**Proyecto Integrador**

**Avance 6**

**Conclusiones Claves**

Dra. Grettel Barcelo Alonso

**EQUIPO 21**

|  |  |
| --- | --- |
|  |  |
| **Bernardo Mijango Flores**  **Dalia Isabel López Tapia**  **David Valles Canedo** | **A01793654**  **A01245026**  **A01191310** |
|  |  |

**9 de junio de 2024**

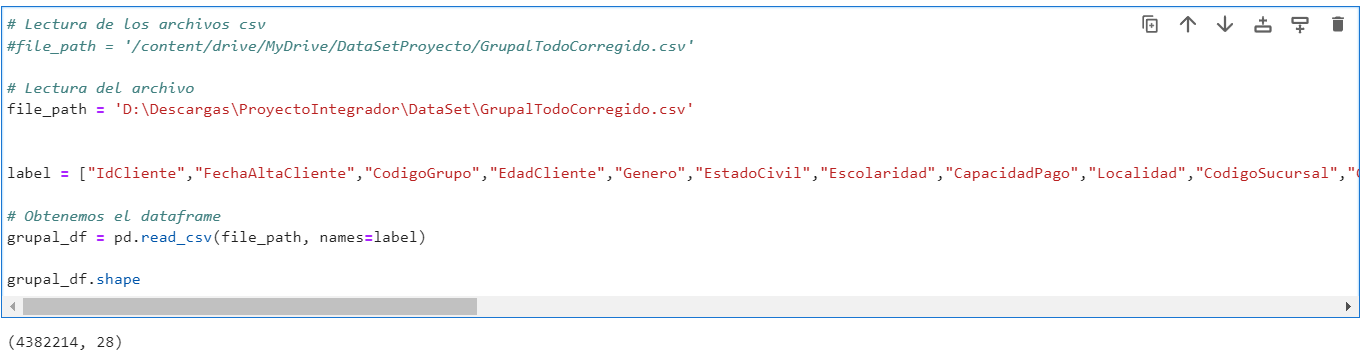
**Objetivo General:**

El objetivo principal de este estudio es desarrollar y evaluar modelos de aprendizaje automático que permitan predecir el estado del crédito de los clientes del producto “Grupal” de la empresa Alsol Contigo utilizando un conjunto de datos que contiene información demográfica y financiera de los clientes, como la edad, estado civil, número de créditos, y diversas métricas crediticias, se busca construir modelos robustos que puedan diferenciar de manera efectiva entre créditos buenos y malos.

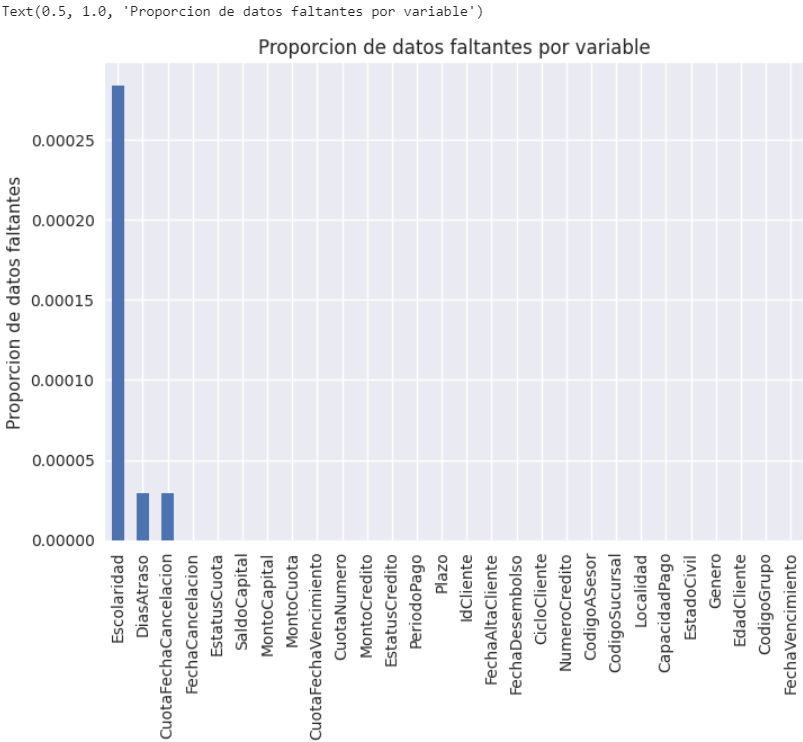
Se busca identificar los modelos con el mejor rendimiento en términos de precisión y eficiencia computacional, con el fin de ofrecer una herramienta confiable y eficaz para la evaluación del riesgo crediticio, utilizando los resultados en la gestión de cobranza temprana con la implementación de un Chat Bot inteligente. El análisis incluye la selección de los modelos más adecuados mediante la comparación de sus puntajes y tiempos de entrenamiento, así como el ajuste destallado de los hiperparámetros para optimizar su desempeño. Además, se evalúa la capacidad predictiva de los modelos seleccionados mediante métricas de evaluación de clasificación y se proporciona una interpretación detallada de los resultados.

