Coder House - Curso de Data Science

Proyecto Final: Sistema crediticio

Autores:
- Adrián Spesot
- Alejo Romano
- Armando López

Profesor:
- Octavio Lafourcade

Tutor:
- Johan Caceido

Comisión: 31490

Fecha de entrega: 24/10/2022

Índice

Índice 1

Descripción del caso de negocio 3

Tabla de versionado 4

Objetivos 4

Diccionario de datos 5

Análisis Exploratorio de Datos (EDA) 7
- Análisis univariado: 8
  - Variables Categóricas 8
  - Variables Numéricas 8
- Análisis bivariado 9
  - Preguntas 9
- Análisis multivariado 11
  - Preguntas 11

Codificación 13

Análisis de componentes principales 13

Algoritmos de Clasificación 14
- K-Nearest Neighbor (KNN) 15
- Decision Tree 15
- Random Forest 15
- Random Forest con PCA 15
- Support Vector Machine (SVM) 16
- Logistic Regression Multiclase 16
- Resultados de modelos multiclase 16
- Pipelines 17
  - Modelo Agrupado 1 17
  - Modelo Agrupado 2 17
  - Resultados Obtenidos 17

Selección de modelo 19
- Mejora del modelo - Random Forest 19
  - Evaluar variables numéricas 19
  - Selección de variables numéricas según relevancia 19
  - Agregar variables categóricas 20
  - Selección de variables categóricas según relevancia 21
  - Balanceo 21
  - Grid Search 22

Matrices de confusión 22
Mejora del modelo - XGBClassifier 23
Evaluar todas las variables 23
Selección de variables según relevancia 24
Balanceo 25
Ajuste de hiperparámetros 25
Matrices de confusión 26
Futuras líneas 27
Conclusión 28

Descripción del caso de negocio

El sistema crediticio es el conjunto de instituciones crediticias, que incluye la autoridad monetaria del país y las entidades de créditos, que actúan como intermediarios entre los agentes económicos que ofrecen y demandan dinero a través de créditos. Las entidades que ofrecen créditos captan el ahorro del público y, con ese capital, solventan el otorgamiento de créditos y realizan inversiones.

Ahora bien, existe un riesgo en este sistema crediticio que es el riesgo de impago, es decir, que las personas físicas o jurídicas que toman un crédito no cumplan con el pago del mismo. Esto supone un problema para las entidades bancarias que brindan los créditos, dado que no reciben el dinero prestado más los intereses pactados.

En la actualidad, las empresas financieras buscan optimizar el sistema de créditos para reducir el riesgo de incumplimiento de pago mencionado. Con el avance de la inteligencia artificial, se crearon modelos que permiten reducir distintos riesgos de manera automática. En particular, en el sistema crediticio se utilizan algoritmos de machine learning (modelos de “caja blanca”) que buscan predecir el comportamiento de un nuevo tomador de crédito.

Para disminuir el riesgo de impago mencionado, la gerencia de la empresa pidió a su equipo de data science que diseñen un modelo de clasificación para separar a las personas tomadoras de créditos en grupos según su scoring crediticio. Para esto, la compañía recolectó información relacionada con los créditos otorgados en los últimos años.

Algunas preguntas de investigación que surgen de este problema y que nos ayudan en el proceso de construcción del proceso son las siguientes:

- ¿Cómo es el perfil de una persona con alto riesgo de impago?
- ¿Es posible predecir si una persona tiene alto riesgo de impago mediante un modelo de clasificación?
- ¿Cuáles son las variables más importantes para la clasificación según el scoring crediticio? ¿Cuáles son las variables menos importantes?

Tabla de versionado

En la siguiente tabla se puede observar las versiones que se entregaron del presente trabajo para su revisión.

Versiones	Fecha
Versión 01 - 1era preentrega	20/08/2022
Versión 02 - 2da preentrega	10/09/2022
Versión 03 - 3era preentrega	01/10/2022
Versión 04 - Entrega final	24/10/2022

Objetivos

El objetivo principal de este proyecto es desarrollar un algoritmo de Machine Learning que permita evaluar distintos parámetros y categorizar la calidad del crédito en bueno, estándar o malo, con el fin de disminuir el riesgo de impago en créditos futuros.

