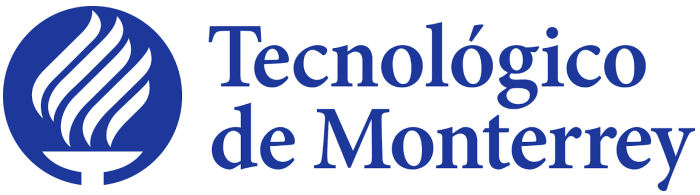

**Análisis y Reporte sobre el desempeño del modelo.**

Diego Zurita Villarreal A01748227

Group 101

Professor: Jorge Uresti

IA avanzada

Campus Estado de México

Septiembre 11, 2024

**Justificación del dataset**

El dataset es apropiado para el modelo de Árbol de decisión. Este consiste en múltiples variables para determinar la enfermedad de un paciente, la combinación de síntomas resulta en el output que se debe generar. En el dataset hay más de 2 posibles enfermedades, por lo que nos encontramos ante un problema de clasificación múltiple. Es por esta razón que se decidió usar este algoritmo para la implementación.

Los datos se separaron en tres conjuntos: entrenamiento, prueba y validación. Primero se hizo una separación de 80/20 para los datos de entrenamiento y prueba respectivamente. Posteriormente se dividieron de nuevo los datos entrenamientos 80/20 para obtener los de validación. De esta manera, la distribución quedó de la siguiente forma:

- Entrenamiento: 64%
- Prueba: 20%
- Validación: 16%

```
# Dividir los datos en conjunto de entrenamiento, validación y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
```