**Análisis de los datos**

En la exploración inicial de los datos se comenzó con la carga de información que contiene datos sobre clientes, créditos, características socioeconómicas, historiales crediticios, entre otros. la carga de datos se realiza utilizando bibliotecas como Pandas en Python, que permite la manipulación eficiente de datos tabulares. Se realiza una exploración de estadística descriptiva obteniendo un análisis completo de la información.



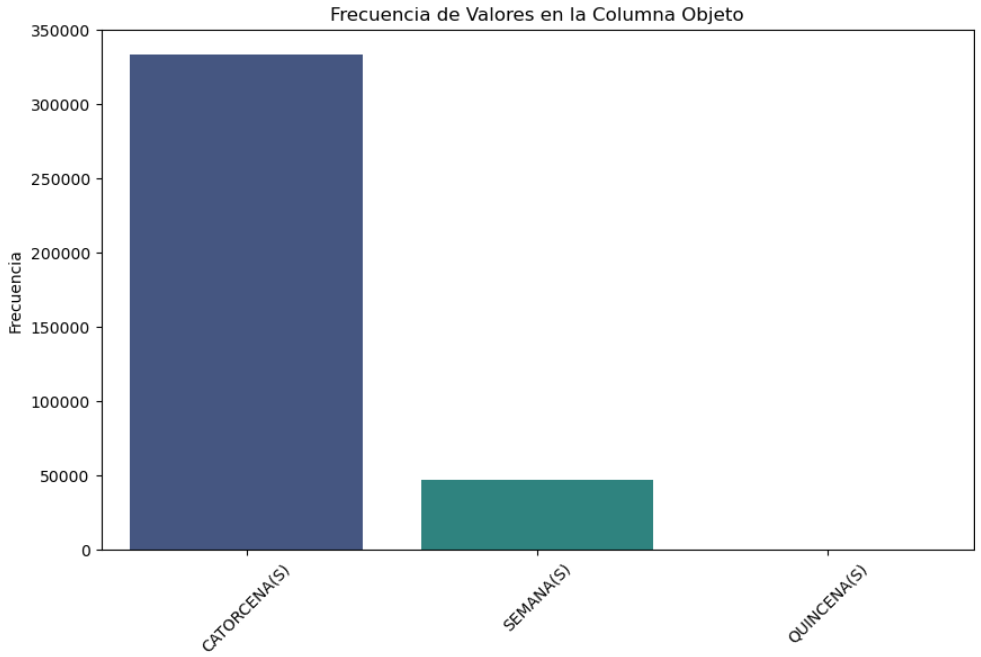
Se determina quitar los valores nulos de la base de datos ya que representa un porcentaje del 0.028% del total de la base de datos.



