Reporte: Predicción de Admisiones a un Programa de MBA utilizando Árboles de Decisión

1. Identificación del Problema y Selección del Algoritmo

Problemática Real

En el proceso de admisión a un programa de MBA, la universidad recibe un gran número de solicitudes cada ciclo, lo que genera la necesidad de optimizar el proceso de selección de candidatos. La idea es desarrollar un modelo predictivo que ayude a identificar automáticamente qué estudiantes tienen más probabilidades de ser admitidos en función de su información de perfil. Esto no solo agilizaría el proceso de selección, sino que también proporcionaría a los administradores del programa información valiosa sobre qué factores son más influyentes en las decisiones de admisión.

Selección del Algoritmo

Para abordar esta problemática, se eligió el **algoritmo de Árboles de Decisión** como técnica de clasificación, ya que es ideal para problemas donde se debe tomar una decisión categórica (en este caso, si el alumno será admitido o no). Además, los árboles de decisión proporcionan interpretabilidad, ya que permiten entender qué características fueron decisivas en la predicción.

Descripción del dataset

El dataset contiene información sobre estudiantes que aplicaron a un programa de MBA en la Universidad de Wharton. Las variables disponibles son las siguientes:

1. **application_id:** Un identificador único para cada aplicación.
2. **gender:** Género del solicitante (Male/Female).
3. **international:** Indica si el solicitante es internacional (True/False).
4. **gpa:** Promedio de calificaciones obtenido por el estudiante en sus estudios previos.
5. **major:** Área de estudio principal del solicitante (e.g., Business, STEM, Humanities).
6. **race:** Raza o etnia del solicitante (e.g., Asian, Black, Hispanic).
7. **gmat:** Puntaje obtenido en el examen GMAT.
8. **work_exp**: Años de experiencia laboral del solicitante.
9. **work_industry:** Industria en la que ha trabajado el solicitante (e.g., Financial Services, Technology, Consulting).
10. **admission:** Estado de admisión (Admit/NaN, donde NaN indica que no fue admitido).

Este conjunto de datos proporciona información clave sobre los antecedentes académicos, la experiencia profesional y el estado de admisión de los estudiantes que solicitaron el programa.

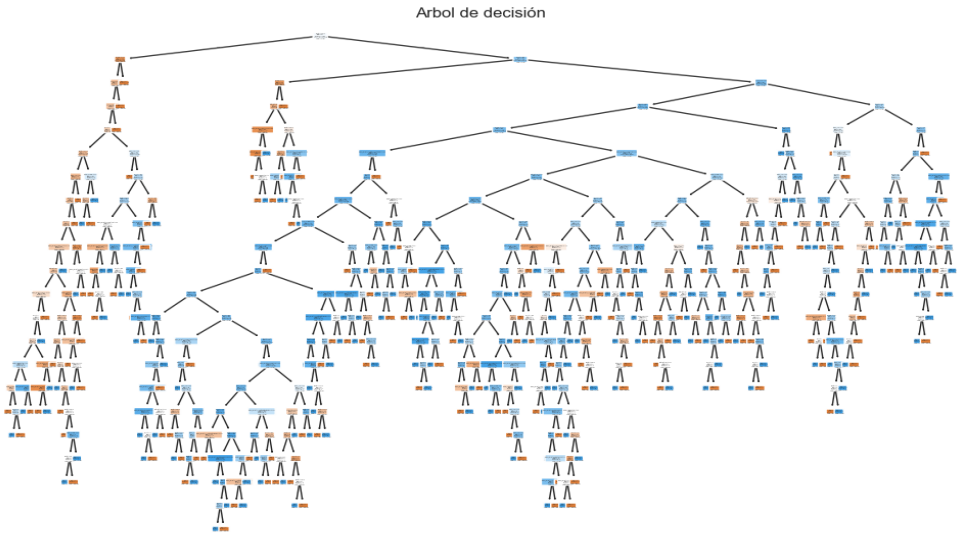
Implementación del Árbol de Decisión

El modelo fue implementado utilizando un dataset de admisiones, en el cual se realizó la limpieza y preparación de los datos. El objetivo principal es predecir la variable binaria **"admission"** (admitido o no), utilizando características del perfil de los estudiantes.

