Momento de Retroalimentación: Módulo 2 Análisis y Reporte sobre el desempeño del modelo. (Portafolio Análisis)

Este análisis sobre el el desempeño del modelo sobre un dataset en específico, se basa en la implementación de Logistic Regression sobre el dataset de Iris. El modelo es capaz de predecir la clase de flores Iris en función de sus características.

La implementación de este algoritmo puede ser encontrado en:
https://github.com/Caceres-A01706972/TC3006C-Mod2-Framework.git

Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train/Test/Validation):

En la implementación del modelo, se realiza una división de los datos de iris.data en conjuntos de entrenamiento y prueba utilizando la función de **train_test_split** de *Scikit-Learn*.

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

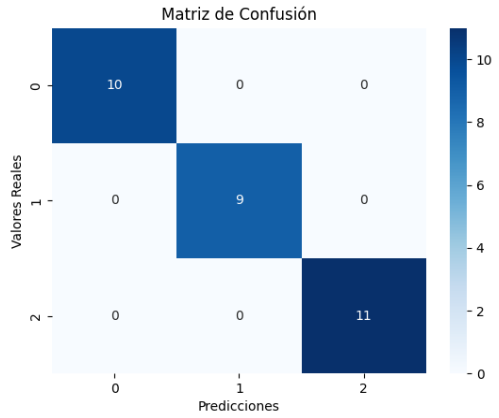
La función de *Scikit-Learn* toma los features, el target (en este caso la clase de la flor), el test_size que especifica la proporción del conjunto de datos que se va a utilizar como conjunto de test. En este caso, se estableció un test_size de 0.2 (significa que el 20% del conjunto de datos será utilizado para ser el test, mientras que el otro 80% será el test).

Diagnóstico y explicación del grado de bias o sesgo (bajo, medio, alto):

El modelo tiene un bajo sesgo, ya que alcanza una precisión del 100% tanto en el conjunto de entrenamiento como en el conjunto de prueba.

```
Precisión en conjunto de entrenamiento: 97.50%
Precisión en conjunto de prueba: 100.00%
```

Esto sugiere que el modelo puede capturar bien la relación entre las características y las etiquetas.
En la matriz de confusión también podemos observar su alta precisión, de los valores reales con las predicciones del modelo.



Diagnóstico y explicación del grado de varianza (bajo, medio, alto):

El grado de varianza parece ser bajo, ya que el modelo tiene una alta precisión tanto en el conjunto de entrenamiento como en el conjunto de prueba.

```
Precisión en conjunto de entrenamiento: 97.50%
Precisión en conjunto de prueba: 100.00%
```

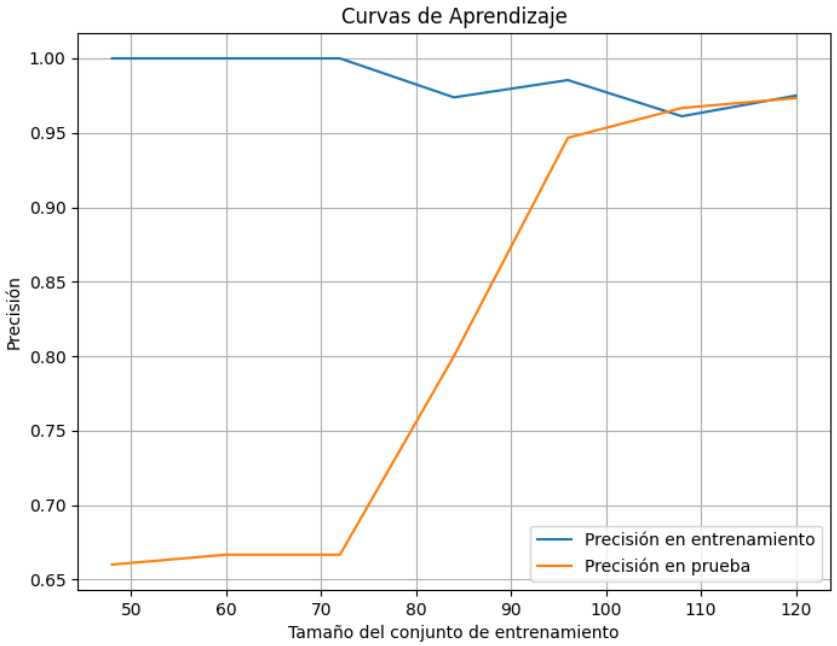
Como el modelo tiene una varianza baja, se generalizará mejor a nuevos datos.
La baja diferencia entre la precisión en ambos conjuntos sugiere que el modelo no está sobre ajustando (alta varianza) los datos de entrenamiento.

Diagnóstico y explicación del nivel de ajuste del modelo (underfit, fit, overfit):

Podemos observar como el modelo no está sobre ajustando, ya que en el conjunto de entrenamiento tiene una buena precisión, y también en el conjunto de test. Y tampoco está subajustado ya que la precisión no es baja en ninguno de los dos (test y train).
Entonces podemos decir que el modelo parece estar bien ajustado ya que logra una alta precisión tanto en el conjunto de train como en el conjunto de test.

Uso de técnicas de regularización o ajuste de parámetros para mejorar el desempeño del modelo:

El modelo de Logistic Regression parece funcionar bien sin necesidad de técnicas adicionales de regularización o ajuste de parámetros.
Esto se refleja en la alta precisión en ambos conjuntos de datos y la falta de signos de sobreajuste.
En el caso de que fuera lo contrario, que se muestre un buen rendimiento en el training pero en el test desempeñe mal, se presentaría varianza alta en el modelo y ya estaría sobre ajustando.
En la siguiente gráfica podemos observar como este fuera el caso si el tamaño del conjunto de entrenamiento fuera menor:



Si esto sucede podemos decir que el modelo ya memorizó los datos de train en lugar de aprender patrones. Y para evitar eso podemos darle más datos de entrenamiento al modelo o también podríamos aplicar técnicas de regularización para controlar la complejidad del modelo al agregar términos de penalización a la función de costo.