Al tener el detalle del comportamiento de cada crédito por cliente se determina realizar un análisis de los datos por crédito agregando una nueva variable para obtener el Máximo días de atraso, la Media de los días de atraso y la Mediana de los días de atraso por cada crédito, es decir, se genera una nueva base de datos por crédito agregando las variables “DAMax”, “DAMean” y “DAMedian”



El producto Grupal cuenta con tres periodos de pagos de pagos “CATORCENA(S)”, “SEMANA(S)” y “QUINCENA(S)”, procedemos a obtener la frecuencia de la variable “PeriodoPago”, consideramos que el comportamiento de los créditos puede ser diferentes entre ellos, por lo que se decide trabajar con los créditos de mayor frecuencia resultando ser los de pagos “CATORCENA(S)”



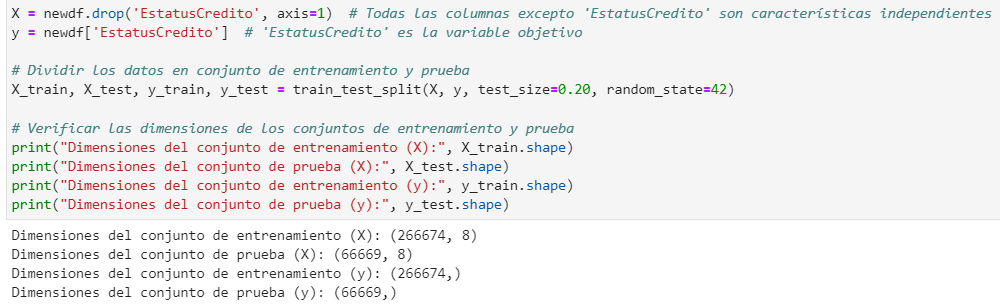
**Variables**

* IdCliente: Número de identificación del cliente
* NumeroCredito: Numero de crédito
* EdadCliente: Edad del cliente al momento del otorgamiento del crédito
* CicloCliente: Numero de créditos contratados por el cliente
* Plazo: Numero de pagos del crédito en periodo CATORCENA(S)
* EstadoCivil: Código del estado civil (1=SOLTERO(A), 2=CASADO(A), 3=VIUDO(A), 4=DIVORCIADO(A), 5=UNION LIBRE, 6=SEPARADO, 7=SE IGNORA)
* DAMax: Dias máximo de atraso
* DAMean: Media de los dias de atraso
* EstatusCredito: Estatus del credito (CAN=PAGADO, CAI=NO PAGADO)

Esta ultima variable se convierte en variable objetivo de valor numérico donde CAN=1 y CAI=0

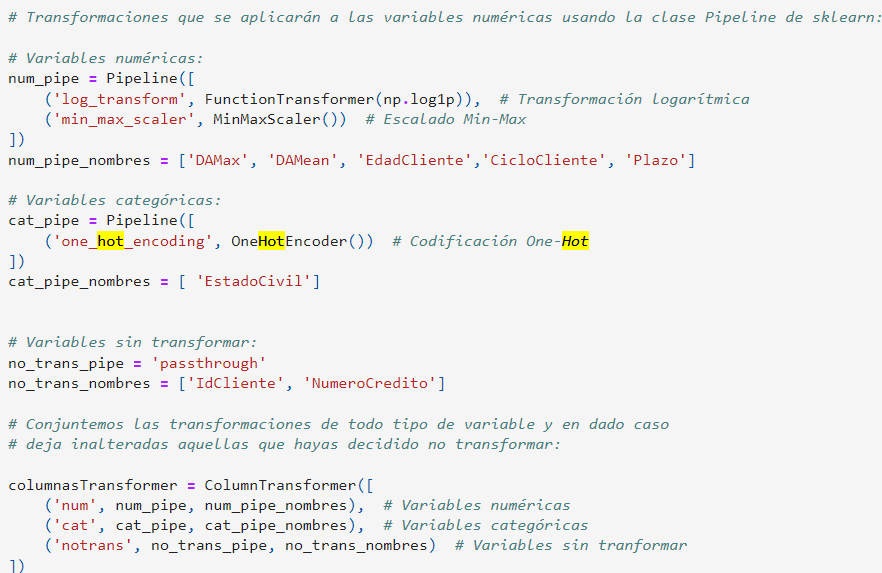


Realizamos la partición del conjunto de datos en entrenamiento del 80% y pruebas del 20%



**Transformaciones y uso del one-hot encoding**

Se aplica una transformación logarítmica y un escalado Min-Max a las columnas númericas, se aplica codifiación One-Hot a las columnas categóricas, se dejan algunas columnas sin transformación y por último se combina todas las transformaciones en un solo transformador.