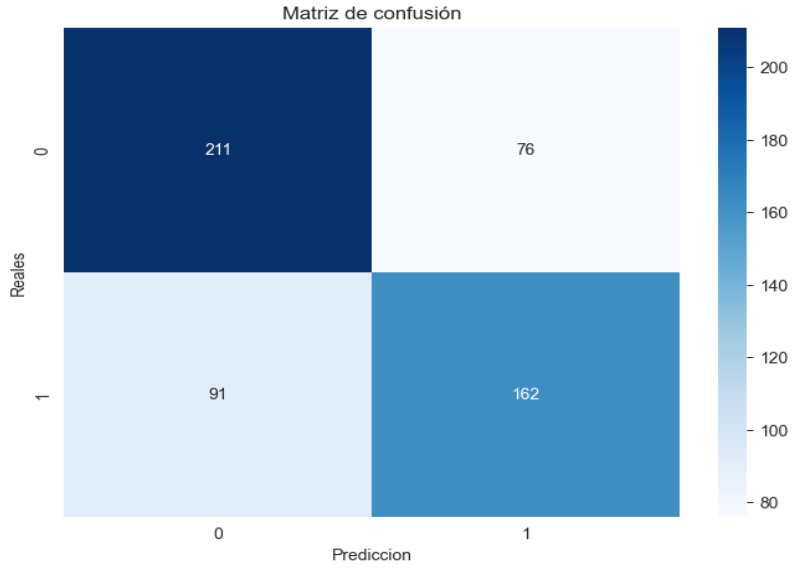
Pasos realizados:

1. **Carga y Preprocesamiento de Datos**:
    - Se cargaron los datos del archivo proporcionado (MBA.csv) y se eliminó la columna de identificación de aplicación, ya que no aportaba valor para la predicción.
    - La columna admission se mapeó a valores binarios, donde "Admit" fue asignado a 1 y los valores nulos fueron mapeados a 0 (para los estudiantes que no fueron admitidos).
2. **Balanceo de Clases**:
    - Al tratarse de un problema desbalanceado (con más rechazos que admisiones), se aplicó un **downsampling** de la clase mayoritaria (rechazos) para equilibrar el conjunto de datos.
3. **Transformación de Variables**:
    - Se realizó un **One-Hot Encoding** para variables categóricas, transformándolas en variables numéricas que el modelo puede utilizar.
4. **División del Conjunto de Datos**:
    - El dataset fue dividido en conjuntos de entrenamiento y prueba, utilizando un 70% para entrenar el modelo y un 30% para validarlo.
5. **Entrenamiento del Árbol de Decisión**:
    - Se entrenó un modelo de **Árbol de Decisión** con los datos de entrenamiento.
    - Se visualizó el árbol utilizando plot_tree, lo que permitió observar cómo se tomaron las decisiones en función de las características de los estudiantes.

En este primer modelo se obtuvo una precisión de 69% obteniendo los siguientes resultados:



