Proyecto Inteligencia Artificial, Informe Final

Daniel Lujan Agudelo

Emanuel López Higuita

Santiago Rivera Montoya

Modelos y Simulación de Sistemas I

Facultad de Ingeniería, Universidad de Antioquia

Octubre 2023

1. Introducción

Home Credit es una institución financiera internacional que se enfoca en el préstamo de dinero a personas con poco o nulo historial crediticio.

Con el fin de buscar mayor rentabilidad, se requiere un modelo de Machine Learning (ML) que prediga con qué probabilidad un cliente solicitante de un crédito, dada una serie de datos personales y financieros, pagará su deuda debidamente.

2. Exploración de datos

La empresa dispuso a través de una competición de Kaggle, un conjunto de datos de sus clientes para entrenar un modelo de ML.

El dataset cuenta con un total de 307,511 registros de clientes, para los que se tienen 122 variables o columnas.

El dataset cuenta con un total de 122 variables, de las cuales 106 son continuas y 16 son categóricas. Estas últimas se describen a continuación (Tabla 1):

NAME_CONTRACT_TYPE	Identificación si el préstamo es en efectivo o revolvente.
CODE_GENDER	Género del cliente.
FLAG_OWN_CAR	Marca si el cliente posee un carro.
FLAG_OWN_REALTY	Marca si el cliente posee una casa o un piso.
NAME_TYPE_SUITE	Quien acompañaba al cliente cuando solicitaba el préstamo.
NAME_INCOME_TYPE	Tipo de ingreso del cliente (empresario, trabajador, baja por maternidad, etc).
NAME_EDUCATION_TYPE	Nivel de educación más alto que alcanzó el cliente.

NAME_FAMILY_STATUS	Estado familiar del cliente.
NAME_HOUSING_TYPE	Cuál es la situación de vivienda del cliente (de alquiler, viviendo con los padres, etc).
OCCUPATION_TYPE	Qué tipo de ocupación tiene el cliente.
FONDKAPREMONT_MODE	Información normalizada sobre el edificio donde vive el cliente.
WALLSMATERIAL_MODE	Información normalizada sobre el edificio donde vive el cliente.
EMERGENCYSTATE_MODE	Información normalizada sobre el edificio donde vive el cliente.
WEEKDAY_APPR_PROCESS_START	En qué día de la semana el cliente solicitó el préstamo.
ORGANIZATION_TYPE	Tipo de organización en la que trabaja el cliente.

Tabla 1

Además, es necesario analizar la información faltante en las columnas. Algunos datos a destacar:

- La columna OWN_CAR_AGE, que proporciona la edad del vehículo de la persona **en caso de tenerlo**, tiene sólo 34% de los datos.
- La columna OCCUPATION_TYPE, que proporciona la ocupación de la persona, tiene un 69% de los datos. Nótese que la información que proporciona esta columna es fundamental para la predicción.

3. Preprocesamiento de datos

Antes de usar algún modelo, es necesario normalizar/procesar los datos, eliminando información irrelevante, y reemplazando la información que falte de la forma más adecuada posible. Utilizamos una función en la cuál se realizan todos los procesos tanto de limpieza como de rellenado de datos faltantes.

3.1. Rellenado de datos faltantes

El primer proceso por el que pasan los datos es por el rellenado de los datos faltantes.

Para el caso de las variables continuas, se decidió que todos los datos faltantes serían reemplazados con el valor medio de la columna correspondiente.

Mientras que para las variables categóricas, para esta primera iteración, se reemplazaron los datos faltantes con la categoría más común en la columna. Esto puede no ser adecuado para algunas variables.

Por ejemplo, en la columna OCCUPATION_TYPE, el valor más común es el de Empleado (Ver Figura 1), sin embargo, lo ideal sería agrupar a las personas con el dato faltante en una nueva categoría 'Desempleado'.