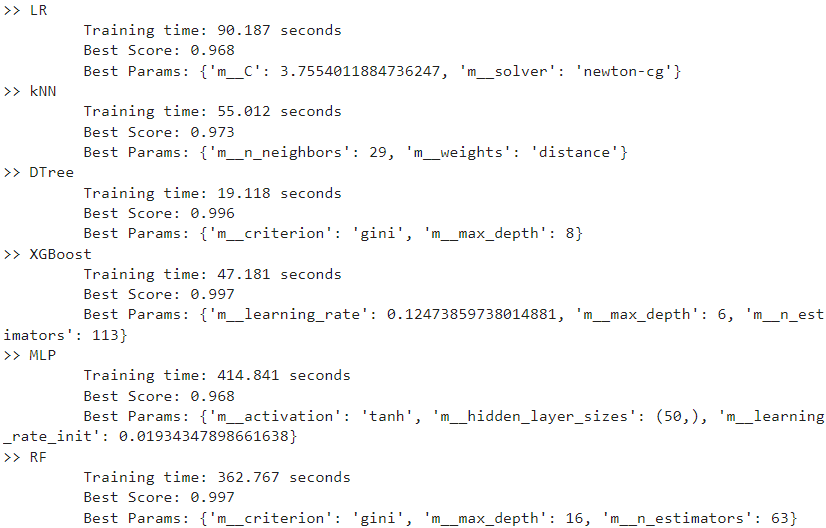
****

Realizamos pruebas con los siguientes modelos:

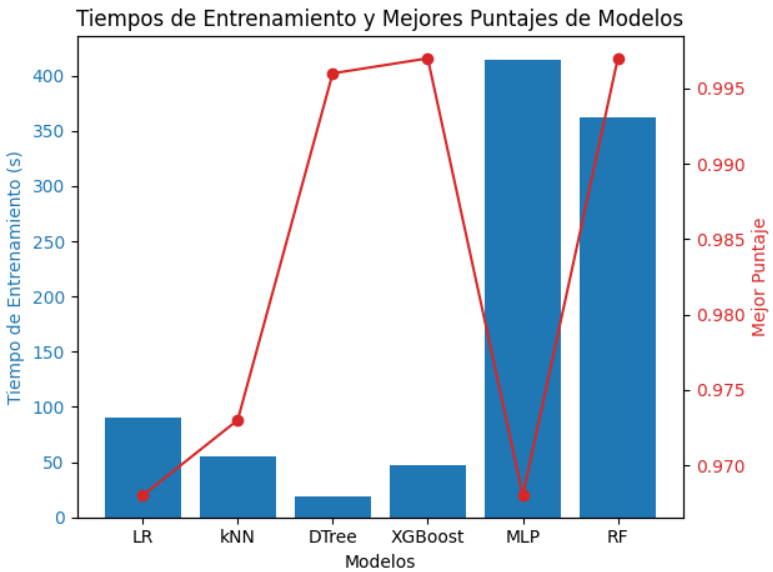
* Regresión Logística: utiliza una función logística para modelar la relación entre una o más variables independientes y una variable dependiente categórica. La salida es una probabilidad que se puede convertir en una clase usando un umbral (generalmente 0.5).
* k-Vecinos más Cercanos (kNN): es un método de clasificación basado en la distancia. Para clasificar una nueva instancia, el algoritmo encuentra los k instancias más cercanas en el espacio de características y asigna la clase más común entre esos vecinos. No requiere un entrenamiento explícito, ya que es un algoritmo basado en instancias.
* Árbol de Decisiones (DTree): es un modelo de predicción que divide repetidamente el espacio de características en subespacios más pequeños utilizando criterios como Gini o Entropía. Cada nodo interno representa una "prueba" en una característica, cada rama representa el resultado de la prueba, y cada nodo hoja representa una clase o valor de predicción.
* XGBoost (Extreme Gradient Boosting): es un método de ensamble que construye un modelo fuerte a partir de una colección de modelos débiles (normalmente árboles de decisión) entrenados secuencialmente. Cada nuevo modelo intenta corregir los errores de los modelos anteriores, mejorando así el rendimiento.
* Perceptrón Multicapa (MLP): es una clase de redes neuronales artificiales que consiste en al menos tres capas de nodos: una capa de entrada, una o más capas ocultas, y una capa de salida. Cada nodo (neurona) en una capa está conectado a cada nodo en la siguiente capa con un peso asociado.
* Random Forest: es un método de ensamble que utiliza múltiples árboles de decisión para realizar la clasificación o regresión. Cada árbol en el bosque es entrenado con una muestra aleatoria del conjunto de datos y cada división en el árbol es seleccionada a partir de un subconjunto aleatorio de las características.

