



Tecnológico de Monterrey

Inteligencia artificial avanzada para la ciencia de datos I (Gpo 101)

**Momento de Retroalimentación: Módulo 2 Análisis y Reporte sobre el
desempeño del modelo**

11 de Septiembre del 2023

Profesor responsable:
Jorge Adolfo Ramírez Uresti

Aprendizaje Máquina

Trabajo realizado por:

Rodrigo Mejía Jiménez

A01752113

Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Estado de
México, Profesional

1. Justificación del uso de dataset

El dataset seleccionado contiene información sobre diferentes medidas de las flores del género Iris, específicamente las especies Iris-setosa, Iris-versicolor e Iris-virginica. Justificaciones de por qué se utilizó iris.csv:

- El dataset es multiclase ya que incluye 3 clases de flores. Lo que permite desarrollar modelos de machine learning para predecir la especie de una flor de Iris en función de sus medidas.
- El dataset es fácil de comprender ya que consta de medidas numéricas que son las longitudes los pétalos y sépalos.
- El tamaño del dataset es adecuado ya que consta de 150 muestras por lo que no es ni grande ni pequeño. Esto lo hace manejable para tareas de entrenamiento y prueba de modelos de machine learning.
- En resumen, el dataset fue elegido ya que se adecua para el desarrollo en este caso de una red neuronal ya que contiene problema de multiclase y nos ayuda para la predicción dependiendo el tamaño y longitud de los pétalos o de los sépalos.

2. Análisis

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

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

Train: Este conjunto se utiliza para entrenar el modelo. Los datos en este conjunto se utilizan para ajustar los parámetros del modelo mediante el proceso de aprendizaje.

Conjunto de Entrenamiento (Train):

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
22	4.6	3.6	1.0	0.2
15	5.7	4.4	1.5	0.4
65	6.7	3.1	4.4	1.4
11	4.8	3.4	1.6	0.2
42	4.4	3.2	1.3	0.2

Shape: (120, 4)

Test: Este conjunto se utiliza para evaluar el rendimiento del modelo después de que se ha entrenado. Los datos en este conjunto no se utilizan durante el entrenamiento y se utilizan únicamente para realizar predicciones y calcular métricas de evaluación, como la precisión, el recall y el error.

```
Conjunto de prueba (Test):
SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
26             5.0           3.4           1.6           0.4
18             5.7           3.8           1.7           0.3
118            7.7           2.6           6.9           2.3
145            6.7           3.0           5.2           2.3
78             6.0           2.9           4.5           1.5
Shape: (15, 4)
```

Validation: Este conjunto se utiliza para ajustar los hiperparámetros del modelo y evitar el overfitting. Mientras que el conjunto de prueba evalúa el rendimiento del modelo, el conjunto de validación se utiliza para tomar decisiones sobre la configuración del modelo, como la elección de hiperparámetros como sería la tasa de aprendizaje para la red neuronal.

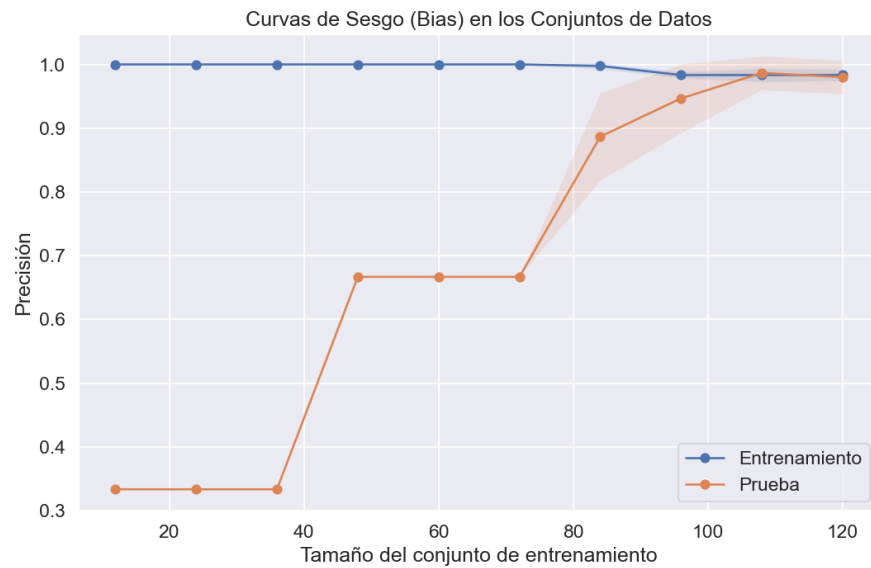
```
Conjunto de validacion (Validation):
SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
143            6.8           3.2           5.9           2.3
56             6.3           3.3           4.7           1.6
128            6.4           2.8           5.6           2.1
69             5.6           2.5           3.9           1.1
68             6.2           2.2           4.5           1.5
Shape: (15, 4)
```

2. Diagnóstico y explicación el grado de bias o sesgo: bajo medio alto

El sesgo (bias) se refiere a la diferencia entre las predicciones y los valores reales de los datos de prueba.