En línea con esto, algunos objetivos específicos son: probar distintos algoritmos de machine learning para la creación del modelo; crear un modelo de clasificación que pueda predecir, la categoría, con una exactitud mayor a 85%; crear un modelo rápido para tener un resultado en el menor tiempo posible y agilizar el proceso de otorgamiento de créditos.

Diccionario de datos

Fuente: https://www.kaggle.com/datasets/parisrohan/credit-score-classification

Número de columnas: 28

Cantidad de registros: 100000

Contexto: Esta base de datos proporciona información sobre la adquisición de créditos personales por personas físicas. Este tema es importante a nivel macroeconómico en todos los países porque así se puede analizar y valorar la calidad crediticia de una población. A su vez, esto ayuda a las instituciones financieras a crear productos de crédito acorde a las necesidades de los clientes con una exposición al riesgo menor. No se especifica la procedencia de los datos, la moneda ni el nombre de la institución financiera.

Descripción de las variables:

Variable	Tipo de dato	Descripción
ID	int	Identificación del registro
Customer_ID	int	Identificación del cliente
Month	str	Mes de la toma del crédito
Name	str	Nombre del cliente
Age	int	Edad del cliente
SSN	int	Número de seguro social
Occupation	str	Ocupación del cliente
Annual_Income	float	Ingreso anual del cliente
Monthly_Inhand_Salary	float	Ingreso mensual en mano del cliente
Num_Bank_Accounts	int	Número de cuentas de banco
Num_Credit_Card	int	Número de tarjetas de crédito
Interest_Rate	int	Tasa de interés del crédito
Num_Loan	int	Cantidad de créditos
Type_of_Loan	str	Tipo de crédito
Delay_from_due_date	int	Retraso desde fecha de vencimiento
Changed_Credit_Limit	float	Cambio en límite de crédito
Num_Credit_Inquiries	int	Número de consultas crediticias
Credit_Mix	str	Mezcla crediticia

Variable	Tipo de dato	Descripción
Outstanding_Debt	float	Deuda pendiente
Credit_Utilization_Ratio	float	Ratio de utilización de crédito
Credit_History_Age	str	Años con historial crediticio
Payment_of_Min_Amount	str	Pago de cantidad mínima
Total_EMI_per_month	float	Pago total por mes
Amount_invested_monthly	float	Cantidad invertida mensual
Payment_Behaviour	str	Comportamiento crediticio
Monthly_Balance	float	Balance mensual
Credit_Score	str	Variable tipo target que indica la calidad del préstamo

Análisis Exploratorio de Datos (EDA)

En primer lugar, se observa que efectivamente el dataframe cuenta con 28 columnas y 100000 registros. Además, antes de realizar modificaciones, se verifica que el dataframe no tenga datos duplicados.

Como primera modificación, se eliminan las variables **ID**, **Custumor_ID**, **Name** y **SSN** dado que son variables identificadoras y, por ende, no son informativas para el modelo. Además, se elimina **Credit_History_Age** ya que no se considera importante para el modelo.

Luego, se observan dos problemas con el resto de las variables. Por un lado, la mayoría están definidas como tipo **object**, lo cual es erróneo. Por otro lado, variables como **Age**, **Annual_Income**, **Num_Loan** y otras, presentan guiones bajo en algunos registros (p.ej. ingreso anual de 44955.64_). Por lo tanto, se eliminan estos caracteres especiales y se corrigen los tipos de datos (ver Figura 1).

