Blue text on a black background

Description automatically generated

**Inteligencia Artificial Avanzada para la Ciencia de Datos (Gpo 102)**

**Curso:**

TC3006C.102

**Campus:**

Monterrey

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

**Equipo 2:**

Rafhael Eduardo Chavez Ramirez | A00832228

**Lugar y Fecha:**

Monterrey, Nuevo León

1 de septiembre del 2024

Datos:  
**Columnas y su descripción**:

* Gender: Género de la persona (Male, Female).
* Age: Edad en años.
* Height: Altura en metros.
* Weight: Peso en kilogramos.
* family\_history\_with\_overweight: Historial familiar de sobrepeso (yes/no).
* FAVC: Frecuencia de consumo de alimentos calóricos (yes/no).
* FCVC: Frecuencia de consumo de vegetales (escala de 1 a 3).
* NCP: Número de comidas principales al día.
* CAEC: Frecuencia de consumo de alimentos entre comidas (Nunca, Algunas veces, Frecuentemente, Siempre).
* SMOKE: Si la persona fuma (yes/no).
* CH2O: Consumo diario de agua (en litros, escala de 1 a 3).
* SCC: Monitoreo de calorías consumidas (yes/no).
* FAF: Frecuencia de actividad física por semana (escala de 0 a 3).
* TUE: Tiempo de uso de dispositivos electrónicos por día (escala de 0 a 2).
* CALC: Consumo de alcohol (Nunca, Algunas veces, Frecuentemente, Siempre).
* MTRANS: Medio de transporte utilizado (Automóvil, Motocicleta, Bicicleta, Transporte público, Caminata).
* NObeyesdad: Variable target que clasifica el nivel de obesidad (Insufficient\_Weight, Normal\_Weight, Overweight\_Level\_I, Overweight\_Level\_II, Obesity\_Type\_I, Obesity\_Type\_II, Obesity\_Type\_III)

Modelo:

Estoy utilizando un árbol de decisión como modelo de clasificación para predecir el nivel de obesidad (NObeyesdad) de las personas en un dataset que contiene diversas características relacionadas con hábitos alimenticios, actividad física y datos demográficos.

Un árbol de decisión es un modelo de machine learning que se utiliza tanto para tareas de clasificación como de regresión. Funciona dividiendo los datos en subconjuntos basados en reglas que se generan a partir de las características de los datos. Estas divisiones se realizan en forma de "ramas", donde cada nodo representa una condición sobre una característica, y las hojas representan las predicciones (en este caso, las diferentes clases de obesidad).

Elegí un árbol de decisión porque es un modelo interpretable, fácil de visualizar y que puede manejar tanto variables numéricas como categóricas sin necesidad de una transformación compleja. Además, los árboles de decisión no requieren normalización de los datos, lo que los hace convenientes para este tipo de dataset con diversas variables de diferentes escalas y tipos.

**Cómo lo implementé**

El modelo fue implementado utilizando la biblioteca sklearn en Python. A continuación, un resumen del proceso:

1. **Preparación de los datos**:
   * Las características (features) incluyen variables como edad, peso, altura, historial familiar de sobrepeso, frecuencia de actividad física, entre otras.
   * La variable objetivo (target) es la columna NObeyesdad, que clasifica a las personas en diferentes categorías de peso: desde Insufficient\_Weight hasta Obesity\_Type\_III.
2. **Entrenamiento del modelo**:
   * Dividí el dataset en conjuntos de entrenamiento y prueba. Esto lo logro con el trainsplit. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)
   * Utilicé un árbol de decisión de clasificación ajustado a los datos para encontrar patrones que permitan predecir la clase de obesidad de una persona con base en sus características.
   * Esto lo hice dos veces. Primero con los hiperparametros default de sklearn y otro buscando los mejores hiperparametros usando gridsearch
3. **Evaluación del modelo**:
   * Evalué el modelo utilizando métricas como la exactitud, precision, recall y F1 Score para medir el desempeño del clasificador en el conjunto de prueba y una grafica de accuracy del modelo en entrenamiento y en test

