Conclusiones:

* Tras realizar el análisis exploratorio de los datos, podemos concluir que estos se encuentran en un formato adecuado para la construcción del modelo y que cumplen con las características óptimas en cuanto a las variables predictoras. Sin embargo, es necesario llevar a cabo un proceso de estandarización, ya que los datos No están en una escala uniforme esto se observa en los gráficos y estadísticas del dataframe planteados y validados en el notebook de Databricks, lo que podría generar ruido al construir el modelo por lo cual se realiza un proceso de estandarización para lograr que todos los datos tengan una data estandarizada con media cero y desviación estándar 1.

Además, se observa un desbalanceo moderado en los datos de las clases 2 y 3, por lo cual se procederá a balancear los datos.

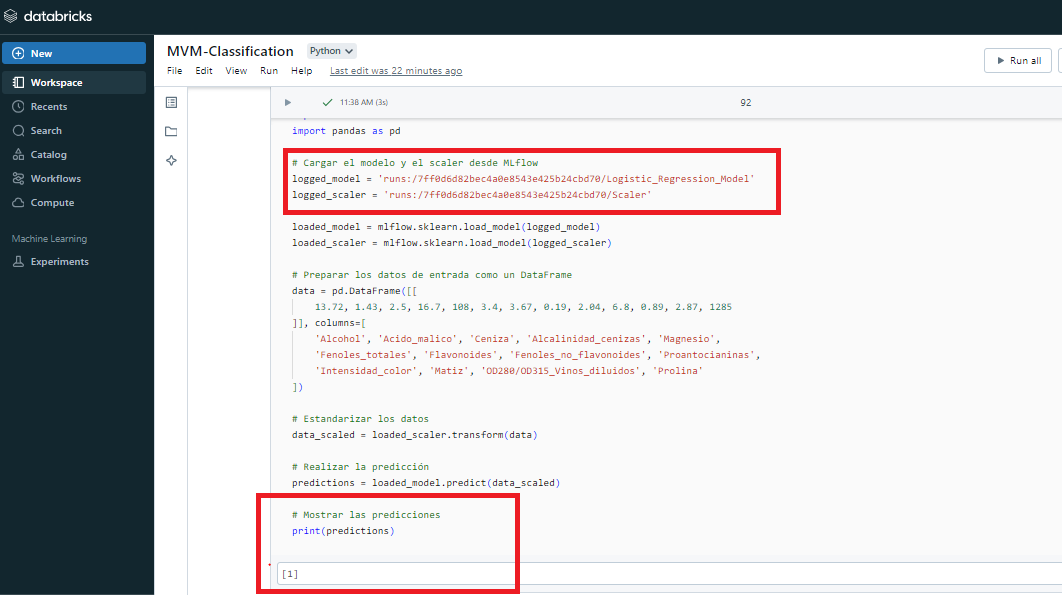
Para la validación del desbalanceo de datos se plantea un rango de desbalanceo en el dataset de la siguiente manera: Desbalanceo Leve (Entre en 20 y 30 %) el cual se concidera aceptable para utilziar en el modelo Desbalanceo Moderado (Entre en 30 y 50 %) el cual se concidera que puede tener un impacti en el rendimiento del modelo Desbalanceo Severo (Mayor a 50 %) el cual se concidera que puede tener un impacto en el rendimiento del modelo

Dado a este planteamiento y viendo el porcentaje de diferencia entre las clases 1,2 y se considera que existe un desbalanceo moderado entre la clase 2 y3, pero ya que supera en 2.39 punto el umbral propuesto se plantea aplicar la tecnica de Balanceo de Data mediante la técnica de SMOTE Synthetic Minority Over-sampling Technique la cual es una técnica de sobre muestreo utilizada para la trata de desbalanceo de datos, el cual genera nuevas muestras sintéticas para aquellas clases minoritarias ayudando a tener un equilibrio en el numero de ejemplos de las diferentes clases. En este técnica se utilizan los K vecinos mas cercanos para asi poder generar las muestras sintéticas, este parametro se configura para generar data balanceada.

* Una vez validado cada modelo podemos observar que dada la poca cantidad de datos algunos modelos no convergen y tienden a generar un overfitting, por lo cual se toma la decisión de implementar modelos mas simples pero de igual de poderosos como lo es la regresión logística el cual se adapta muy bien a los datos presentes en el dataset y permite disminuir el consumo de recursos computacionales brindando asi tiempos mas cortos de entrenamiento y generando excelentes predicciones.
* Podemos observar que como deuda técnica este modelo puede ser publicado mediante un api y una interfaz grafica para poder ser consumido por usuarios finales, esto lo podríamos lograr mediante la implementación de framework de fastapi con Python y un contenedor de Docker el cual puede ser desplegado en ACI de azure.
* Otra deuda técnica podría ser en mejorar los modelos generando data sintética la cual permita tener un dataset mucho mas robusto y con una mayor cantidad de data para asi generar un mejor entrenamiento de los modelos y poder implementar modelos más robustos.
* Al momento de ejecutar el modelo predictivo con los datos brindado se generan los siguientes resultados:

[13.72,1.43,2.5,16.7,108,3.4,3.67,0.19,2.04,6.8,0.89,2.87,1285]

**Predicion : Tipo vino 1**



[12.37,0.94,1.36,10.6,88,1.98,0.57,0.28,0.42,1.95,1.05,1.82,520 –

**Predicción : Tipo vino 2**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Errores registrados: al momento de registrar los diferentes artefactos en mlflow, se me olvido registrar el escalador de StandarScaler. Lo cual genero un error al momento de realizar las predicciones de los dos muestras de vinos ya que el modelo funcionada adecuadamente, pero debíamos scalar los datos ya que en todo el proceso de construcción del modelo este fue un paso que se implemento para los datos de entrenamiento y testeo al momento de la construcción del modelo el cual fue scalar los datos a una media cero y una desviación estándar 1, este error se logro corregir para asi poder logar una correcta predicción de las 2 muestras brindas al final del desarrollo del proyecto, con lo cual se pudo cargar el modelo y el scalador construido logrando asi generar las predicciones con los modelos y métricas registradas en mlflow.