Arbol de desición



Matriz de confusión

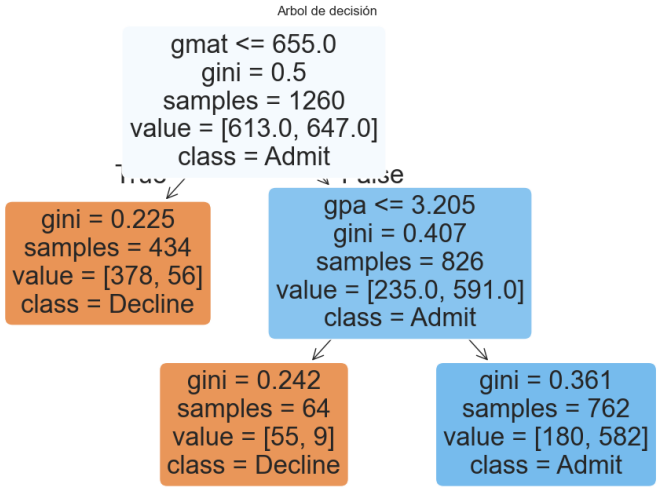
6. **Ajuste del Modelo**

Para mejorar el desempeño del árbol de decisión, se implementó una técnica de **pruning** o poda utilizando el **parámetro ccp_alpha**. Esta técnica permitió reducir el sobreajuste del modelo, mejorando su generalización en el conjunto de prueba.

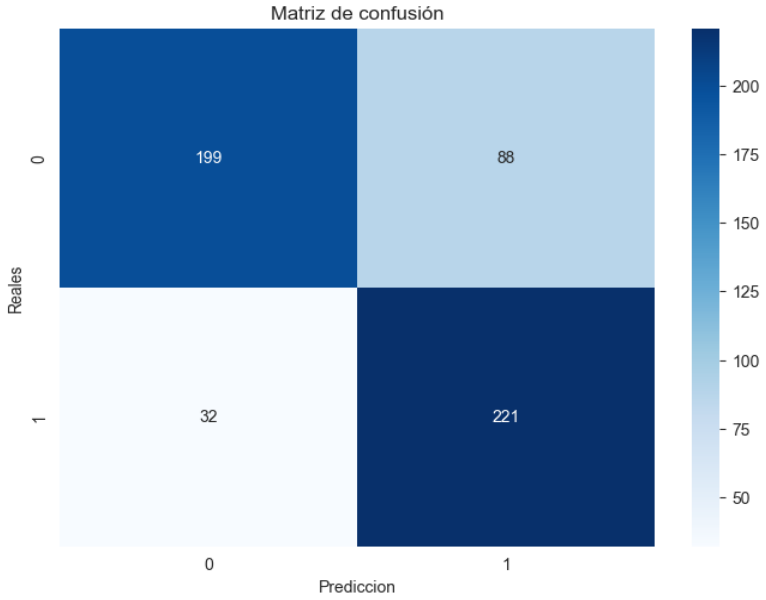
- Se analizaron varios valores de ccp_alpha y se seleccionó el que maximizó la precisión en un proceso de validación cruzada.

- Con este valor de poda, se reentrenó el árbol y se obtuvieron mejores resultados de precisión en el conjunto de prueba.

Después de obtener el mejor valor CCP, se volvió a entrenar el modelo obteniendo los siguientes resultados:



Arbol de decisión mejorado



Matriz de confusión

Paso 4: Evaluación del Modelo

Para evaluar el desempeño del modelo, se utilizaron varias métricas:

1. **Exactitud (Accuracy)**:
    - La precisión obtenida en el conjunto de prueba fue **0.79** aproximadamente, lo que indica que el 75% de las predicciones fueron correctas.
2. **Reporte de Clasificación**:
    - El reporte incluyó métricas como precisión, recall y F1-score, que proporcionaron un panorama más detallado del desempeño del modelo en las clases de admitido y no admitido.
3. **AUC-ROC**:
    - El valor del **AUC-ROC** fue de **0.79**, lo que indica que el modelo tiene un buen poder discriminativo entre los admitidos y no admitidos.
4. **Matriz de Confusión**:
    - Se presentó una matriz de confusión que muestra la cantidad de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Esta herramienta es útil para entender dónde el modelo está cometiendo errores y qué tipo de predicciones son más problemáticas.
5. **Curva ROC**:
    - La **curva ROC** mostró visualmente el trade-off entre la tasa de verdaderos positivos y la tasa de falsos positivos, con un área bajo la curva de **0.79**, lo cual es un buen indicador de desempeño.