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

Mi modelo esta dividido en valores de entrenamiento y de testo

**Resultado de aprendizaje del modelo con hiperparametros default**

|  |  |
| --- | --- |
| Conjunto | Exactitud |
| Entrenamiento | 0.66 |
| Prueba | 0.63 |

En estos resultados observamos como el conjunto de entrenamiento y de prueba se comportan muy similar lo que levanta sospechas de problemas de la varianza y el sesgo que revisaremos en el diagnostico

**Resultado de aprendizaje del modelo con mejores hiperparametros**

|  |  |
| --- | --- |
| Conjunto | Exactitud |
| Entrenamiento | 0.94 |
| Prueba | 0.74 |

Vemos que hay una mejora significativa en el modelo cuando cambiamos los hiperarametros usando la técnica de grid search. Aquí vemos que existe una gran diferencia entre el entrenamiento y la prueba, esto a simple vista parece tener algunos problemas, pero más adelante veremos que demuestra una mejoría en el modelo.

**Diagnóstico del Grado de Sesgo (Bias) y Varianza:**

**Sesgo (Bias):**

* **Sesgo Alto (Underfitting):** El modelo es demasiado simple para capturar la complejidad de los datos. Esto suele ser evidente si el modelo tiene un rendimiento bajo tanto en el conjunto de entrenamiento como en el de prueba.
* **Sesgo Bajo (Good Fitting):** El modelo captura bien la relación entre las características y la variable objetivo en el conjunto de entrenamiento y de prueba.

**Cómo Evaluar:**

* **Curvas de Aprendizaje:** Grafica la precisión (o error) en función del tamaño del entrenamiento para el conjunto de entrenamiento y el de prueba. Un alto sesgo se muestra como una precisión baja en ambos conjuntos.
* **Métricas de Rendimiento:** Compara el error de entrenamiento y el error de prueba. Si ambos son altos, puede ser un indicio de alto sesgo.

**Curvas de aprendizaje del modelo con hiperparametros default**

A graph with numbers and lines

Description automatically generated

En la gráfica, las precisiones en el conjunto de entrenamiento y de prueba son bajas y las curvas están cercanas entre sí, esto indica un modelo con alto sesgo. No está capturando adecuadamente los patrones de los datos.

**Curvas de aprendizaje con hiperparametros encontrados con gridsearch**

A graph of a graph showing the difference between a line and a line

Description automatically generated with medium confidence

Ya mejorando los hiperparametros vemos la curva de entrenamiento se queda alta, lo que implica un bajo sesgo y como la de prueba está subiendo implicando que si se aumenta el conjunto de entrenamiento tengamos un mejor desempeño.

**Varianza:**

* **Varianza Alta (Overfitting):** El modelo se ajusta demasiado a los datos de entrenamiento y tiene un rendimiento bajo en el conjunto de prueba. Esto indica que el modelo ha aprendido el ruido del entrenamiento en lugar de generalizar.
* **Varianza Baja (Good Fitting):** El modelo generaliza bien, mostrando un buen rendimiento tanto en los datos de entrenamiento como en los de prueba.

**Cómo Evaluar:**

* **Curvas de Aprendizaje:** Grafica el error en función del tamaño del entrenamiento para los conjuntos de entrenamiento y de prueba. Una alta varianza se muestra cuando el error de entrenamiento es bajo pero el error de prueba es alto.
* **Métricas de Rendimiento:** Si el modelo tiene un bajo error de entrenamiento pero un alto error de prueba, indica alta varianza.

**Curvas de aprendizaje del modelo con hiperparametros default**

=A graph with numbers and lines

Description automatically generated

En la gráfica el hecho de que la curva de prueba siga a la curva de entrenamiento y no haya una gran diferencia entre ellas sugiere que el modelo tiene **baja varianza**. Esto significa que el modelo generaliza bien, pero podría estar cometiendo errores sistemáticos debido al alto sesgo del modelo

**Curvas de aprendizaje con hiperparametros encontrados con gridsearch**

A graph of a graph showing the difference between a line and a line

Description automatically generated with medium confidence

Ya mejorando los hiperparametros vemos que el hecho de que las curvas de entrenamiento y de prueba estén **cercanas entre sí** indica que el modelo tiene una **baja varianza**. En otras palabras, el modelo está generalizando bien a nuevos datos, lo que significa que no está sobreajustando los datos de entrenamiento.