Month	category
Age	int64
Occupation	category
Annual_Income	float64
Monthly_Inhand_Salary	float64
Num_Bank_Accounts	int64
Num_Credit_Card	int64
Interest_Rate	int64
Num_of_Loan	int64
Type_of_Loan	category
Delay_from_due_date	int64
Num_of_Delayed_Payment	float64
Changed_Credit_Limit	float64
Num_Credit_Inquiries	float64
Credit_Mix	category
Outstanding_Debt	float64
Credit_Utilization_Ratio	float64
Payment_of_Min_Amount	category
Total_EMI_per_month	float64
Amount_invested_monthly	float64
Payment_Behaviour	category
Monthly_Balance	float64
Credit_Score	object

Figura 1: listado de las variables con el tipo de datos asignado

Hay que destacar que no se corrigió el tipo de dato de la variable respuesta, **Credit_Score**. Esto se debe a que, luego del data wrangling, se realiza su correspondiente codificación.

Por otra parte, se decide realizar el estudio para el rango etario comprendido entre 18 a 56 años, por lo que se aplica un filtro a la variable edad.

Finalmente, se analizó los datos faltantes en el dataframe. Las siguientes variables presentan missing values:

- Monthly_Inhand_Salary: 15001
- Type_of_Loan: 11408
- Num_of_Delayed_Payment: 7002
- Amount_Invested_Monthly: 4478
- Monthly_Balance: 2868
- Changed_Credit_Limit: 2091
- Num_Credit_Inquiries: 1965

Para estas variables, con excepción de Monthly_Inhand_Salary, se imputaron los datos faltantes con la mediana, dado que la distribución era asimétrica. La variable de salario mensual en mano se analizó aparte por dos razones: presenta muchos datos faltantes (15% aprox.) y parece tener relación con la variable **Annual_Income**, dado que ambas variables hacen referencia al salario. Por lo tanto, se realizó un análisis de correlación entre las dos variables, obteniéndose un valor aproximado de 0.998. Este valor indica una correlación casi perfecta entre las dos variables, por lo que se decide eliminar **Monthly_Inhand_Salary**.

A continuación, se efectúa un análisis univariado, bivariado y multivariado para obtener más información sobre cada variable y la relación que pueden tener entre ellas.

Análisis univariado:

Variables Categóricas

- **Month:** se observa que todos los meses presentan casi la misma cantidad de registros crediticios.
- **Occupation:** se observa que todas las ocupaciones presentan cantidades parecidas. La que más se destaca es la profesión de Abogados y la que menos se destaca es Manager.
- **Credit_Mix:** se observa como más frecuente una mezcla crediticia estándar y la menos frecuente es la mezcla crediticia mala.
- **Payment_of_Min_Amount:** se observa que hay más proporción de “Yes” que de “No”.
- **Payment_Behaviour:** se observa que el nivel de comportamiento destacado son los créditos a aquellas personas con bajos gastos y pequeños pagos mientras que la menos frecuente son las personas con bajos gastos y altos pagos.
- **Credit_Score:** se observa que el puntaje crediticio que más se destaca es el regular mientras que el menos relevante es el bueno.

Variables Numéricas

- **Age:** se aprecia una mediana de edad de 33 años y tenemos una distribución bastante uniforme con mayor presencia entre los 18 y 45 años, en menor medida personas de edades entre 45 y 56 años. A su vez, se observa que no hay presencia de outliers.
- **Interest_Rate:** se observa una gran cantidad de valores outliers y se debe realizar un tratamiento para buscar un mejor análisis.
- **Delay_from_due_date:** se observa una escasa cantidad de outliers. La mediana se encuentra en 20, y en el gráfico de distribución se pueden identificar 2 segmentos bien marcados: uno en el rango de 10-30 y otro entre 30-60 aproximadamente.
- **Num_of_Delayed_Payment:** idem a Interest_Rate.
- **Num_Credit_Inquiries:** idem a Interest_Rate.
- **Outstanding_Debt:** tiene una mediana en los 1100 y se observa presencia de outliers. En el gráfico de distribución se puede identificar 3 segmentos: la más frecuente entre 0 y 1500, seguida de otro segmento con una frecuencia intermedia entre los 1500 y 2700 y por último, en menor frecuencia, desde los 2700 a los 5000.

- **Credit_Utilization_Ratio:** en general, se observa que la media de utilización de crédito disponible está en los 33%, y que el porcentaje de utilización de los créditos se concentran entre los 25% y el 38%
- **Total_EMI_per_month:** se observa una gran cantidad de valores outliers.