Un alto sesgo indica que el modelo simplifica en exceso la relación de las características de entrada y las salidas deseadas (underfitting)

Un modelo con bajo sesgo representa de manera más precisa la relación y puede hacer predicciones más cercanas a los valores reales.



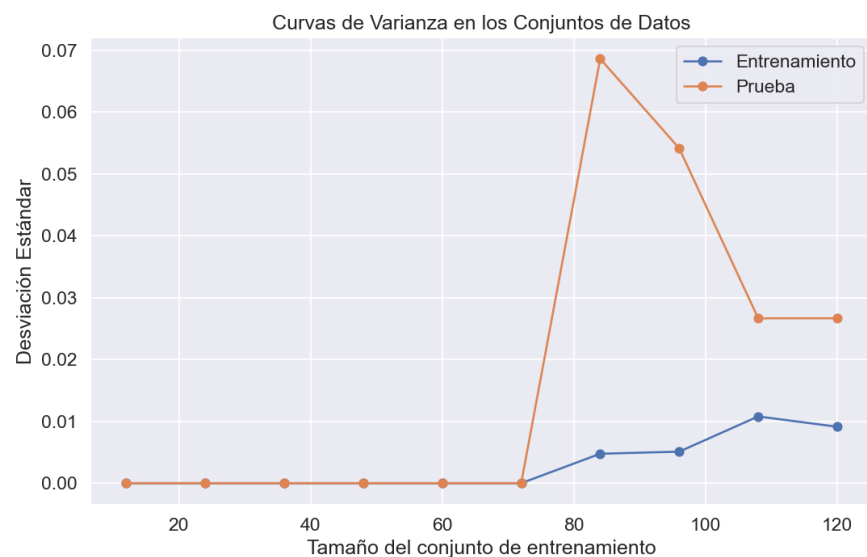
El sesgo obtenido en el modelo realizado es el siguiente:

Sesgo (Bias): 0.0200

El bias se consideraría bajo por lo que no existe underfitting

3. Diagnóstico y explicación el grado de varianza: bajo medio alto

La varianza se refiere a la variabilidad de las predicciones del modelo para diferentes conjuntos de datos de entrenamiento. Una alta varianza implica que el modelo es sensible a las fluctuaciones en los datos de entrenamiento y puede generar overfitting. Por otro lado, un modelo con baja varianza generalmente es más estable pero puede tener un sesgo más alto.



La varianza en el modelo es la siguiente:

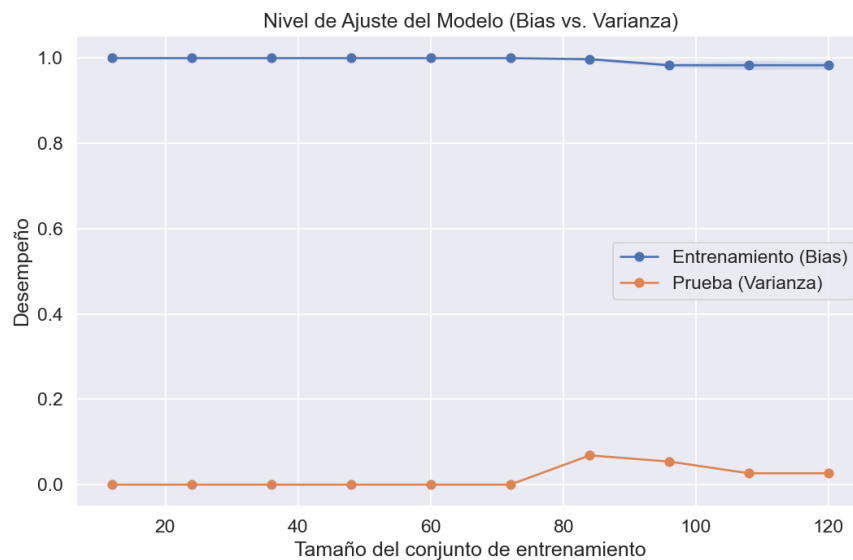
Varianza: 0.0007

La varianza se considera baja por lo que no existe overfitting

4. Diagnóstico y explicación el nivel de ajuste del modelo: underfitt fitt overfitt

Con los valores obtenidos calculando el sesgo y la varianza en el punto donde la diferencia entre train y test se minimiza obtenemos lo siguiente:

Calculo de bias y varianza en el punto de diferencia entre train y test:
Grado de Bias (Sesgo): 0.0167 - Bajo
Grado de Varianza: -0.0033 - Bajo
El modelo tiene buen equilibrio entre el bias y la varianza por lo que existe fitt



Viendo los numeros anteriores, se indica que el modelo tiene un buen equilibrio entre sesgo y varianza. Un sesgo bajo implica que el modelo se ajusta bien a los datos de entrenamiento, y una varianza baja indica que el modelo generaliza bien a datos no vistos.

3. Basándote en lo encontrado en tu análisis utiliza técnicas de regularización o ajuste de parámetros para mejorar el desempeño de tu modelo y documenta en tu reporte cómo mejoró este.

El reporte de resultados con el primer modelo de red neuronal diseñado arrojó los siguientes resultados:

Red neuronal con dos capas ocultas con 10 neuronas cada una

```
clasificador_red = MLPClassifier(hidden_layer_sizes=(10, 10), max_iter=1000, random_state=42)
```

```
Classification report:
              precision    recall  f1-score   support

     0           1.00        1.00        1.00         10
     1           0.89        0.89        0.89          9
     2           0.91        0.91        0.91         11

 accuracy          0.93
macro avg          0.93        0.93        0.93         30
weighted avg       0.93        0.93        0.93         30
```

Sesgo (Bias): 0.0333

Varianza: 0.0009

Calculo de bias y varianza en el punto de diferencia entre train y test:

Grado de Bias (Sesgo): 0.0300 - Bajo

Grado de Varianza: -0.0033 - Bajo

El modelo tiene buen equilibrio entre el bias y la varianza por lo que existe fitt

Ahora para mejorar el modelo, se dejaron las mismas dos capas ocultas pero ahora con 11 neuronas cada una y se obtuvo el siguiente resultado:

```
clasificador_red = MLPClassifier(hidden_layer_sizes=(11, 11), max_iter=1000, random_state=42)
```

```
Classification report:
              precision    recall  f1-score   support

     0           1.00        1.00        1.00         10
     1           1.00        1.00        1.00          9
     2           1.00        1.00        1.00         11

 accuracy          1.00
macro avg          1.00        1.00        1.00         30
weighted avg       1.00        1.00        1.00         30
```

Sesgo (Bias): 0.0200

Varianza: 0.0007

Calculo de bias y varianza en el punto de diferencia entre train y test:

Grado de Bias (Sesgo): 0.0167 - Bajo

Grado de Varianza: -0.0033 - Bajo

El modelo tiene buen equilibrio entre el bias y la varianza por lo que existe fitt

Como podemos observar, el accuracy mejora bastante al igual que la precisión, el recall y el f1-score. Por otra parte, la varianza y el bias también disminuyen por lo que se vuelve un mejor modelo.

Otra técnica usada fue la de regularización

```
clasificador_red_regularizado = MLPClassifier(hidden_layer_sizes=(2, 2), max_iter=7000, alpha=1, random_state=42)
```

En esta parte se agregó alpha a MLPClassifier de scikit-learn, la cual controla la regularización L2 en la red neuronal. Agregar regularización puede ayudar a reducir el overfitting. En este caso se eligió por prueba y error el valor de alpha donde observamos que disminuye

Al hacer lo anterior obtuvimos lo siguiente respecto al bias y la varianza

```
Sesgo (Bias): 0.0267  
Varianza: 0.0011
```

Aquí podemos observar que la varianza disminuye por lo que alpha si ayudó en la parte del overfitting

Otra técnica utilizada fue el early stopping.

```
clasificador_red = MLPClassifier(hidden_layer_sizes=(500, 500),  
                                max_iter=7000,  
                                alpha=0.0001,  
                                solver='adam',  
                                random_state=42,  
                                early_stopping=True,  
                                validation_fraction=0.2,  
                                n_iter_no_change=10) #(10, 10)
```