	OCCUPATION_TYPE
Laborers	151577
Sales staff	32102
Core staff	27570
Managers	21371
Drivers	18603
High skill tech staff	11380
Accountants	9813
Medicine staff	8537
Security staff	6721
Cooking staff	5946
Cleaning staff	4653
Private service staff	2652
Low-skill Laborers	2093
Waiters/barmen staff	1348
Secretaries	1305
Realty agents	751
HR staff	563
IT staff	526

(Figura 1)

3.2. Variables categóricas a numéricas

Para transformar las variables categóricas a números y poder usarlas como datos de entrenamiento para el modelo, se usó One-Hot Encoding sobre todas las variables tabuladas en la Tabla 1.

3.3. Posibles variables innecesarias

Dentro del dataset, existen algunas variables que pueden no ser de utilidad para predecir si un cliente pagará o no el crédito, y que, en cambio, inducen error en el modelo. Una vez identificadas estas columnas, es conveniente eliminarlas del conjunto de datos de entrenamiento.

Para la primera predicción, no se ha quitado ninguna columna. Sin embargo, se han identificado algunas variables que posiblemente deban ser eliminadas:

- En qué día de la semana el cliente solicitó el préstamo (WEEKDAY_APPR_PROCESS_START).
- Quien acompañaba al cliente cuando solicitaba el préstamo (NAME_TYPE_SUITE).

4. Primer modelo predictivo con Random Forest

La primera predicción hecha se realizó usando el modelo **Random Forest Classifier** usando 100 árboles (o estimadores).

4.1. Resultados

Usando el dataset procesado y el dataset para pruebas que la competición incluye, se obtuvieron los siguientes resultados:

SK_ID_CURR	100001	100005	100013	100028	100038	100042	100057	100065	100066	100067	...
TARGET	0.11	0.11	0.08	0.06	0.1	0.05	0.02	0.05	0.06	0.23	...

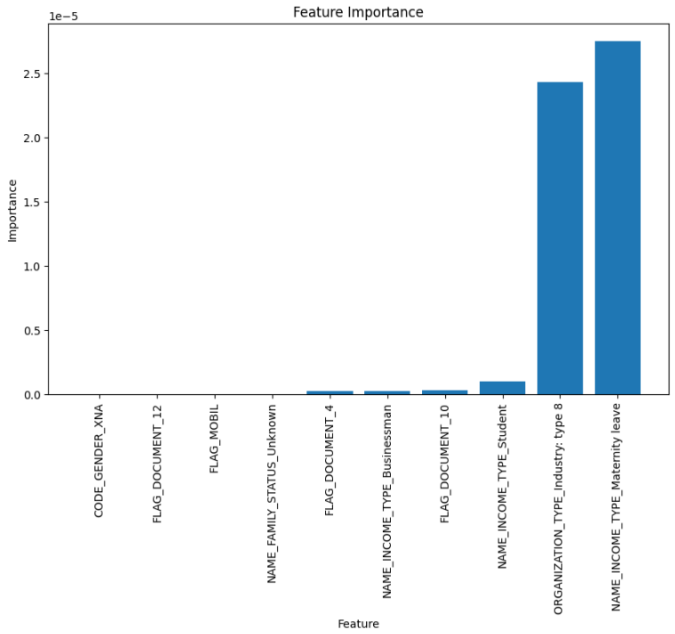
(Figura 2)

Dónde TARGET es la variable objetivo y significa la probabilidad de que el cliente pague su deuda.

El puntaje obtenido usando el **área bajo la curva ROC** como métrica de desempeño, al subir estos resultados a Kaggle fue de ≈ 69.6%

Según el modelo entrenado, los dos factores más influyentes en la probabilidad de que el cliente pague su deuda son los siguientes (ver Figura 3):

- Si los ingresos de la persona provienen del permiso de maternidad.
- Si el tipo de organización en la que trabaja el cliente es una empresa.



