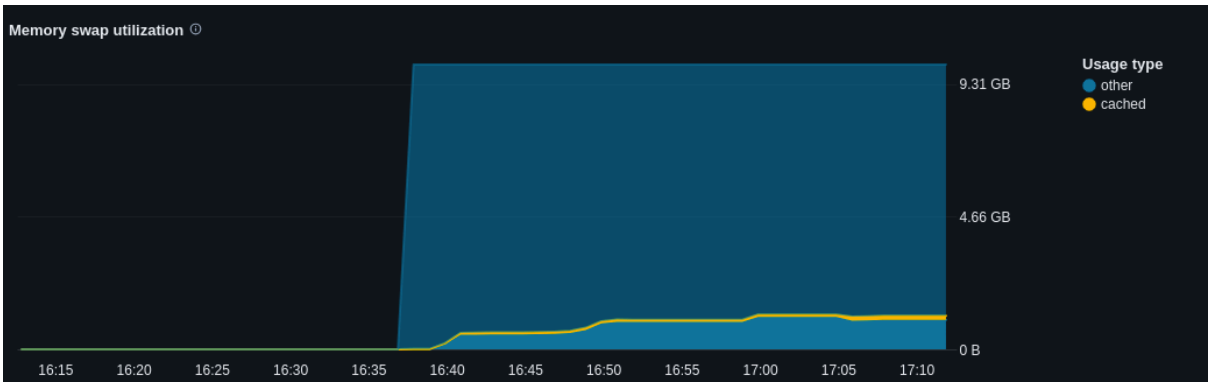

Figura 3: Uso de intercambio de memoria

Análisis Exploratorio de Variables (EDA)

Se puede observar que en el dataset no hay datos nulos (Figura 4)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       178 non-null    int64
 1   1       178 non-null    float64
 2   2       178 non-null    float64
 3   3       178 non-null    float64
 4   4       178 non-null    float64
 5   5       178 non-null    int64
 6   6       178 non-null    float64
 7   7       178 non-null    float64
 8   8       178 non-null    float64
 9   9       178 non-null    float64
 10  10      178 non-null    float64
 11  11      178 non-null    float64
 12  12      178 non-null    float64
 13  13      178 non-null    int64
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
```

Figura 4: Análisis de datos nulos

En la Figura 5 podemos observar los valores atípicos. La variable con el índice 13 (Proline) tiene un rango de variación mayor en comparación con las demás variables. Ésto puede provocar una escala desigual y por lo tanto puede afectar la observabilidad de patrones en las demás variables. Se pueden observar datos atípicos en las siguientes variables:

- 2 (Malic acid)

- 3 (Ash)
- 4 (Alcalinity of ash)
- 5 (Magnesium)
- 9 (Proanthocyanins)
- 10 (Color intensity)
- 11 (Hue)

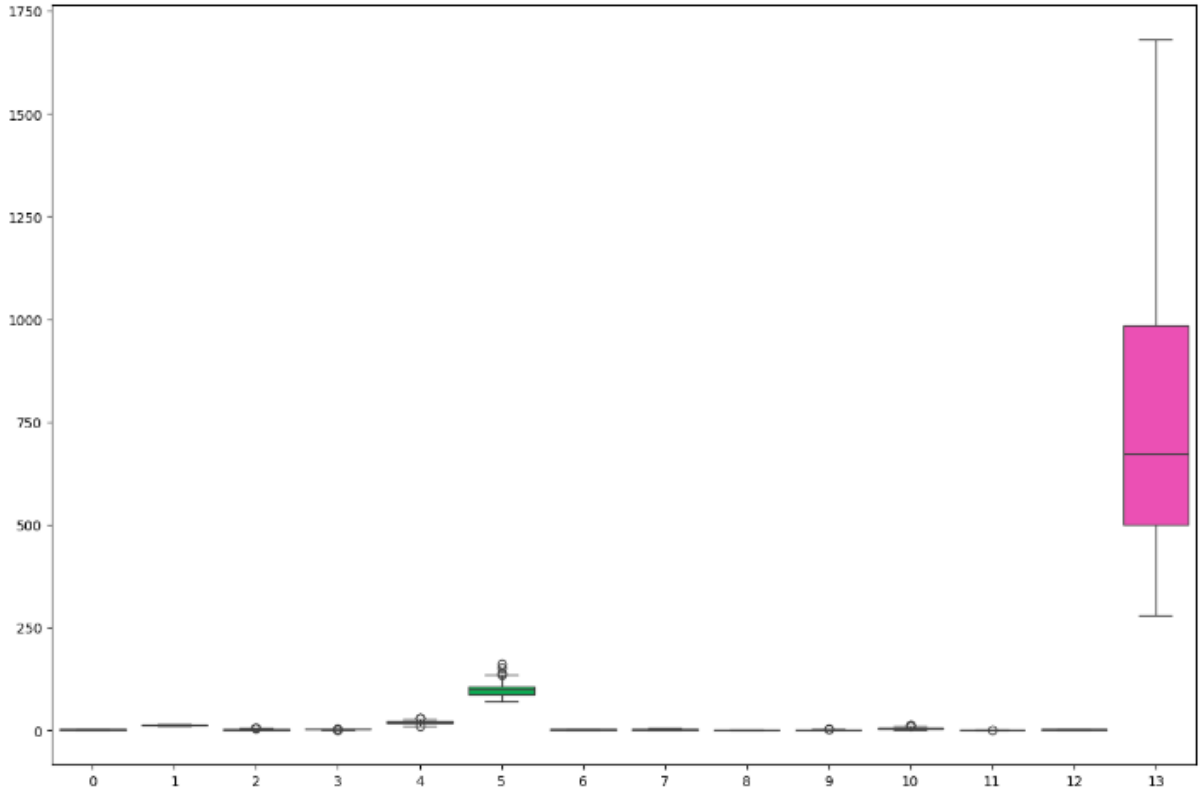

Figura 5: Análisis de datos valores atípicos

Realizando cálculos, podemos observar que las variables con más datos atípicos son las siguientes:

- 4 (Alcalinity of ash)
- 5 (Magnesium)
- 10 (Color intensity)

Con 4 datos atípicos cada una.

Tomando como ejemplo la variable 5 (Magnesium), podemos observar los valores atípicos en la Figura 6.

Observando la Figura 7, podemos ver que las clases del datasets están relativamente balanceadas, sin embargo hay una diferencia de aproximadamente el 12.92% entre la clase mayoritaria y la clase minoritaria

- Clase 2 -> 71 datos -> 39.88% del dataset
- Clase 1 -> 59 datos -> 33.14% del dataset
- Clase 3 -> 48 datos -> 26.96% del dataset

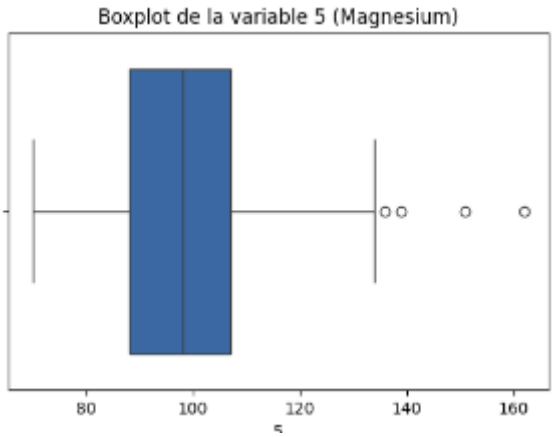

Figura 6: Valores atípicos de la variable 5 (Magnesium)

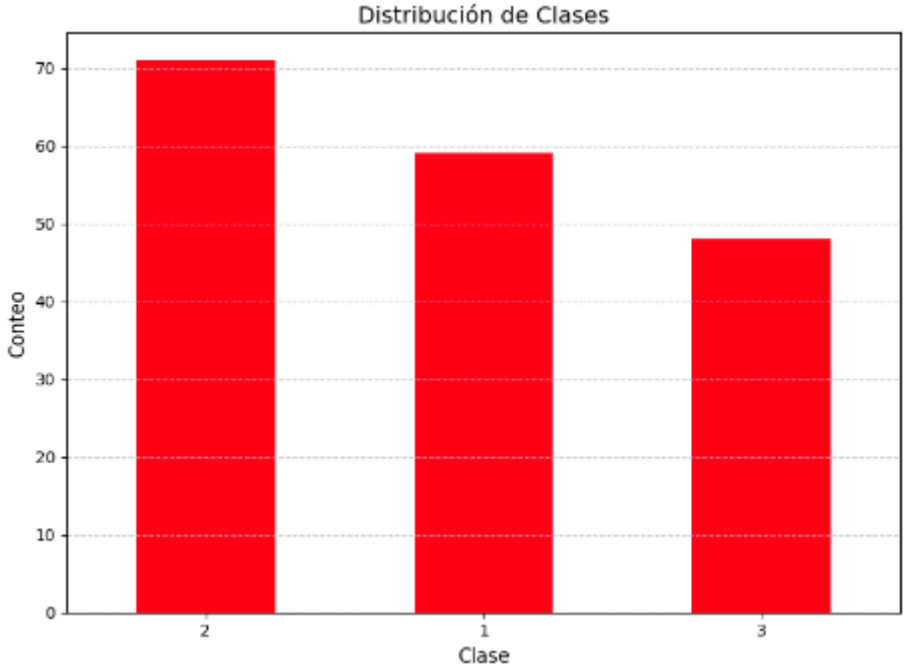

Figura 7: Distribución de clases del dataset

Modelamiento de Datos

Para el modelamiento de datos se utilizó un módulo llamado **Trainer** para facilitar la implementación de distintos modelos y experimentos.

