**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**

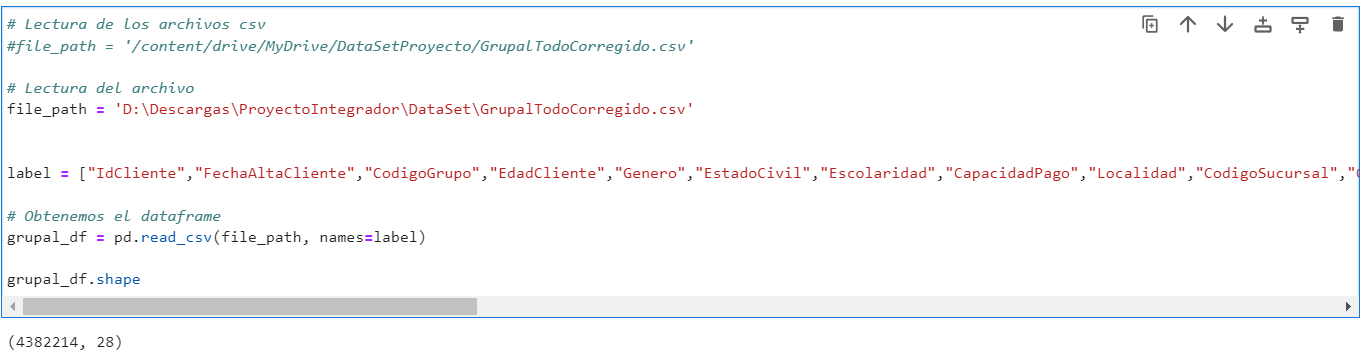
**Objetivo General:**

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.

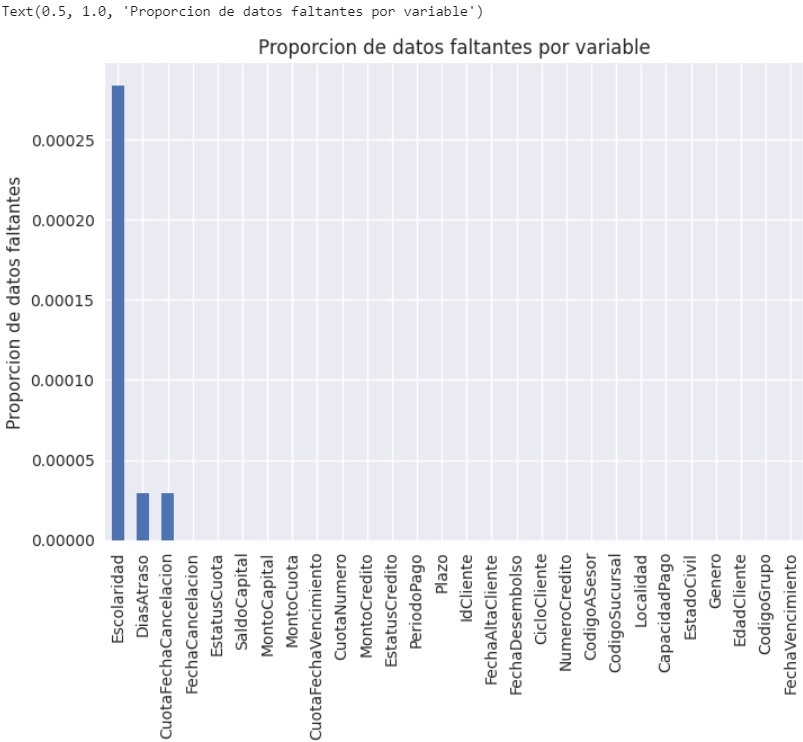
Se busca identificar los modelos con el mejor rendimiento en términos de precisión y eficiencia computacional, con el fin de ofrecer una herramienta confiable y eficaz para la evaluación del riesgo crediticio, utilizando los resultados en la gestión de cobranza temprana con la implementación de un Chat 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 detallado 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.