(Figura 3)

Otro dato a destacar sobre los resultados, es que el promedio de la probabilidad de que el cliente pague es, aproximadamente, 8.53%. Este porcentaje es similar al de personas que pagaron su deuda satisfactoriamente en el dataset de entrenamiento (promedio de la variable objetivo): 8.07%.

4.2. Mejores hiperparámetros

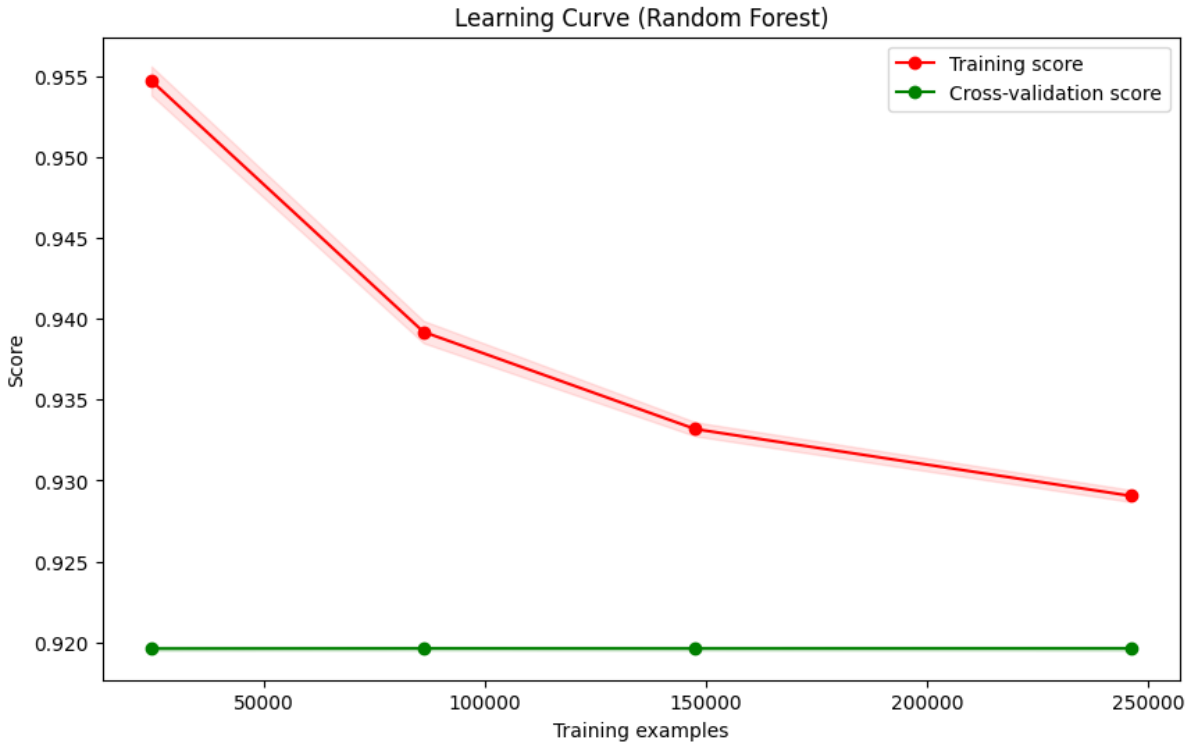
Luego, se realizó otra iteración con el mismo modelo, pero antes buscando los parámetros que mejores resultados arrojaban para nuestro conjunto de datos.

Se realizaron pruebas con 10, 50, 100 y 200 árboles, en combinación con 10 o 20 niveles de máxima profundidad de los árboles. Es decir, 8 diferentes combinaciones de parámetros para el modelo.

De las pruebas realizadas, se obtuvo que el modelo generaba mejores resultados con 100 árboles y con 20 niveles de profundidad máxima.