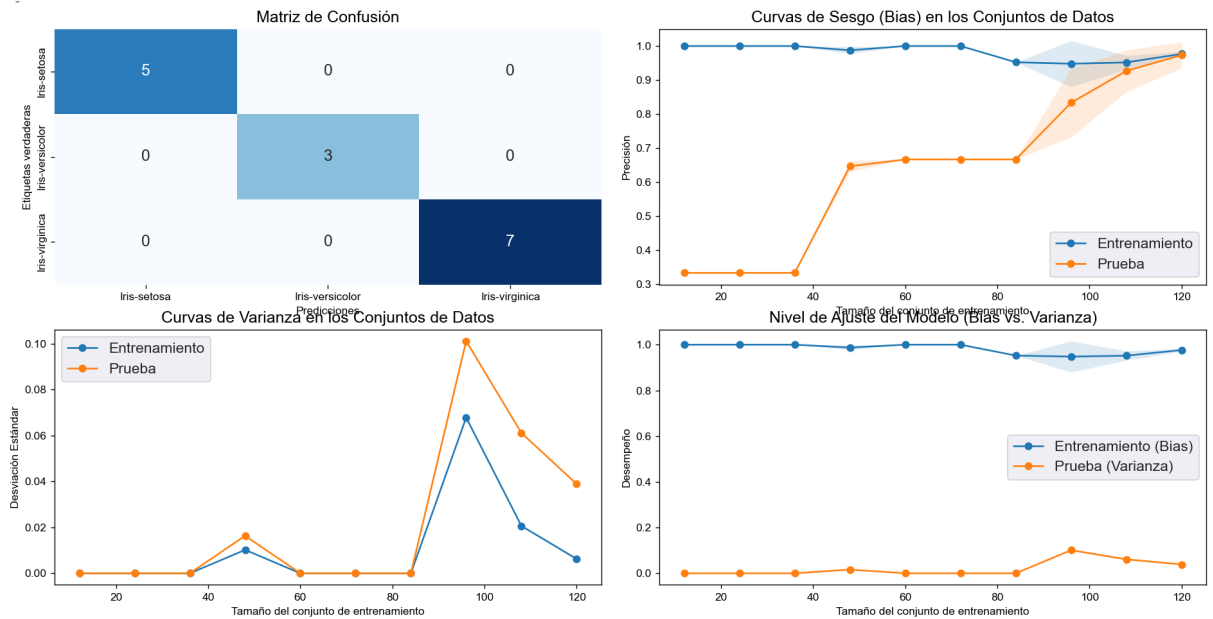
El early stopping es una técnica de regularización utilizada para evitar el sobreajuste al detener el entrenamiento del modelo tan pronto como el rendimiento en un conjunto de datos de validación comience a empeorar.

Además se agregaron otros parámetros con la finalidad de mejorar el modelo como fueron:

validation_fraction: Es una proporción del conjunto de entrenamiento que se separará para ser utilizado como conjunto de validación interno para el early stopping. En este caso fue 0.2, lo que significa que se utilizará 20% de los datos de entrenamiento para validación.

n_iter_no_change: Este parámetro es el número de iteraciones sin mejora en la precisión del conjunto de validación para esperar antes de detener el entrenamiento. En este caso es 10, lo que significa que el entrenamiento se detendrá si la precisión del conjunto de validación no mejora en 10 iteraciones consecutivas.

Los resultados son los siguientes:



Por último, también se puede realizar una normalización de los datos, esto sirve para:

- **Convergencia más Rápida:** Al entrenar una red neuronal usando el método de gradiente descendente o variantes de este, si las características tienen diferentes escalas, el gradiente puede dirigirse fuertemente en direcciones particulares.
- **Convergencia Más Estable:** Al tener características en escalas similares, el patrón de optimización es más regular.
- **Permite Uso Efectivo de Regularización:** La normalización asegura que la regularización se aplica por igual a todas las características, lo que es esencial para que la regularización funcione correctamente.