Entre los modelos probados, la Regresión Logística (LR) mostró un ROC AUC muy bajo (0.330), indicando un bajo rendimiento en la distinción de clases, a pesar de las métricas de precisión y recall altas, probablemente debido al desequilibrio de clases. k-Vecinos más Cercanos (kNN) presentó métricas perfectas en el conjunto de entrenamiento y altas en el de prueba (ROC AUC de 0.865), sugiriendo sobreajuste con buena generalización. Árbol de Decisiones (DTree) y XGBoost mostraron métricas consistentemente altas con ligero sobreajuste y tiempos de entrenamiento razonables. Perceptrón Multicapa (MLP) tuvo un ROC AUC de 0.500, indicando una incapacidad para discriminar entre clases, con altos tiempos de entrenamiento y métricas similares a la regresión logística. Por último, Random Forest destacó con métricas excelentes y un ROC AUC de 0.989, sugiriendo un modelo bien ajustado con excelente capacidad de discriminación y buena generalización.

Definimos los espacios de búsqueda de hiperparámetros para cada uno de los modelos. Ajustando a cada modelo utilizando RandomizedSearchCV dentro del bucle de entrenamiento, Realizamos algunos cambios en las transformaciones de las variables numéricas y categóricas mejoramos los modelos con hiperparámetros y obtuvimos los siguientes resultados.



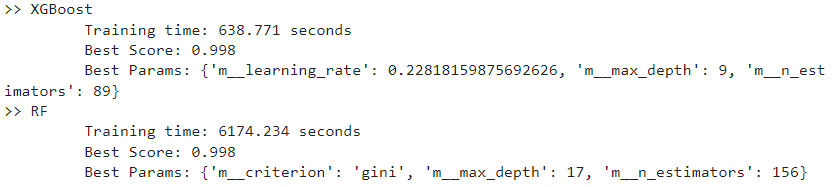
En el análisis, la regresión logística logró un puntaje alto de 0.968 con un tiempo de entrenamiento corto, indicando eficiencia y precisión. El modelo kNN obtuvo un excelente puntaje de 0.973 y tiempos de entrenamiento breves, beneficiándose de un alto número de vecinos y ponderación por distancia. El modelo árbol de decisión logró un puntaje casi perfecto de 0.996 con tiempos de entrenamiento muy bajos, manteniendo una complejidad moderada. XGBoost alcanzó el puntaje más alto de 0.997 con tiempos razonables de entrenamiento, optimizando parámetros para evitar el sobreajuste. El modelo MLP logró un puntaje de 0.968, pero con un tiempo de entrenamiento largo, señalando un alto costo computacional. Bosque aleatorio también obtuvo un puntaje excelente de 0.997, con tiempos de entrenamiento considerables y un buen equilibrio entre precisión y generalización. Estos resultados resaltan la efectividad de XGBoost y Random Forest en precisión, y la eficiencia del Árbol de Decisión en tiempo de entrenamiento, mientras que el MLP es más costoso computacionalmente.