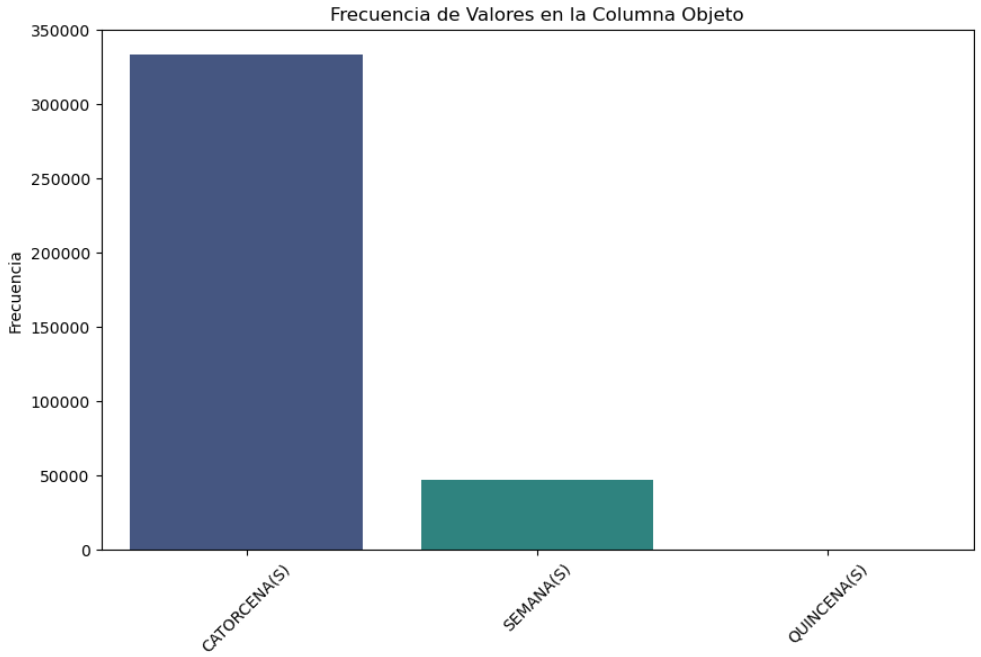
Se determina quitar los valores nulos de la base de datos ya que representa un porcentaje del 0.028% del total de la base de datos.



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)”



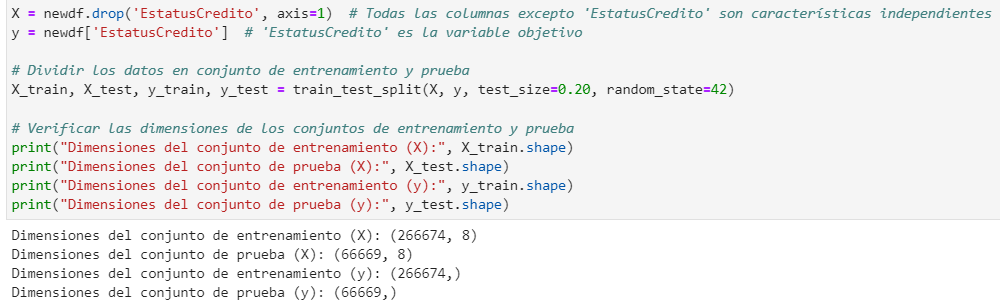
**Variables**

* IdCliente: Número de identificación del cliente
* NumeroCredito: Número de crédito
* EdadCliente: Edad del cliente al momento del otorgamiento del crédito
* CicloCliente: Número de créditos contratados por el cliente
* Plazo: Número 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: Días máximo de atraso
* DAMean: Media de los días de atraso
* EstatusCredito: Estatus del crédito (CAN=PAGADO, CAI=NO PAGADO)

Esta última 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 numéricas, se aplica codificación One-Hot a las columnas categóricas, se dejan algunas columnas sin transformación y por último se combinan todas las transformaciones en un solo transformador.