Usando estos parámetros, se logró un resultado un poco mejor, con un acierto de aproximadamente 70.46%

4.3. Curva de aprendizaje



La gráfica obtenida para este modelo indica que mientras mayor es la cantidad de datos de entrenamiento, peores resultados se obtienen. Parece indicar que el modelo no está siendo capaz de captar la profundidad de los datos y la relación entre ellos. Esto se puede deber a que el modelo es muy simple, mientras que los datos de entrenamiento son más complejos.

5. Modelo con Logistic Regression

Luego, se decidió intentar con un modelo predictivo diferente, en este caso, Logistic Regression. Para la primera iteración usando este modelo, se utilizaron los parámetros por defecto y se obtuvieron los siguientes resultados:

Es decir, un resultado considerablemente peor que el obtenido con el modelo de Random Forest.

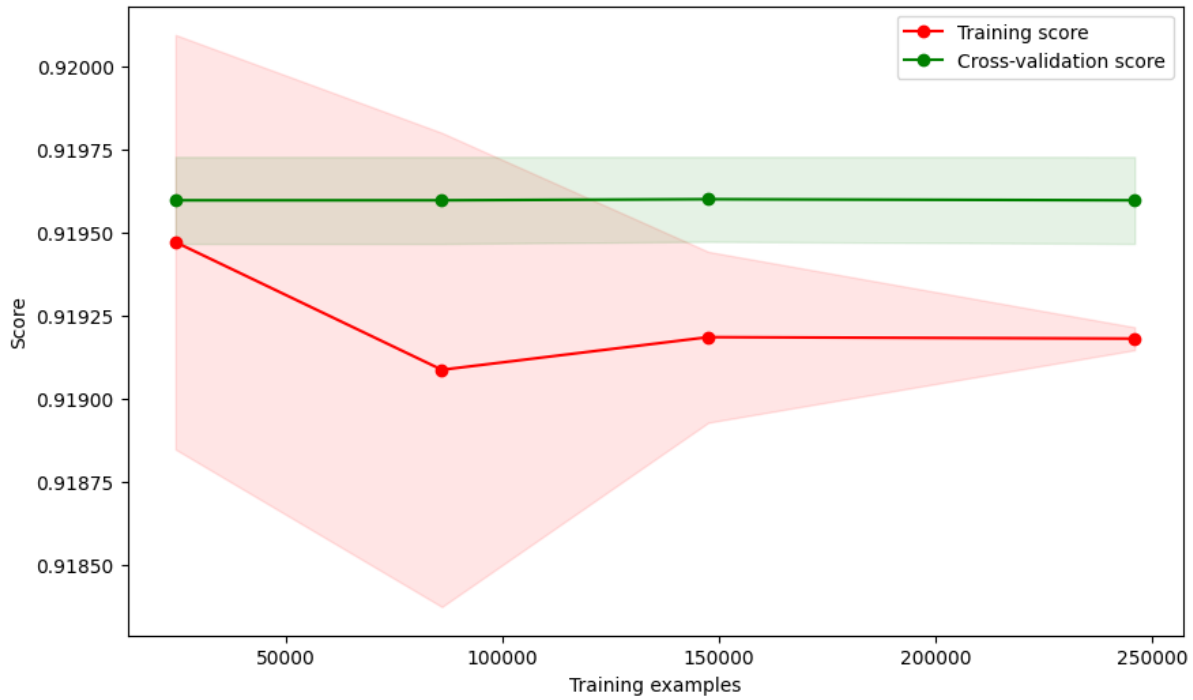
En un intento para obtener mejores resultados, también se realizó la predicción personalizando los parámetros del modelo.

Se probó con distintas formas de penalización: L1, L2 y elasticnet, y valores de C (que es inverso a la fuerza de regularización) de 0.1 y 1.0.

La combinación de hiperparámetros que mejores resultados arrojó es L2 y 0.1.

Sin embargo, no se obtuvieron mejores resultados predictivos.