**2. Diagnóstico y Explicación del Nivel de Ajuste del Modelo:**

**Underfitting (Modelo Subajustado):**

* El modelo es demasiado simple y no captura la relación subyacente en los datos.
* **Características:** Error alto tanto en el conjunto de entrenamiento como en el de prueba.

**Cómo Evaluar:**

* **Curvas de Aprendizaje:** Error alto en ambos conjuntos de datos.

**Fitting Adecuado (Good Fitting):**

* El modelo captura bien la relación entre las características y la variable objetivo.
* **Características:** Errores relativamente bajos tanto en el conjunto de entrenamiento como en el de prueba.

**Cómo Evaluar:**

* **Curvas de Aprendizaje:** Los errores en ambos conjuntos son bajos y relativamente cercanos.

**Overfitting (Modelo Sobreajustado):**

* El modelo se ajusta demasiado a los datos de entrenamiento y tiene un rendimiento deficiente en los datos de prueba.
* **Características:** Error bajo en el conjunto de entrenamiento y alto en el de prueba.

**Cómo Evaluar:**

* **Curvas de Aprendizaje:** El error de entrenamiento es bajo, pero el error de prueba es alto.
* **Métricas de Rendimiento:** El modelo tiene un rendimiento mucho mejor en el conjunto de entrenamiento que en el de prueba.

**Curvas de aprendizaje del modelo con hiperparametros default**

=A graph with numbers and lines

Description automatically generated

En la grafica pregrid search, vemos como existe underfitting. Esto se debe a que la precisión de entrenamiento va bajando a 0.6 junto con la precisión de la curva de prueba que muestre que también es bajo la precisión. Ni en entrenamiento ni en prueba el modelo es muy bueno.

**Curvas de aprendizaje con hiperparametros encontrados con gridsearch**

A graph of a graph showing the difference between a line and a line

Description automatically generated with medium confidence

Ya mejorando los hiperparametros vemos como el comportamiento del modelo está en un buen ajuste. Las curvas cercanas y un rendimiento alto en ambos conjuntos sugieren que el modelo no está ni sobreajustando ni subajustando los datos.

No se observa un problema de underfitting ni de overfitting. El modelo está aprendiendo correctamente los patrones de los datos de entrenamiento y los aplica bien en los datos de prueba. la curva de entrenamiento se mantiene alta y constante, esto sugiere que el modelo no está perdiendo capacidad para aprender a medida que se entrena con más datos. Está manteniendo un buen nivel de rendimiento sin ajustes excesivos.

Uso de técnicas para mejorar de ajuste de parámetros.

Para optimizar los hiperparámetros, se utilizó GridSearchCV, que permite evaluar exhaustivamente todas las combinaciones posibles de los hiperparámetros definidos en el grid. Se emplearon 5 pliegues en la validación cruzada y se utilizó la exactitud (accuracy) como métrica de evaluación.

**Resultados de la Búsqueda de Hiperparámetros**

El ajuste de hiperparámetros resultó en los siguientes mejores valores:

* **Criterio**: 'gini'
* **Profundidad Máxima**: None (sin límite)
* **Muestras Mínimas para Dividir**: 5
* **Muestras Mínimas en Nodo Hoja**: 1
* **Características Máximas**: None (considera todas las características)

Estos hiperparámetros proporcionaron el mejor rendimiento según la métrica de exactitud durante la validación cruzada.

**Evaluación del Modelo**

Con los mejores hiperparámetros identificados, se entrenó el modelo final y se realizaron predicciones en el conjunto de prueba. La exactitud del modelo con los hiperparámetros óptimos fue de **0.74**. Esto indica una mejora en la capacidad del modelo para generalizar a datos no vistos y proporcionar predicciones precisas.

Métricas antes de la mejora

A screenshot of a computer code

Description automatically generated

Métricas después de la mejora

A screenshot of a computer

Description automatically generated