*Imagen 1. Codificación de los conjuntos de datos*

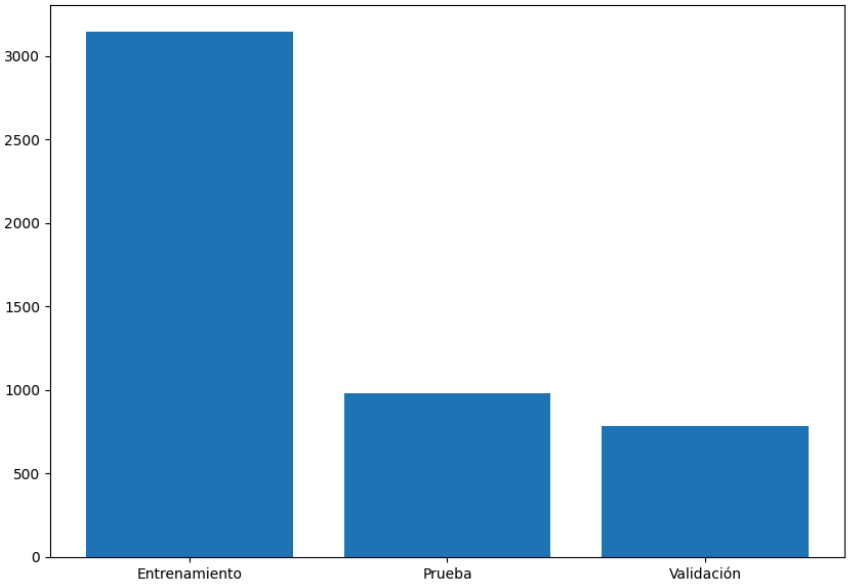


*Imagen 2. Distribución de los conjuntos de datos*

## Gráficas de resultados

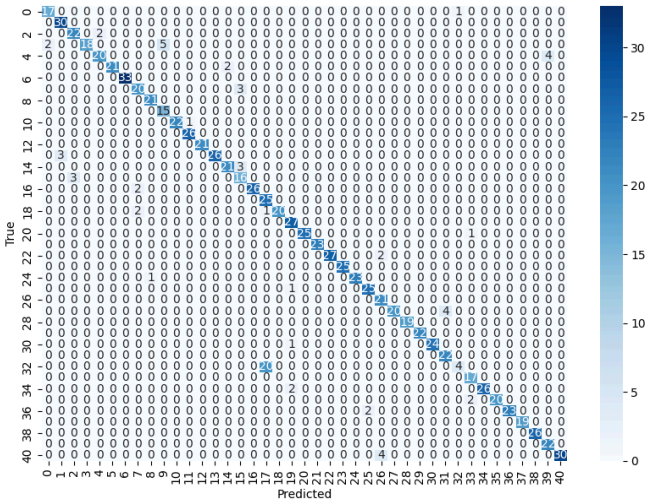


*Imagen 3. Matriz de confusión con datos de prueba*

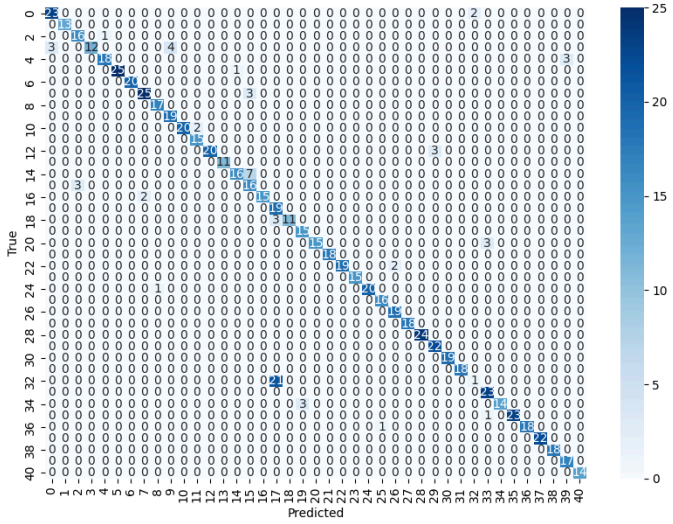


*Imagen 4. Matriz de confusión con datos de validación*

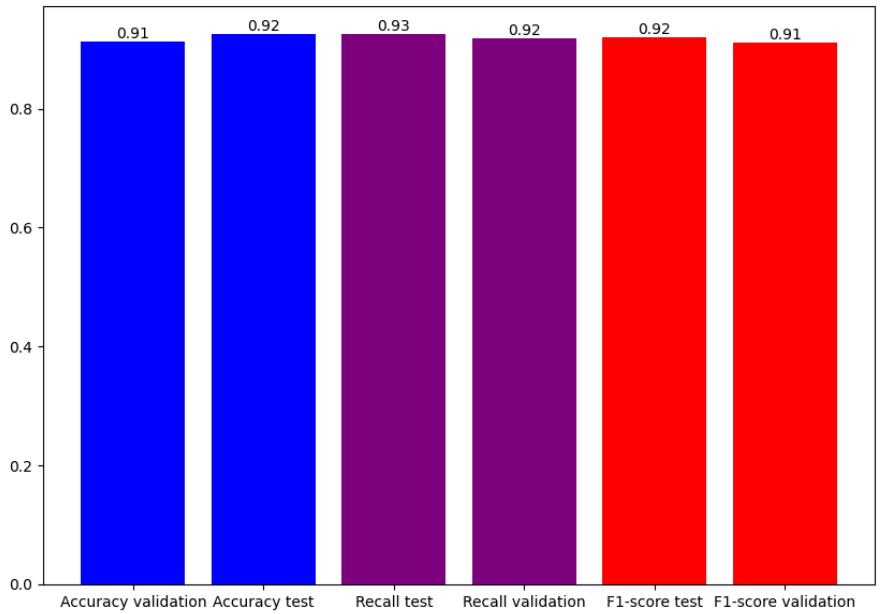


*Imagen 5. Gráfica de barras de las métricas de los datos de prueba y validación*

**BIAS**

Precisión del modelo:  0.91
La precisión y el recall están cerca de 1.00, lo que indica un buen rendimiento por parte del modelo.
Se puede diagnosticar un bias bajo ya que las predicciones son muy precisas, lo que sugiere que el modelo está capturando correctamente las relaciones en los datos.