Conclusión

En este proyecto, se utilizó un **Árbol de Decisión** para predecir si los estudiantes serían admitidos a un programa de MBA. El modelo mostró un buen desempeño, con una precisión del 75% y un AUC-ROC de 0.83. El ajuste mediante poda mejoró la generalización del modelo, y la visualización del árbol permitió entender las características más influyentes en la decisión de admisión. Este modelo puede ser utilizado como una herramienta de soporte para el departamento de admisiones, ayudando a priorizar solicitudes y tomar decisiones más informadas y eficientes.

Chatbot en DialogFLow

El chatbot desarrollado está diseñado para responder preguntas relacionadas con el desarrollo de este proyecto. Puede proporcionar detalles sobre los siguientes aspectos:

- Proceso de limpieza de datos realizado
- Información sobre el modelo utilizado
- Pruebas efectuadas
- Evaluaciones de rendimiento
- Otros aspectos relevantes del proyecto

Este chatbot ha sido creado en la plataforma **Dialogflow**, donde se han entrenado tanto las posibles preguntas que el usuario podría realizar, como las respuestas correspondientes.

A continuación, se enumeran las principales preguntas que el chatbot es capaz de responder:

- Saludo inicial
- ¿Cuál es el problema que estás resolviendo con tu proyecto?
- ¿Qué algoritmo utilizaste y por qué lo seleccionaste?
- ¿Qué tipo de datos se utilizaron en el modelo?
- ¿Cómo maneja el modelo las variables categóricas?
- ¿Cómo implementaste el modelo en este proyecto?
- ¿Cómo fue evaluado el desempeño del modelo?
- ¿Cuál fue la precisión del modelo?
- ¿Cómo se visualizaron los resultados del modelo?
- ¿Cuáles son los beneficios de utilizar este modelo para predecir admisiones?
- ¿Cuáles son las limitaciones del modelo?
- ¿Es posible ajustar el modelo para mejorar la precisión?
- ¿Cómo puede mejorar el modelo en el futuro?
- ¿Qué herramientas usaste en este proyecto?
- Despedirse

Video de funcionamiento:

https://www.youtube.com/watch?v=qao3Iq6c2zo

Canción con Large Language Models (LLMs)

Esta canción fue generada con la ayuda de LLM, empleando la plataforma CHAT GPT.

Lo primero fue pedir al LLM generar un prompt compatible con la plataforma SUNO, la solicitud fue la siguiente:

“genera un prompt para generar una canción en plataforma **SUNO**, esta debe hablar sobre temas de machine learning, más específicamente sobre arboles de decisión, que hable de sus posibilidades y sus limitantes, y las formas de poder mejorar estos modelos, este prompt debe contener 200 caracteres máximo.”

El prompt obtenido fue el siguiente:

"Los árboles de decisión dividen con precisión, pero a veces su simplicidad trae confusión. Podar ramas, ajustar, o usar bagging y boosting, mejoras clave para que aprendan con mayor solución. Clasifican, pero aún hay limitación."

Este prompt se utilizo directo en la plataforma **SUNO** para generar la letra y melodía de la canción, la letra es la siguiente:

https://suno.com/song/3150f0a4-35aa-422d-9979-68360d59dbf5

[Verso]

Dividen bien los árboles de decisión

Pero su simplicidad trae confusión

Podar ramas cortar ajustar

Siempre buscando mejorar

[Verso]

A veces no hay ruta clara

En el bosque de datos se ampara

Usa bagging dicen boosting también

Mejoras clave se ven en el vaivén

[Estribillo]

Aprende algoritmo sin dudar

Senderos claros hallarás

En la complejidad encontrarás

La solución que buscas más

[Verso]

Su precisión puede ser cruel

En ramas errantes puede caer

Corta ajusta con fuerza y fe

No dejes que el error te destruya

[Verso]

Ellos dicen que estos métodos son reyes

En tus datos profundos clavados como leyes

Bagging promete estabilidad robusta

Boosting asegura que nada se frustra

[Estribillo]

Aprende algoritmo sin dudar

Senderos claros hallarás

En la complejidad encontrarás

La solución que buscas más