5.1. Curva de aprendizaje



Como se puede observar en la gráfica obtenida, la variación en la curva de aprendizaje es muy alta, lo que puede indicar que el modelo es extremadamente sensible a los datos de entrenamiento, y al entrenarlo con distintos subconjuntos del dataset original, produce resultados radicalmente diferentes.

6. Modelos no supervisados

Debido a los malos resultados obtenidos al usar el modelo de Logistic Regression, decidimos mantener el modelo de Random Forest y los hiperparámetros hallados para las siguientes iteraciones que combinan modelos no supervisados con el modelo mencionado.

6.1. Clustering con KMeans

Para la siguiente iteración, se decidió usar clustering con el algoritmo KMeans para generar un nuevo dataset de entrenamiento para el modelo de Random Forest.

Antes de proceder con la predicción, usando el puntaje de Silhouette, se encontró que el mejor número de clústeres para el algoritmo de KMeans es 6, teniendo en cuenta que se realizaron pruebas con 6, 12 y 20 clústeres.

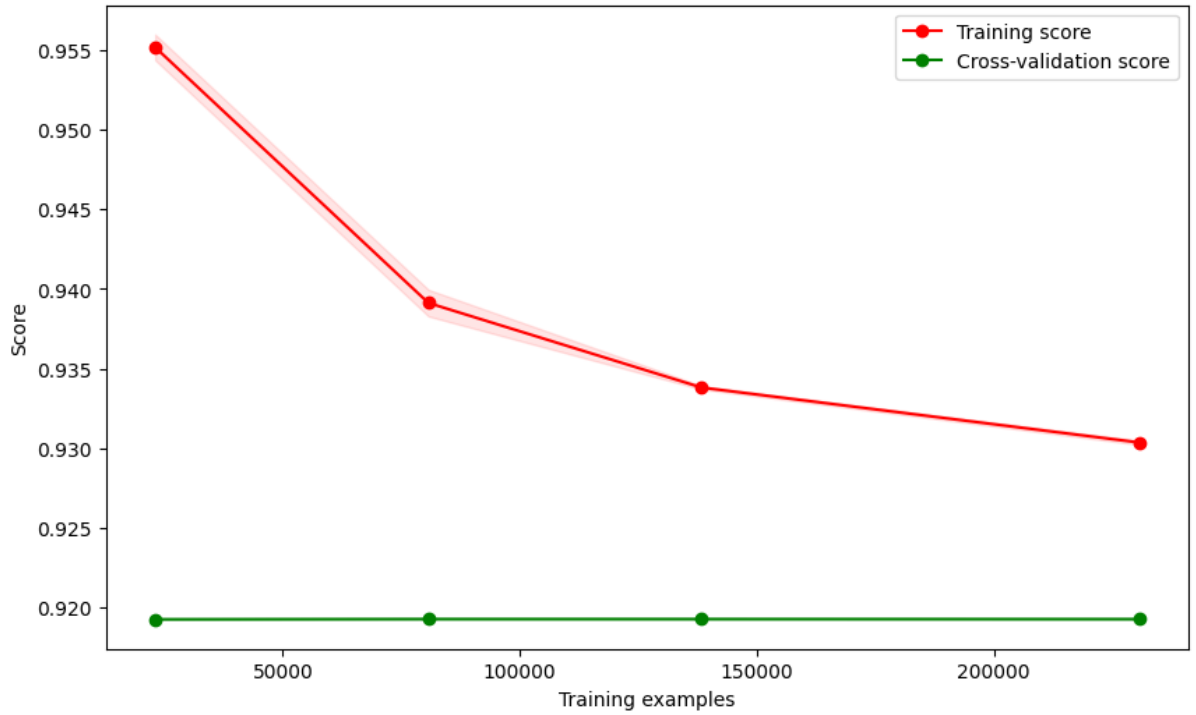
Luego, se realizó el respectivo clústering sobre el conjunto de datos, y con el fin de usar la información extraída de KMeans, se añadió una columna al dataset representando al número de clúster al que pertenece cada registro del dataset, y se usó el resultado como entrenamiento del modelo predictivo.