**Varianza**

Aunque la mayoría de las predicciones son correctas, en algunas clases se puede apreciar un rendimiento bajo, como la 32 con una f1 de 0.08 y un recall de 0.05. Analizando estos números se podría inferir que el modelo presenta un leve grado de overfitting ya que el modelo no funciona bien en todos los casos.

**Nivel de ajuste del modelo**

Lo más probable es que el modelo presente un leve overfitting. Esto se puede decir por el bajo bias y la varianza relativamente alta en algunas de las clases. Como se ha dicho antes, si bien el modelo hace buenas predicciones, este no funciona para todos los casos. Esto quiere decir que el modelo se ha ajustado de más a los datos de entrenamiento por lo cual al ser expuesto a valores totalmente nuevos tiende a fallar.

**Uso de técnicas**

*Análisis de importancia de características:* Usé esta técnica para analizar la importancia de las características en busca de disminuir el espacio de búsqueda del modelo al enfocarse en características más relevantes. Sin embargo, al tratarse de síntomas que en ocasiones está su campo vacío, no veo tan claro cuál característica se considera no eficiente. Cada fila al menos tiene un síntoma, por lo que el valor 1 tiene la calificación más alta en el análisis. Lo que supongo es que al ser más raro que un paciente tenga 10 síntomas, es por eso que sale que las últimas características son las menos relevantes.

*GridSearchCV:* Usé esta técnica para poder encontrar la mejor combinación de hiper parámetros para mi árbol de decisión, ya que al haber tantos de ellos con tantos posibles valores, encontrar la combinación perfecta manualmente hubiera sido muy laborioso. Gracias a esto pude poner 4 hiper parámetros, cuando antes solo había puesto 2. El modelo subió su precisión de 0.91 a 1.00 con la implementación de esta técnica

*Min_samples_split and min_samples_leaf:* Agregué estos dos otros parámetros para reducir la complejidad del árbol y evitar que este se ajuste demasiado a los datos de entrenamiento. Antes solo había usado criterion y max_depth. Agregar estos hiper parámetros fue posible gracias a GridSearchCV, que permite encontrar el mejor valor para cada parámetro.

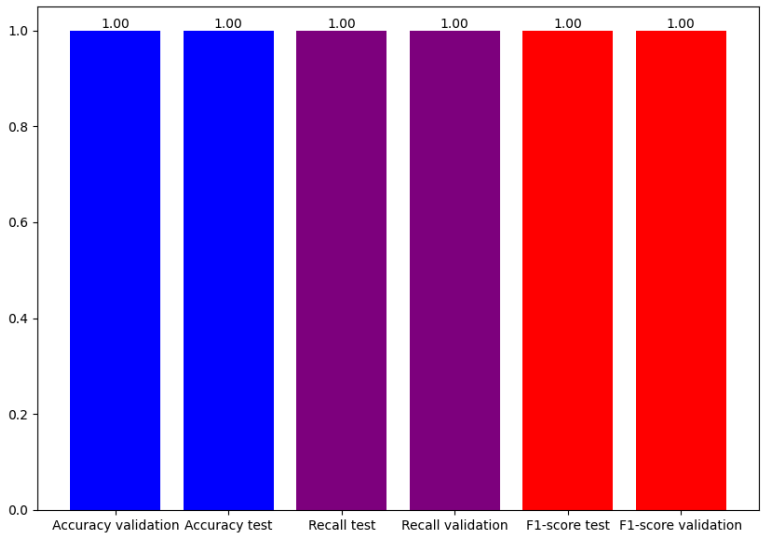


*Imagen 6. Gráfica de barras de las métricas de los datos de prueba y validación después de haber agregado las 3 técnicas anteriores*