Análisis bivariado

Al realizar la matriz de correlación, se encontró que los siguientes pares de variables presentan una correlación mayor a 0.4:

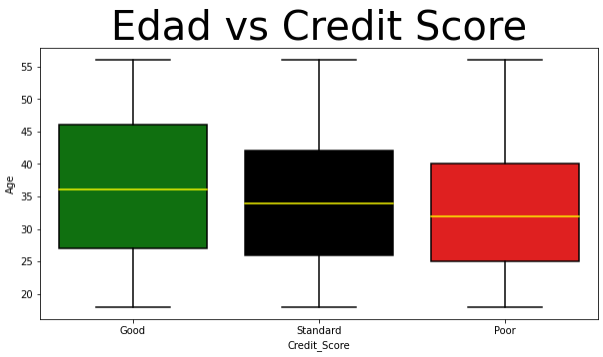
Variable 1	Variable 2	Correlación
Annual_Income	Monthly_Balance	66.84%
Annual_Income	Total_EMI_per_month	61.85%
Num_Credit_Inquiries	Interest_Rate	61.93%
Num_Credit_Inquiries	Delay_from_due_date	52.02%
Num_Credit_Inquiries	Outstanding_Debt	59.61%
Interest_Rate	Delay_from_due_date	57.31%
Interest_Rate	Outstanding_Debt	62.74%
Delay_from_due_date	Outstanding_Debt	55.80%
Changed_Credit_Limit	Outstanding_Debt	42.15%

Por otro lado, al realizar el pairplot y algunos scatterplot se encontró patrones interesantes entre las variables. Por ejemplo, algunas combinaciones de variables presentan bloques de puntos que limitan con zonas “prohibidas” (zonas donde no hay puntos), mientras que otras combinaciones presentan tendencias crecientes.

Preguntas

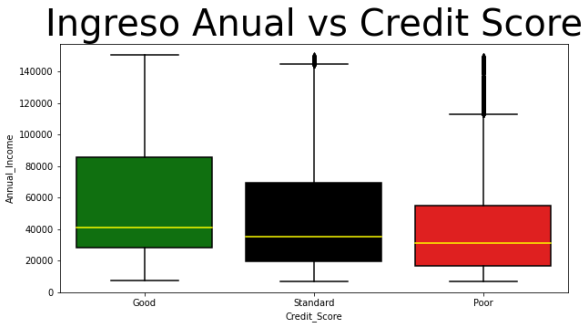
1. ¿Cómo es el riesgo crediticio según la edad?

Se observa que la media y la mediana de edad son mayores para personas con un credit score bueno (media de 36.50) en comparación con un credit score malo (media de 32.68). Además, se observa que la mitad de los datos para Good se encuentra en un rango de edad mayor que el resto. Sin embargo, la diferencia no es significativa.



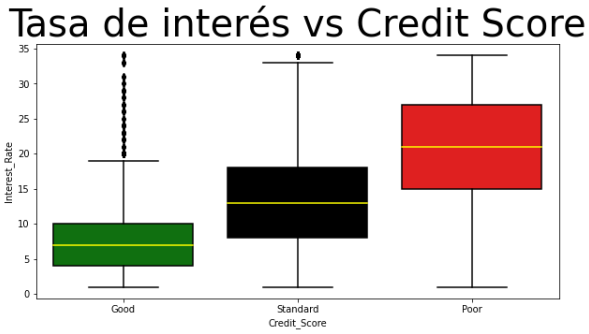
2. ¿Cómo es el credit score en términos del ingreso anual?

Se observa que aquellas personas con salarios más altos suelen tener menos riesgo de impago. Además, se observa una asimetría creciente hacia Good. Nuevamente, la diferencia no es significativa.