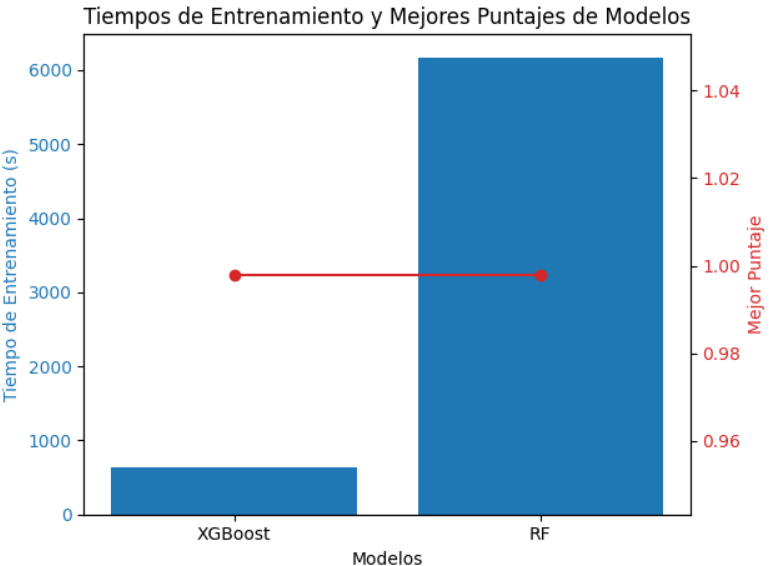
Se realiza un ajuste sobre los dos mejores modelos obtenidos “XGBoost” y “Random Forest” ya que ambos modelos tienen un “Best Score” de 0.997



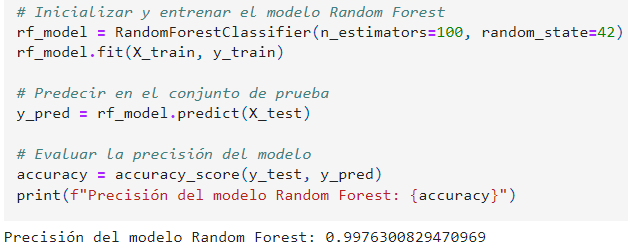
Se obtienen los siguientes resultados.



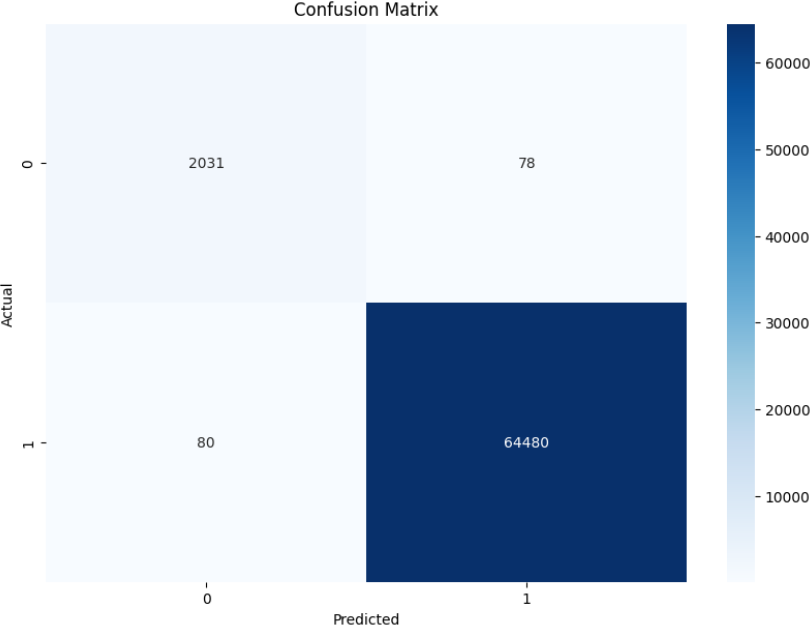
Los modelos XGBoost y Random Forest muestran un rendimiento excepcionalmente alto con un "Best Score" de 0.998. XGBoost tiene un tiempo de entrenamiento razonable de aproximadamente 639 segundos y un rendimiento casi perfecto, con parámetros que equilibran complejidad y evitar el sobreajuste. En contraste, Random Forest, aunque alcanza el mismo rendimiento, requiere un tiempo de entrenamiento significativamente mayor de aproximadamente 6174 segundos, debido a su mayor profundidad de árboles y número de estimadores. Ambos modelos son altamente efectivos, pero XGBoost es preferible cuando el tiempo de entrenamiento es crítico debido a su mayor eficiencia computacional. Random Forest, sin embargo, sigue siendo una opción robusta si los recursos de tiempo y computación no son limitaciones.