6.1.1. Resultados

Realizando el procedimiento anterior, se obtuvo el siguiente puntaje:

Es decir, se mejoró el resultado obtenido con respecto a la predicción realizada anteriormente usando sólo Random Forest.

6.1.2. Curva de aprendizaje



6.2. Análisis de componentes principales (PCA)

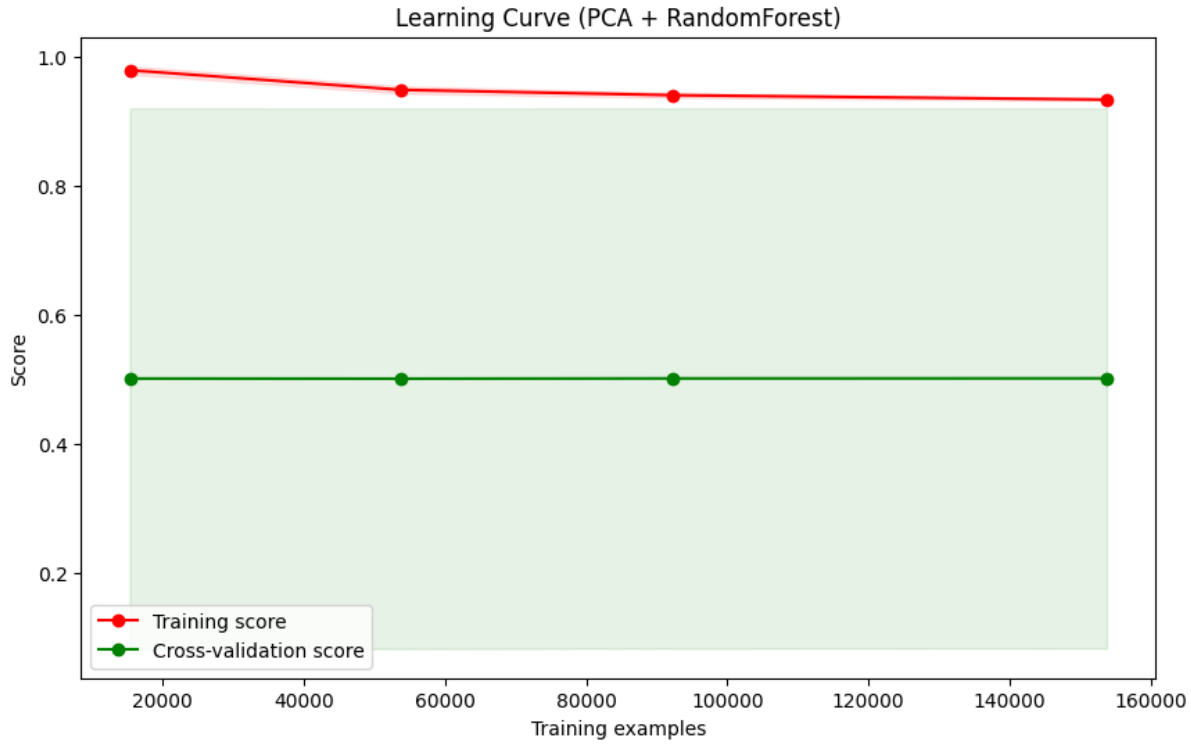
Como alternativa, se planteó usó PCA para reducir la dimensionalidad del conjunto de datos antes de entrenar el modelo predictivo.

Al igual que con los algoritmos anteriores, se buscaron los mejores parámetros. En este caso, se realizaron tests sobre el parámetro de número de componentes para el algoritmo de PCA. Se probó con los valores de 5, 10 y 20, y se obtuvo que el mejor resultado para el dataset se generaba con el valor 20.