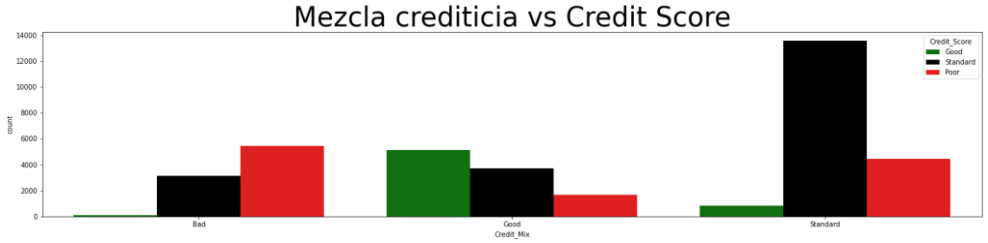


3. ¿Cómo afecta la tasa de interés al riesgo de impago?

Se puede observar una diferencia importante entre las tasas de interés para los tres grupos. A medida que la tasa de interés aumenta, el riesgo de incumplimiento de pago también aumenta.



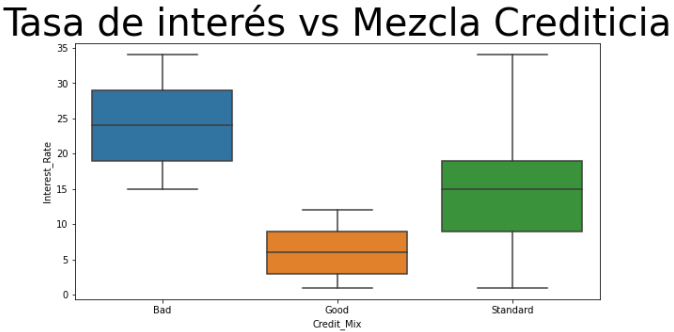
4. ¿Cómo se relaciona la mezcla crediticia con el credit score?



Se observa que una mala mezcla crediticia implica un alto riesgo de impago (credit score poor alto), es decir, la cantidad de personas que pagan el crédito es casi nula. Esto es coherente dado que una mala mezcla crediticia indica que la persona presenta varias deudas o préstamos pendientes, lo cual dificulta obtener y pagar un nuevo crédito.

Por otro lado, una buena mezcla crediticia implica una mayor probabilidad de pagar el crédito.

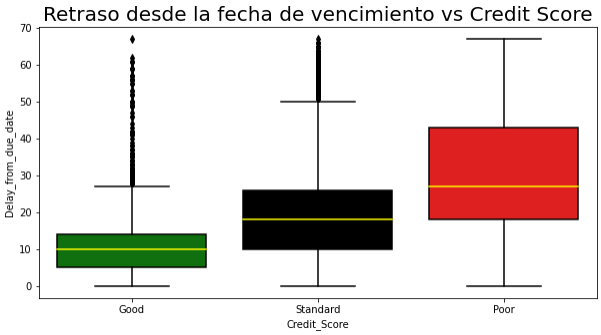
5. ¿Cómo se relaciona la tasa de interés con la mezcla crediticia?



Se observa que aquellas personas con buena mezcla crediticia acceden a créditos con tasas de interés menores que las personas con mala mezcla crediticia. La diferencia de la tasa de interés es significativa al comparar Bad con Good, mientras que una mezcla crediticia Standard presenta tasas de intereses variadas.

6. ¿Cómo son los retrasos desde la fecha de vencimiento según el credit score?

Se observa que las personas con un credit score malo tienden a tener retrasos mayores desde la fecha de vencimiento. Por otro lado, los que tienen un credit score bueno tienden a tener retrasos menores desde la fecha de vencimiento, con excepción de algunos valores outliers (p.ej. una persona que siempre paga a tiempo y justo un mes no pudo pagar).