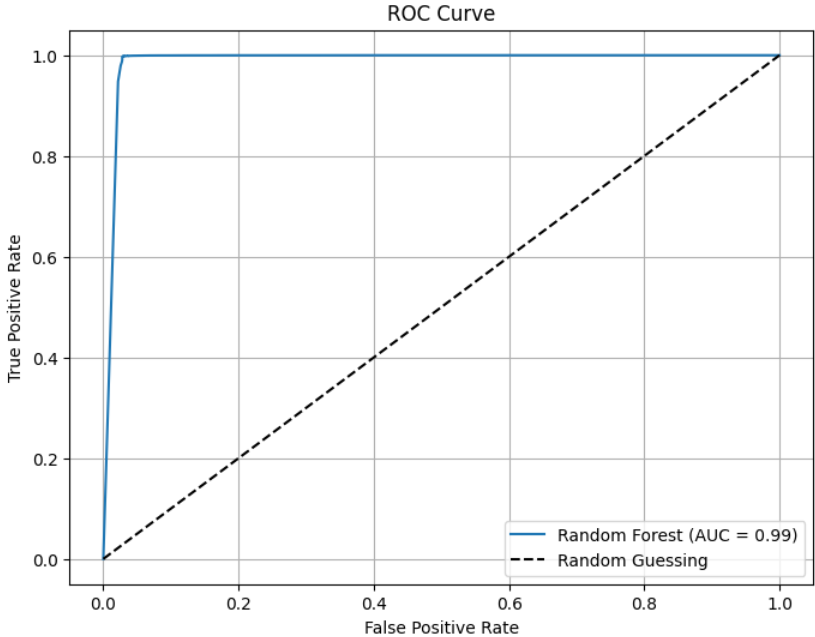
Dado que ambos modelos ofrecen la misma precisión, pero XGBoost requiere un tiempo de entrenamiento mucho menor, se recomienda optar por XGBoost por su eficiencia en términos de tiempo y recursos. Sin embargo, dado el tamaño y la naturaleza del conjunto de datos, técnicas de ensamble como Random Forest o Gradient Boosting son recomendables para empezar, debido a su robustez y capacidad para manejar diversos tipos de datos sin sobreajuste. Random Forest es especialmente adecuado para conjuntos de datos con variables numéricas y categóricas, ofreciendo una opción sólida y robusta para el análisis.



La precisión del modelo Random Forest de 0.9976300829470969 es un resultado extremadamente alto. Sin embargo, es importante interpretar esta métrica con cuidado y considerar otros factores antes de concluir que el modelo es excelente.



Los resultados de la matriz de confusión indican que el modelo Random Forest está funcionando de manera excepcionalmente buena.



La curva ROC del modelo, con un AUC de 0.988, sugiere un desempeño excelente. Esto indica que el modelo es muy eficaz en la discriminación entre clases positivas y negativas, con una alta sensibilidad y especificidad a lo largo de diferentes umbrales de decisión. Este resultado refuerza la confianza del modelo y su aplicabilidad en contextos reales, ya que ofrece una combinación óptima de precisión y robustez.