Resultados

```

Conjunto de Entrenamiento (Train):
[[0.08823529 0.66666667 0.         0.04166667]
 [0.41176471 1.         0.0877193  0.125    ]
 [0.70588235 0.45833333 0.59649123 0.54166667]
 [0.14705882 0.58333333 0.10526316 0.04166667]
 [0.02941176 0.5       0.05263158 0.04166667]]
Shape: (120, 4)

Conjunto de prueba (Test):
[[0.20588235 0.58333333 0.10526316 0.125    ]
 [0.41176471 0.75      0.12280702 0.08333333]
 [1.         0.25      1.03508772 0.91666667]
 [0.70588235 0.41666667 0.73684211 0.91666667]
 [0.5        0.375     0.61403509 0.58333333]]
Shape: (15, 4)

Conjunto de validacion (Validation):
[[0.73529412 0.5       0.85964912 0.91666667]
 [0.58823529 0.54166667 0.64912281 0.625    ]
 [0.61764706 0.33333333 0.80701754 0.83333333]
 [0.38235294 0.20833333 0.50877193 0.41666667]
 [0.55882353 0.08333333 0.61403509 0.58333333]]

```

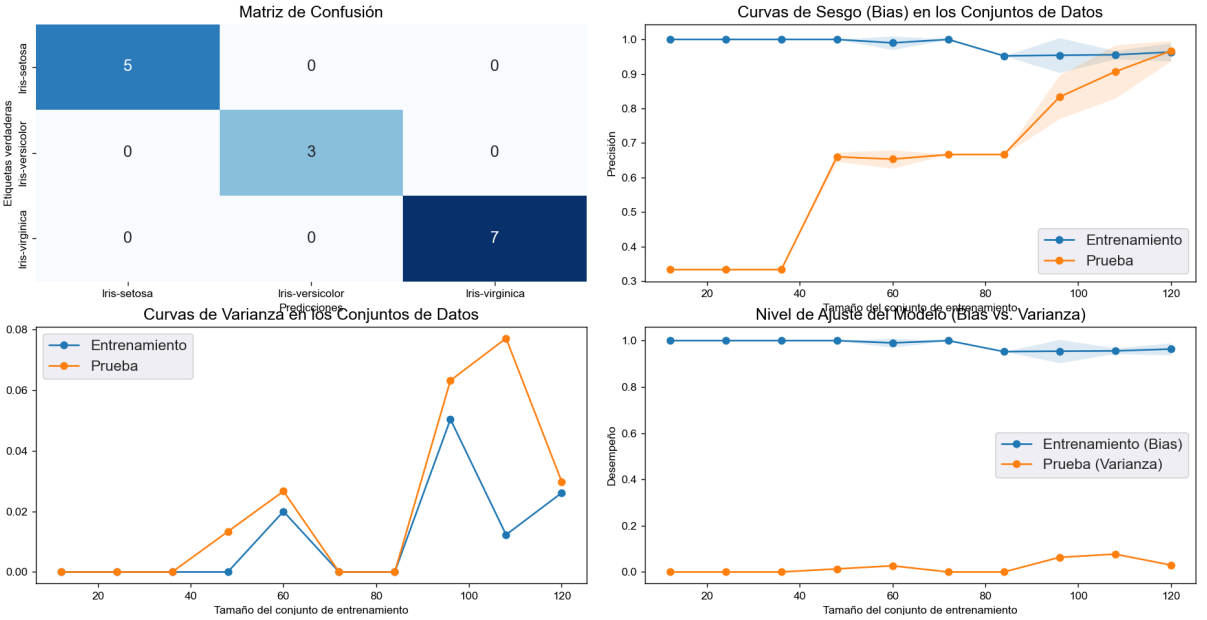


```
Classification report:
      precision    recall  f1-score   support


     0         1.00      1.00      1.00         5
     1         1.00      1.00      1.00         3
     2         1.00      1.00      1.00         7

 accuracy         1.00
macro avg         1.00      1.00      1.00        15
weighted avg         1.00      1.00      1.00        15

Sesgo (Bias): 0.0333
Varianza: 0.0009
Calculo de bias y varianza en el punto de diferencia entre train y test:
Grado de Bias (Sesgo): 0.0367 - Bajo
Grado de Varianza: 0.0033 - Bajo
El modelo tiene buen equilibrio entre el bias y la varianza por lo que existe fitt
```



Referencias:

- Romero, I. (2021, 4 marzo). *La dicotomía sesgo-varianza en modelos de machine learning - Keepler | Cloud Data Driven Partner*. Keepler | Cloud Data Driven Partner.
<https://keepler.io/es/2021/03/la-dicotomia-sesgo-varianza-en-modelos-de-machine-learning/>
- Gonzalez, L. (2022). Sesgo y varianza en machine learning.  *Aprende IA*.
<https://aprendeia.com/bias-y-varianza-en-machine-learning/>