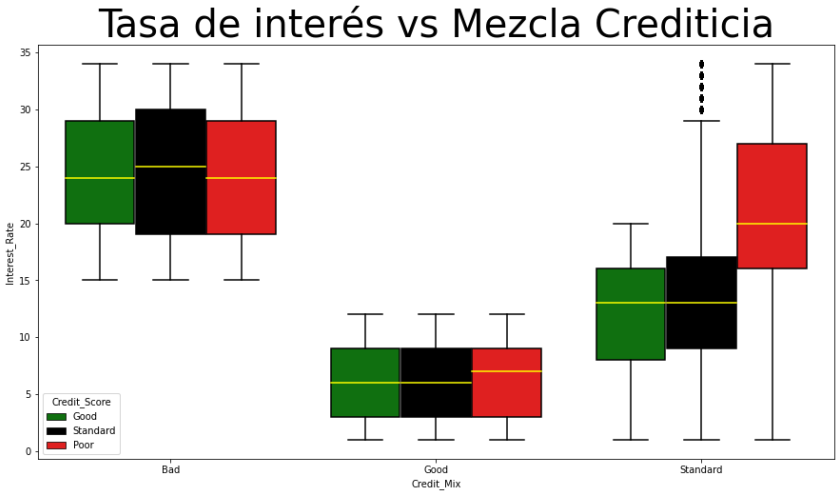
Análisis multivariado

Al realizar el mismo análisis, pero discriminando con la variable respuesta (“Credit Score”), se puede describir con mayor detalle el comportamiento de las variables y cómo estas afectan al riesgo crediticio.

Se puede observar en el pairplot (en notebook) que en algunos pares de variables existe una notable diferenciación entre “Good” y “Poor”. En cambio, los valores “Standard” se encuentran dispersos por todo el gráfico, es decir, mezclados con los otros dos valores mencionados.

Preguntas

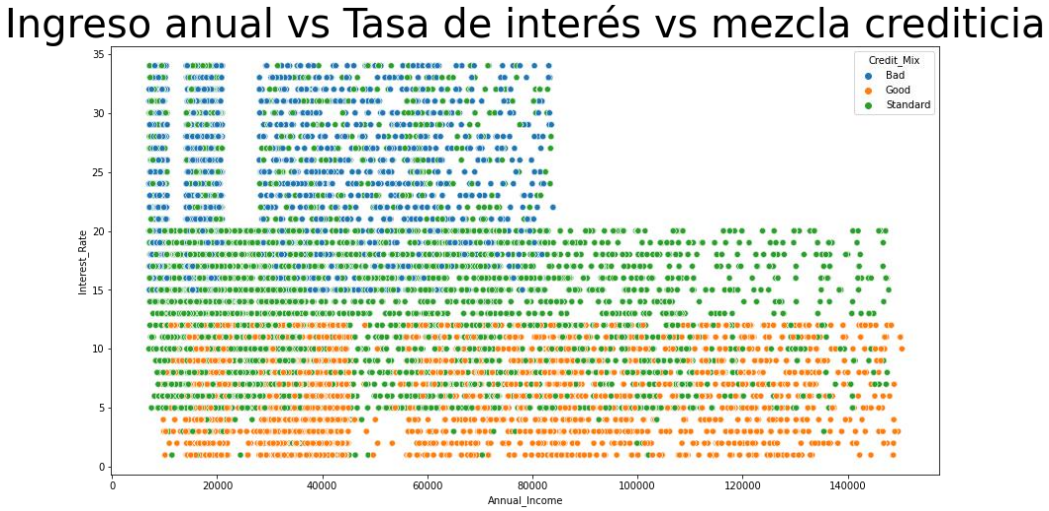
1. ¿Cómo es la relación entre la tasa de interés y la mezcla crediticia cuando se separa según el credit score?



Se observa que la relación entre la tasa de interés y la mezcla crediticia no varía según credit score para los valores Bad y Good. Para el valor Standard de mezcla crediticia sí se