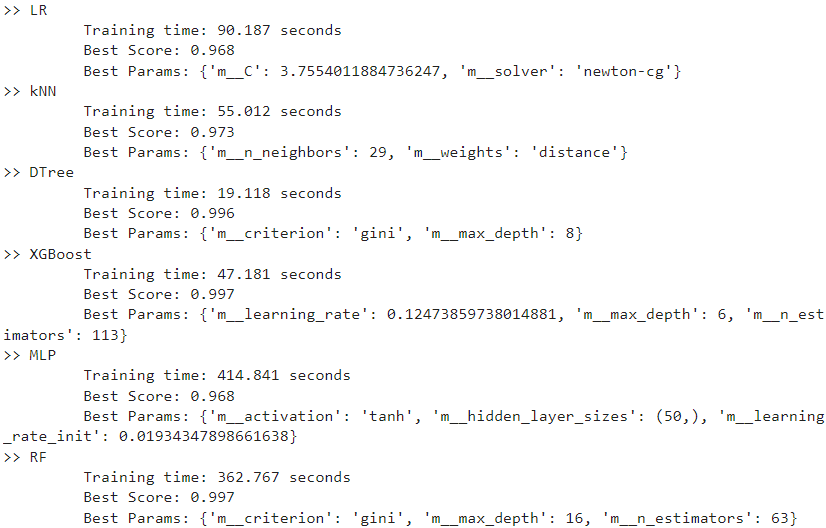
****

Realizamos pruebas con los siguientes modelos:

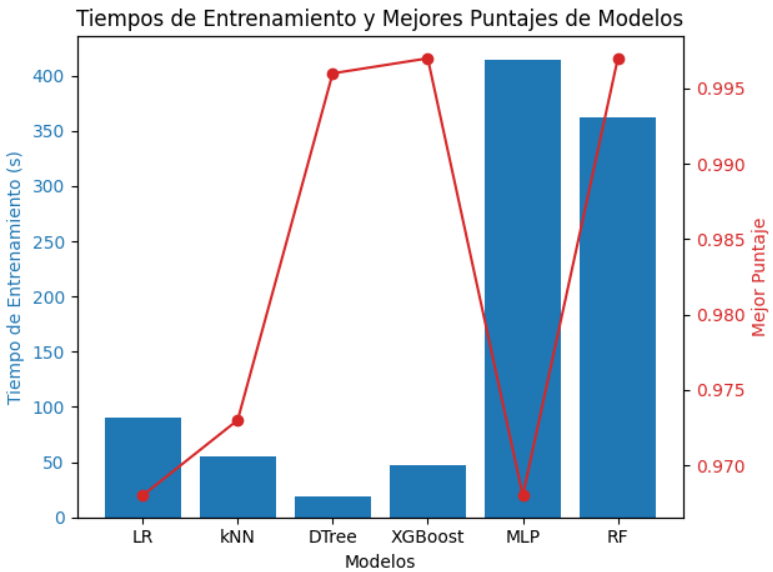
* Regresión Logística: utiliza una función logística para modelar la relación entre una o más variables independientes y una variable dependiente categórica. La salida es una probabilidad que se puede convertir en una clase usando un umbral (generalmente 0.5).
* k-Vecinos más Cercanos (kNN): es un método 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.



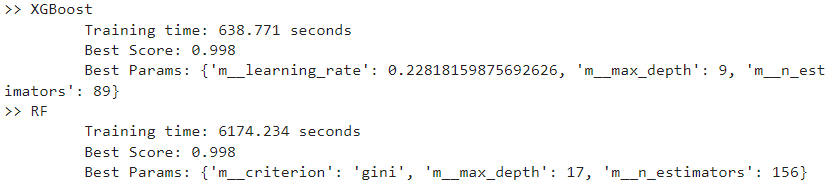
En el análisis, la regresión logística logró un puntaje alto de 0.968 con un tiempo de entrenamiento corto, indicando eficiencia y precisión. El modelo kNN obtuvo un excelente puntaje de 0.973 y tiempos de entrenamiento breves, beneficiándose de un alto número de vecinos y ponderación por distancia. 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.