Módulo Trainer

La clase **`Trainer`** es un contenedor modular que permite:

1. **Preprocesar los datos** (escalado y preparación de características y etiquetas).
2. **Dividir el dataset** en subconjuntos de entrenamiento y prueba.
3. **Entrenar y evaluar un modelo** utilizando métricas como accuracy, precisión, recall, y F1-score.
4. **Registrar el modelo y las métricas** en MLflow.
5. **Visualizar la matriz de confusión** para interpretar los resultados.

El diagrama de la Figura 8 ilustra las principales responsabilidades de la clase, distribuyendo las tareas entre los datos, el modelo y las métricas.

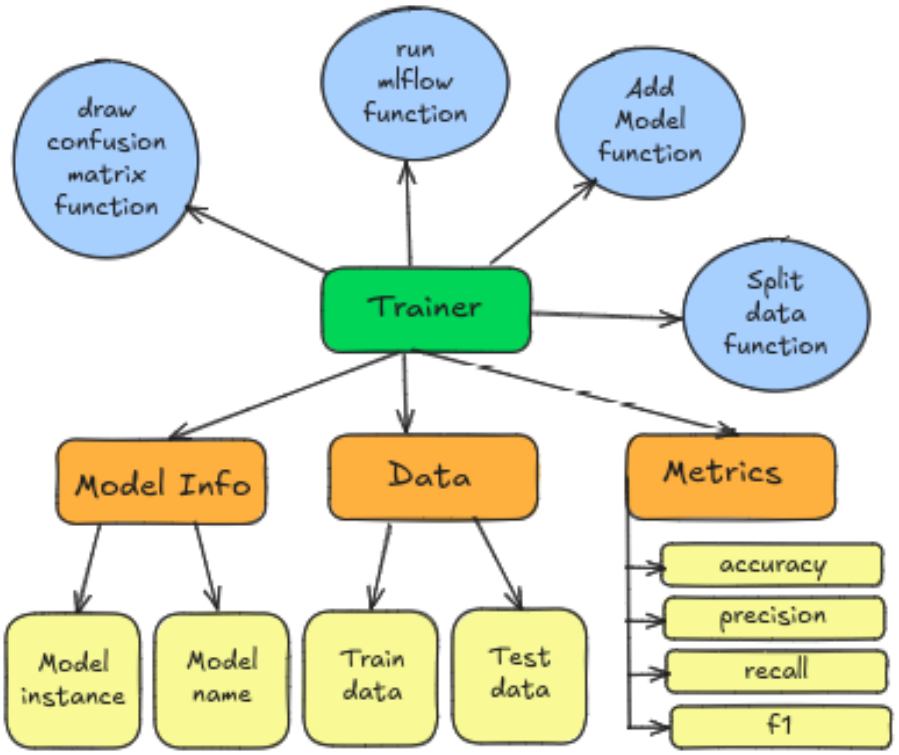

Figura 8: Módulo Trainer

Resultados XGBoost

Los resultados de la Figura 9 muestran las métricas de rendimiento con una exactitud (accuracy) general de 0.94, lo que indica que clasifica correctamente el 94% de las instancias en el conjunto de prueba. La clase 0 tiene una precisión perfecta (1.00), lo que significa que todas las predicciones positivas para esta clase son correctas. Sin embargo, el

recall es ligeramente menor (0.93), lo que indica que algunos ejemplos reales de esta clase no fueron identificados. La clase 1 tiene un recall perfecto (1.00), lo que significa que todos los ejemplos reales de esta clase fueron correctamente identificados, pero la precisión es menor (0.88), lo que sugiere algunos falsos positivos. La clase 2 tiene métricas balanceadas con precisión y recall de 1.00 y 0.88 respectivamente, lo que indica que identifica la mayoría de los ejemplos correctamente, pero algunos no son detectados.