6.2.1. Resultados

Se planteó un Pipeline, que se encarga de aplicar el algoritmo de PCA al dataset, y el resultado usarlo para entrenar el modelo de Random Forest. Realizando este proceso, se obtuvieron los siguientes resultados:

6.2.2. Curva de aprendizaje



7. Retos y consideraciones de despliegue

Uno de los retos principales, y que inicialmente representó la primera dificultad con la que nos enfrentamos, fue el preprocesamiento del dataset. Este desafío adquirió una relevancia significativa debido a la considerable cantidad de filas presentes en el conjunto de datos. La magnitud de esta información generó obstáculos iniciales, ya que la manipulación y la preparación de un conjunto de datos extenso suelen ser tareas complejas.

El proceso de preprocesamiento es fundamental para garantizar la calidad y la utilidad de los datos en cualquier análisis posterior. En este contexto, abordar un dataset con un gran número de filas implica considerar aspectos como la limpieza de datos, la identificación y manejo de valores atípicos, la normalización de variables y la gestión de posibles valores faltantes.

La amplitud del dataset no solo aumenta la complejidad de estas tareas, sino que también puede tener implicaciones en términos de eficiencia computacional. La necesidad de optimizar los procesos para trabajar de manera efectiva con grandes conjuntos de datos se convierte en una prioridad, y la selección de técnicas de preprocesamiento adecuadas se vuelve esencial para garantizar resultados precisos y significativos en las fases posteriores del análisis.

Por otro lado, otro reto que tuvimos al realizar las predicciones fue tener en cuenta las combinaciones de los hiperparámetros, debido a que entre mayor número de combinaciones, mayor era el tiempo de ejecución que tomaba en ejecutarse.

En respuesta a esto, implementamos estrategias de optimización y técnicas de búsqueda de hiperparámetros que nos permitieron equilibrar la exhaustividad de la exploración con la gestión eficiente del tiempo de ejecución, garantizando así un proceso de predicción más efectivo y viable en términos de recursos computacionales.

Antes de desplegar un modelo, es crucial establecer un nivel de desempeño mínimo basado en métricas relevantes. El proceso de despliegue debe incluir validación cruzada, implementación gradual y un plan de reversión. El monitoreo en tiempo real, alertas automáticas, y reentrenamiento automático son fundamentales para garantizar el rendimiento sostenible del modelo. La gestión de versiones y el cumplimiento normativo y ético son consideraciones esenciales. La documentación clara y la comunicación efectiva

entre equipos aseguran respuestas rápidas a posibles problemas durante el despliegue y la operación del modelo en producción.

8. Conclusiones

Importancia del Preprocesamiento de Datos:

El desafío inicial de manejar un conjunto de datos extenso destaca la importancia crítica del preprocesamiento de datos. Estrategias eficientes y técnicas especializadas son esenciales para abordar conjuntos de datos extensos, garantizando la calidad y utilidad de la información.

Optimización de Hiperparámetros:

La exploración de múltiples combinaciones de hiperparámetros durante la predicción destaca la relevancia de la optimización para mejorar el rendimiento del modelo. Estrategias de búsqueda eficientes son esenciales para equilibrar la precisión del modelo con la eficiencia computacional.

Resultados del Modelo Random Forest:

El modelo Random Forest demostró un rendimiento prometedor con un área bajo la curva ROC del 69.6%. La identificación de factores influyentes subraya la capacidad del modelo para revelar patrones relevantes en el conjunto de datos.

Desafíos con Logistic Regression:

La exploración de Logistic Regression reveló desafíos significativos, con un rendimiento inferior al modelo Random Forest. La sensibilidad extrema a los datos de entrenamiento sugiere limitaciones en la capacidad de este modelo para capturar la complejidad de los datos.

Enfoque de Modelos no Supervisados:

La combinación de modelos no supervisados, como KMeans y PCA, con el modelo Random Forest proporcionó mejoras en la predicción. La utilización de clustering y la reducción de dimensionalidad demuestran la utilidad de enfoques complementarios para abordar la complejidad del problema, mejorando el rendimiento predictivo.