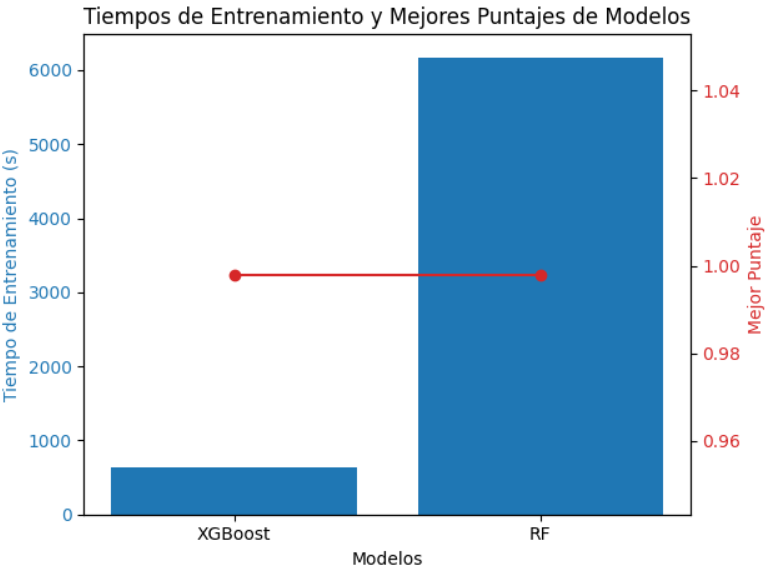
Se realiza un ajuste sobre los dos mejores modelos obtenidos “XGBoost” y “Random Forest” ya que ambos modelos tienen un “Best Score” de 0.997



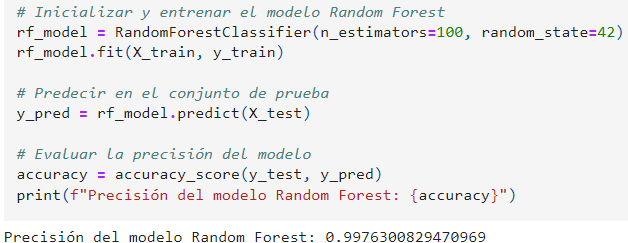
Se obtienen los siguientes resultados.



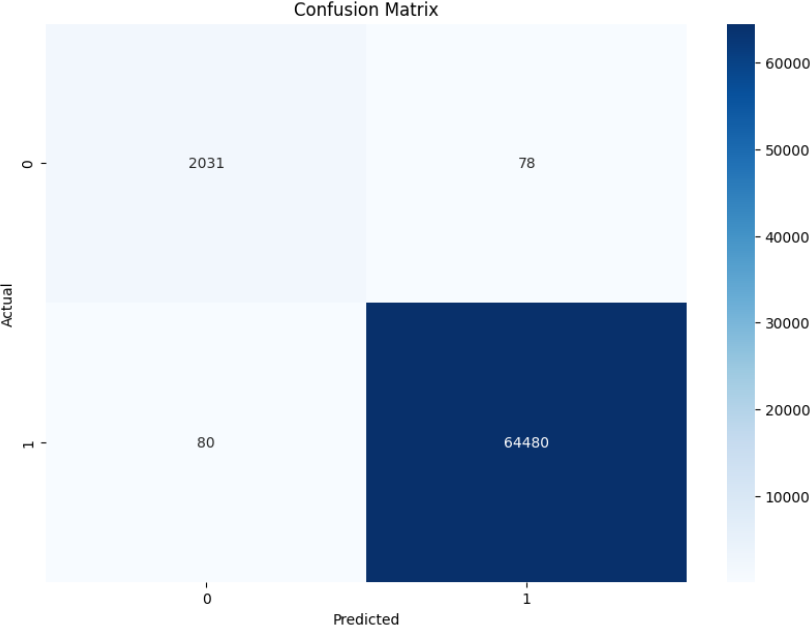
Los modelos XGBoost y Random Forest muestran un rendimiento excepcionalmente alto con un "Best Score" de 0.998. XGBoost tiene un tiempo de entrenamiento razonable de aproximadamente 639 segundos y un rendimiento casi perfecto, con parámetros que equilibran complejidad y evitar el sobreajuste. En contraste, 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.