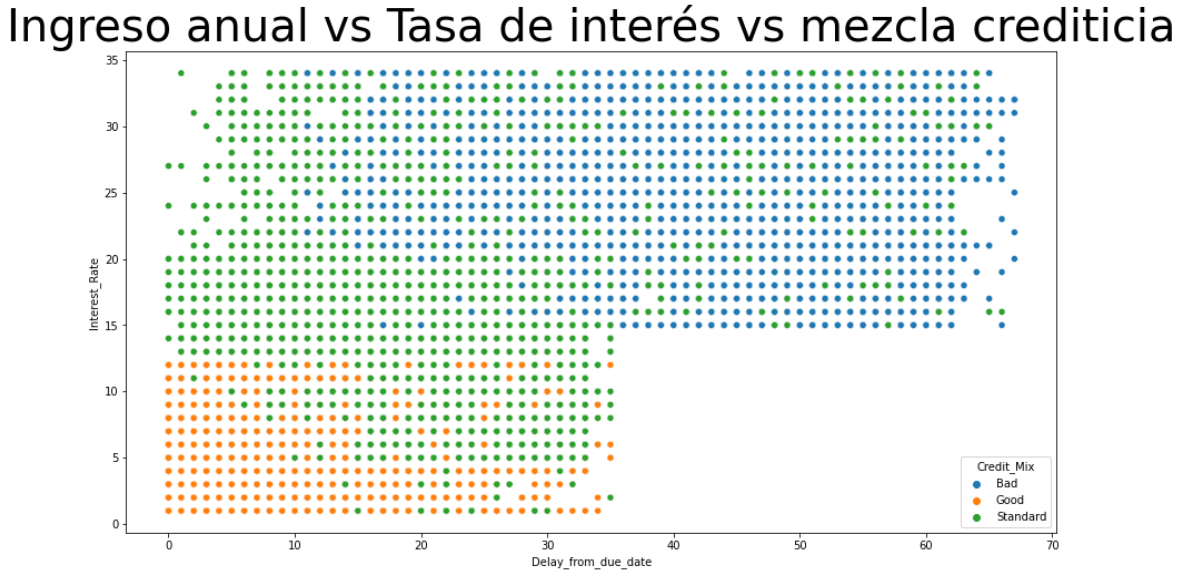
observa un cambio entre los valores de credit score. Los registros con un buen credit score se concentran en valores pequeños de tasa de interés cuando la mezcla crediticia es Standard. Para los valores Poor se encuentran distribuidos entre 0 y 35% de tasa de interés.

2. ¿Cómo es la relación entre el ingreso anual y la tasa de interés al discriminar por mezcla crediticia?



Se observa que las personas con ingresos anuales superiores a 85000 acceden a créditos con intereses menores al 20%. También se puede ver que todas las personas con un credit mix malo presentan salarios menores a 85000 y acceden a créditos con tasas de interés mayor a 15%. Por último, se destaca lo bien separados que están los grupos de Good y Bad de la mezcla crediticia.

3. ¿Cómo afecta la mezcla crediticia a la relación entre ingreso anual y tasa de interés?



Esta relación es similar a la analizada en la pregunta anterior.

Codificación

En esta sección, se procede a obtener las variables dummy de las variables categóricas. Esto se debe a que las variables categóricas con más de dos valores posibles no se pueden ingresar a un modelo, ya que se debe analizar la asociación de cada estado con la variable respuesta. De esta manera, cada variable categórica se divide en tantas columnas como valores posibles tenga. Cada columna se completa con 0 y 1, indicando con 0 cuando no presenta dicho valor y con 1 cuando sí presenta el valor (ver Figura 2).