	precision	recall	f1-score	support
0	1.00	0.93	0.96	14
1	0.88	1.00	0.93	14
2	1.00	0.88	0.93	8
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

Figura 9: Tabla de resultados XGBoost

Resultados Regresión Logística

En los resultados de la Figura 9 se observa un excelente rendimiento global con una exactitud (accuracy) de 0.98, lo que significa que el modelo clasifica correctamente el 98% de los ejemplos en el conjunto de prueba. Para la clase 0, tanto la precisión, el recall y el F1-score son perfectos (1.00), lo que indica que el modelo identifica correctamente todos los ejemplos de esta clase sin cometer errores. La clase 1 tiene una precisión perfecta (1.00) y un recall de 0.95, lo que significa que aunque todos los ejemplos identificados como pertenecientes a esta clase son correctos, hay un pequeño porcentaje de ejemplos reales de esta clase que no fueron capturados. La clase 2 tiene un recall perfecto (1.00) y una precisión de 0.93, lo que indica que todos los ejemplos de esta clase fueron identificados correctamente, aunque algunos ejemplos clasificados como esta clase fueron falsos positivos.

0	1.00	1.00	1.00	19
1	1.00	0.95	0.98	21
2	0.93	1.00	0.97	14
accuracy			0.98	54
macro avg	0.98	0.98	0.98	54
weighted avg	0.98	0.98	0.98	54

Figura 10: Tabla de resultados Regresión Logística

Resultados Random Forest

Los resultados de la Figura 11 muestran un rendimiento perfecto del modelo, con una precisión, recall y F1-score de 1.00 para todas las clases y métricas globales. Esto implica que el modelo clasificó correctamente todas las instancias en el conjunto de prueba, sin cometer ningún error.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19
1	1.00	1.00	1.00	21
2	1.00	1.00	1.00	14
accuracy			1.00	54
macro avg	1.00	1.00	1.00	54
weighted avg	1.00	1.00	1.00	54

Figura 11: Tabla de resultados Random Forest

Resultados Decision Trees

Los resultados de la Figura 12 muestran un rendimiento global aceptable del modelo con una exactitud (accuracy) de 0.92, lo que significa que el modelo clasifica correctamente el 92% de las instancias en el conjunto de prueba. Se observa también que presenta problemas para identificar correctamente los ejemplos de la clase 2, como se refleja en su bajo recall (0.62). Sin embargo, el análisis detallado por clase revela un desequilibrio en el desempeño, especialmente en la clase 2 que en este caso es la clase mayoritaria.

	precision	recall	f1-score	support
0	0.93	1.00	0.97	14
1	0.88	1.00	0.93	14
2	1.00	0.62	0.77	8
accuracy			0.92	36
macro avg	0.94	0.88	0.89	36
weighted avg	0.93	0.92	0.91	36

Figura 12: Tabla de resultados Decision Trees

Conclusiones

A Pesar que los resultados se pueden considerar favorables en la mayoría de los modelos, llegando a un 100% de rendimiento en todas las métricas en algunos modelos, se considera

que no se puede llegar a una conclusión fiel al problema citando el teorema No Free Lunch de David Wolpert. Debido a que la cantidad de datos con las que se entrenaron los modelos es limitada, es difícil concluir que los modelos se ajusten en un 100% al problema per se. Sin embargo, los modelos con mejores resultados en este experimento fueron Random Forest y Regresión Logística.

Referencias

- https://scikit-learn.org/stable/
- Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.