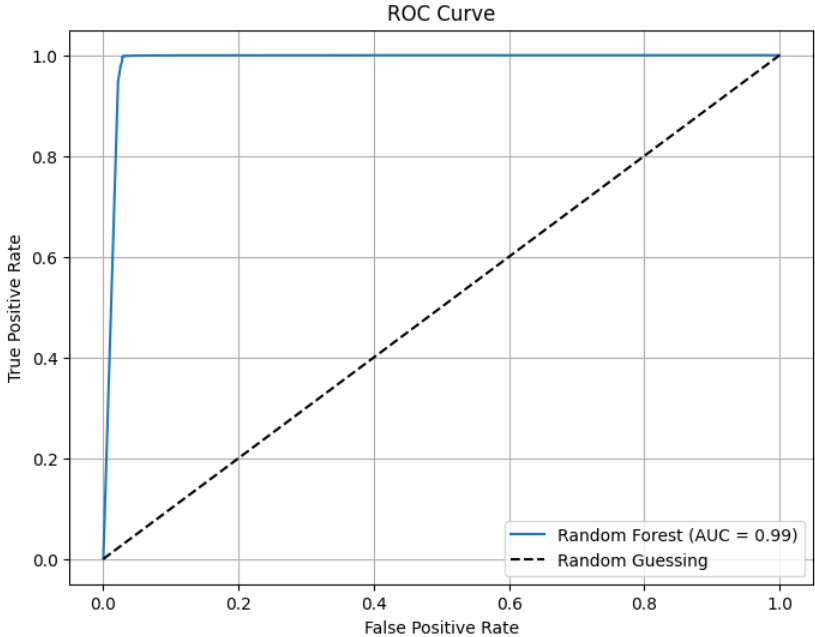
Dado que ambos modelos ofrecen la misma precisión, pero XGBoost requiere un tiempo de entrenamiento mucho menor, se recomienda optar por XGBoost por su eficiencia en términos de tiempo y recursos. Sin embargo, dado el tamaño y la naturaleza del conjunto de datos, técnicas de ensamble como Random Forest 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.

**¿El rendimiento del modelo es lo suficientemente bueno para su implementación en producción?**

Sí, el rendimiento del modelo es lo suficientemente bueno para su implementación en producción. Ambos modelos, XGBoos y Random Forest, muestran un rendimiento excepcionalmente alto con un “Best Score” de 0.998, indicando una precisión casi perfecta. XGBoost es especialmente eficiente con un tiempo de entrenamiento de aproximadamente 639 segundos, lo que lo hace ideal para entornos donde el tiempo de entrenamiento es crítico. Random Forest, aunque requiere más tiempo de entrenamiento, sigue siendo una opción robusta y altamente efectiva, particularmente adecuada para conjuntos de datos con variables numéricas y categóricas.

Además, la precisión del modelo Random Forest de 0.9976, junto con una matriz de confusión que muestra una alta tasa de aciertos y un AUC de 0.988, sugiere un excelente desempeño en la discriminación entre clases. Estos resultados indican que el modelo no solo es preciso, sino también robusto y confiable para su aplicación en contextos reales. Sin embargo, es importante seguir monitoreando el rendimiento del modelo en producción y considerar otros factores como la capacidad de generalización y el costo computacional dependiendo del contexto y los recursos disponibles.

**¿Existe margen para mejorar aún más el rendimiento?**

Sí, siempre existe margen para mejorar el rendimiento de los modelos, incluso cuando los resultados son ya muy altos. Algunas estrategias que podrían aplicarse para intentar mejorar aún más el rendimiento pueden ser:

El ajuste de hiperparámetros, realizando una búsqueda más exhaustiva de hiperparámetros utilizando técnicas como Grid Search en lugar de Randomized Search. Explorar combinaciones adicionales de hiperparámetros que no se han considerado.

La ingeniería de características, creando nuevas características a partir de las existentes (feature engineering) que puedan proporcionar información adicional al modelo. Probar con técnicas de selección de características para eliminar características irrelevantes o redundantes que podrían estar afectando negativamente el modelo.

Probar modelos adicionales que no se hayan considerado y ensamblarlos con los modelos existentes. Revisar y mejorar las técnicas de preprocesamiento, como la normalización o estandarización de los datos. Aunque los modelos actuales ya muestran un rendimiento excepcional, explorar estas áreas pueden proporcionar pequeñas mejorar adicionales que podrían ser significativas dependiendo del contexto y los requisitos específicos de la aplicación en producción.

**¿Cuáles serían las recomendaciones clave para poder implementar la solución?**

Realizar validaciones adicionales en datos no vistos para confirmar la robustez del modelo seleccionado. Esto puede ayudar a garantizar que el modelo generalice bien a nuevas muestras y no esté sobreajustado.

