Momento de Retroalimentación: Módulo 2 Implementación de una técnica de aprendizaje máquina sin el uso de un framework. (Portafolio Implementación)

Introducción La base de datos inicial contenía variables no numéricas, lo que requirió la aplicación de un label encoder para convertirlas en numéricas.

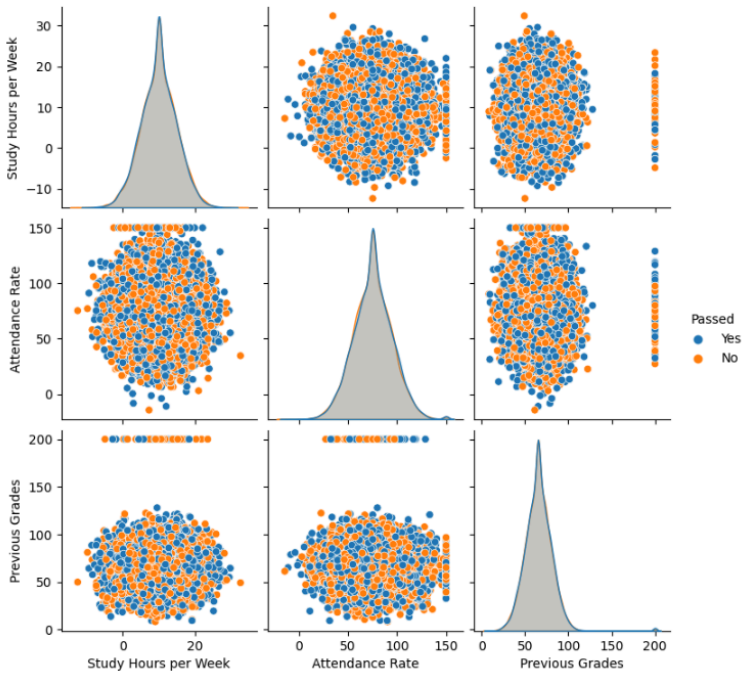
Limpieza de la base de datos Después de convertir las variables en numéricas, se realizó un análisis estadístico que reveló varios problemas:

- Datos faltantes: Había muchos valores NaN (Not a Number) en la base de datos.
- Datos atípicos: Se encontraron valores que no tenían sentido, como porcentajes de asistencia negativos.

Para tratar estos problemas, se tomaron las siguientes medidas:

- Quitar outliers: Se eliminaron los valores atípicos que no seguían la tendencia general de la base de datos.
- Rellenar NaN: Se rellenaron los valores faltantes con un método adecuado

Análisis de resultados Después de la limpieza de la base de datos, se esperaba encontrar una relación clara entre las variables y la variable a predecir. Sin embargo, no se encontró una relación clara entre ellas.



Función Sigmoide: La función sigmoide convierte las predicciones en probabilidades entre 0 y 1. Esta conversión es fundamental para la regresión logística, ya que permite interpretar las predicciones como probabilidades de que un estudiante apruebe. La función sigmoide asegura que el modelo devuelva valores adecuados para la clasificación binaria.

Función de Costo: La función de costo evalúa la diferencia entre las predicciones del modelo y las etiquetas reales. Calcula cuán bien se ajusta el modelo a los datos observados. El objetivo durante el entrenamiento es minimizar esta diferencia, es decir, reducir el costo, para que las predicciones del modelo sean lo más precisas posible.

Optimización: El algoritmo de descenso por gradiente se utiliza para ajustar los parámetros del modelo. En cada iteración, calculo el gradiente, que indica cómo deben cambiar los parámetros para reducir el costo. La tasa de aprendizaje (alpha) determina el tamaño de los ajustes en cada iteración. Un valor adecuado para la tasa de aprendizaje es crucial para garantizar que el modelo converja a una solución óptima sin ser demasiado lento o inestable. El número de iteraciones asegura que el modelo tenga suficiente tiempo para aprender de los datos.

Evaluación:

```
  import pandas as pd
Precisi�n del modelo: 0.4981132075471698
Precisi�n (precision): 0.49802044342067375
Recall: 0.9987008300252617
F1 Score: 0.6646173058911117
Matriz de Confusi�n:
TP: 13837, TN: 23, FP: 13947, FN: 18
Accuracy: 0.4981132075471698
```

El modelo claramente no tuvo los valores que se buscaban pero conociendo el dataset y lo mal implementado que estaba no me sorprendió ver estos valores, podemos ver que hay un sesgo demasiado grande hacia FP, es decir estudiantes que no aprobaron y el modelo marco que sí, esto puede ser debido a que en general todas las variables estaban en rasgos muy similares y se distribuyen de una manera muy normal sin tener correlación con la variable a predecir, sinceramente me pregunté si a lo mejor esto era porque la función lo los parámetros no eran los adecuado, por lo que decidí probar varios de estos y ninguno dio resultado, fui un poco más allá y probe con modelos más especializados, pero los resultados fueron los mismos.

	Accuracy	F1 Score	Precision	Recall
Logistic Regression	0.504582	0.454599	0.500654	0.416304
Decision Tree	0.493621	0.486890	0.489385	0.484420
Random Forest	0.492902	0.490430	0.488841	0.492029
Gradient Boosting	0.497754	0.497573	0.493757	0.501449
Support Vector Classifier	0.503863	0.476984	0.499802	0.456159
Neural Network	0.497754	0.591315	0.495710	0.732609
XGBoost	0.496855	0.499285	0.492938	0.505797
LightGBM	0.500629	0.501346	0.496623	0.506159
CatBoost	0.498113	0.494113	0.494024	0.494203

Por lo que puedo concluir que probablemente la base de datos era inventada y con valores muy aleatorios que no tenían ningún tipo de sentido al intentar predecir si los estudiantes pasaron o no.