Ajustar los parámetros del modelo seleccionado utilizando técnicas de búsqueda más exhaustivas, como Grid Search, para maximizar el rendimiento. Implementar un sistema de monitoreo continuo para evaluar el rendimiento del modelo en producción y detectar posibles desviaciones o degradaciones en el rendimiento. Considerar la escalabilidad del modelo para manejar mayores volúmenes de datos en producción. Asegurarse de que la infraestructura y los recursos computacionales sean suficientes para soportar la carga de trabajo esperada.

Documentar adecuadamente el proceso de desarrollo del modelo, incluidos los pasos de preprocesamiento de datos, selección de características, ajuste de hiperparámetros y evaluación del modelo. Además, establecer un plan de mantenimiento para actualizar y mejorar el modelo con el tiempo.

Integrar el modelo en el flujo de trabajo existente de la organización, definir claramente los stakeholders y las personas interesadas en participar en la implementación del modelo, asegurándose de que sea compatible con otras herramientas y sistemas utilizados. Proporcionar capacitación adecuada al personal encargado de operar y mantener el modelo en producción, asegurándose de que comprendan cómo utilizarlo correctamente y cómo interpretar los resultados. Al definir los stakeholders, se asegura una comunicación efectiva y una colaboración adecuada entre todas las partes interesadas durante todas las etapas del proyecto. Establecer un proceso de gestión de cambios para realizar actualizaciones y mejoras en el modelo de manera controlada y minimizar el impacto en la producción.

**¿Qué tareas / procedimientos son accionables para las partes interesadas (stakeholders)?**

Algunas tareas y procedimientos accionables para los stakeholders tanto internos como externos de la organización son:

**Equipo de Desarrollo de Modelos**

Desarrollar el modelo Random Forest de acuerdo con los requisitos y estándares establecidos. Realizar pruebas exhaustivas del modelo para garantizar su funcionalidad y precisión. Documentar el proceso de desarrollo del modelo y proporcionar información técnica detallada para su revisión.

**Equipo de TI / Ingeniería**

Implementar el modelo en el entorno de producción de la organización de manera segura y eficiente. Configurar sistemas de monitoreo y alerta temprana para detectar cualquier problema con el modelo en tiempo real. Garantizar la integración adecuada del modelo con otros sistemas y procesos existentes.

**Equipo de Operaciones**

Coordinar con el equipo de TI para asegurar una implementación suave y sin interrupciones del modelo en producción. Establecer protocolos de escalación para manejar cualquier problema o emergencia relacionada con el modelo. Proporcionar capacitación y soporte continuo a los usuarios finales que interactúan con el modelo.

**Equipo de Gestión / Directivos**

Revisar y aprobar el plan de implementación del modelo, así como asignar los recursos necesarios para su ejecución. Supervisar el progreso de la implementación y tomar decisiones estratégicas basadas en los resultados del modelo. Comunicar los beneficios y la importancia del modelo a otras partes interesadas dentro de la organización.

**Clientes / Usuarios Finales**

Proporcionar retroalimentación sobre la usabilidad y efectividad del modelo una vez implementado. Comunicar cualquier problema o dificultad encontrada al interactuar con el modelo en producción. Utilizar el modelo de manera efectiva para tomar decisiones informadas y mejorar los resultados comerciales.

**Socios Comerciales**

Colaborar con la organización para integrar el modelo en procesos comerciales compartidos. Proporcionar datos adicionales o insights relevantes que puedan mejorar el rendimiento del modelo. Participar en discusiones estratégicas sobre cómo aprovechar mejor los resultados del modelo para beneficio mutuo.

**Reguladores / Entidades de Cumplimiento**

Asegurarse de que la implementación del modelo cumpla con todas las regulaciones y normativas aplicables. Proporcionar orientación y supervisión sobre el uso ético y legal de los datos y las predicciones del modelo. Participar en auditorías y revisiones periódicas para garantizar la transparencia y responsabilidad en el uso del modelo.

Al asignar roles y responsabilidades específicos a cada grupo de stakeholders, se puede garantizar una implementación exitosa y efectiva del modelo Random Forest, maximizando así su valor y beneficio tanto para la organización como para sus socios y clientes.