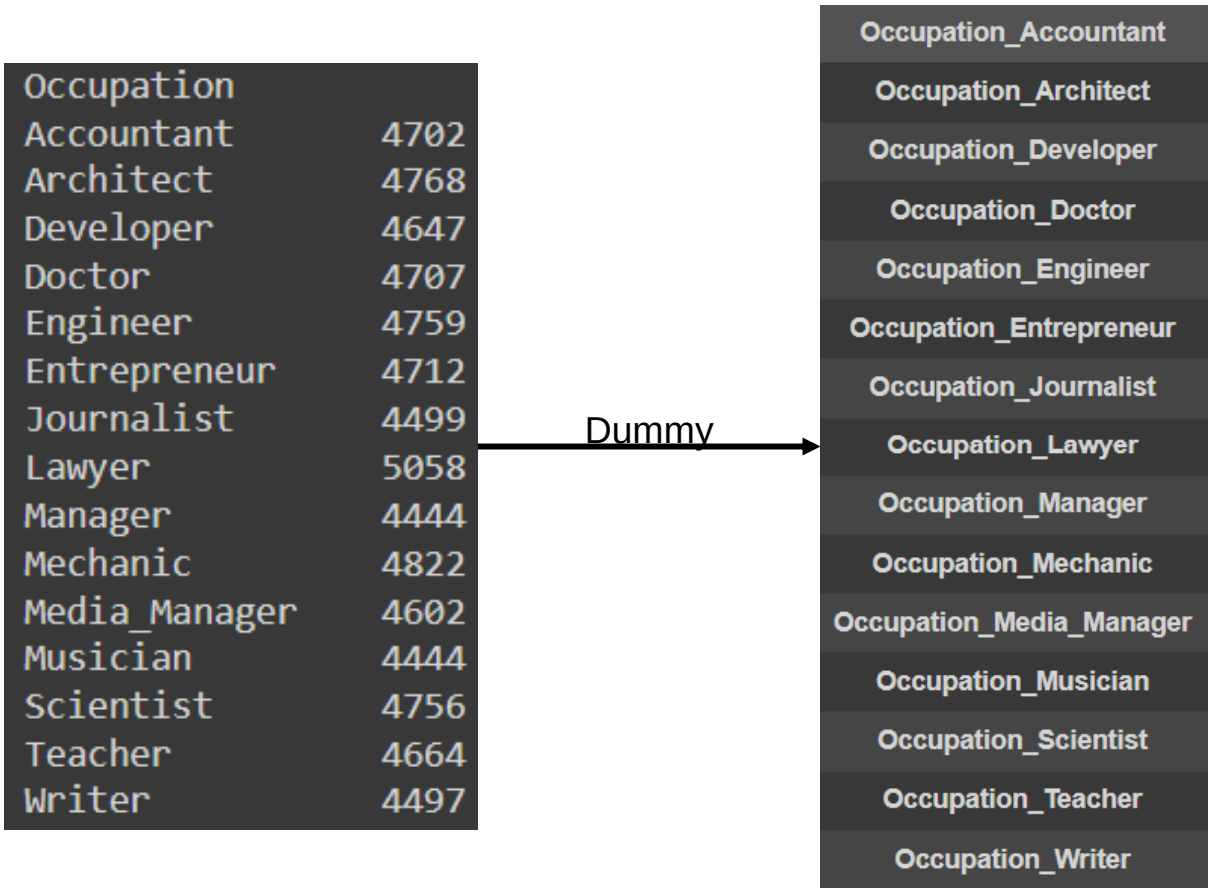


*Figura 2: a la izquierda, se observan los valores posibles de la variable **Occupation**. Al obtener las variables dummy, se obtienen una nueva variable por cada valor posible de la izquierda. Dentro de cada nueva variable, se completa con 0 y 1*

Por otro lado, se codificó la variable respuesta, **Credit_Score**, con 0 para el valor Poor, 1 para el valor Standard y 2 para el valor Good.

Análisis de componentes principales

Este análisis tiene como resultado disminuir la cantidad de variables que presenta el dataset. En primer lugar, se analizó la variable **Monthly_Inhand_Salary** en la sección Análisis Exploratorio de los Datos, dado que presentaba una cantidad importante de datos faltantes. Como esta variable hace referencia a los ingresos económicos de las personas, se realizó un estudio de correlación con la variable **Annual_Income**, que también hace referencia a los ingresos. Se encontró una correlación de 0.9978, lo cual indica que ambas variables comparten información relevante para el modelo y sería redundante tenerlas en el dataset. Por este motivo, se decide eliminar la variable **Monthly_Inhand_Salary**.

Para el resto de las variables numéricas, se realiza una matriz de correlación para observar si hay variables que presentan una asociación fuerte, es decir, comparten información relevante para el modelo (es redundante). En principio, ningún par de variables presentan una correlación alta como para poder simplificar variables.

Para completar este análisis, se realizó un Principal Component Analysis (PCA). Para esto, se utilizó todas las variables, incluidas las dummies. Se obtuvo que el primer componente explica casi la totalidad de la varianza del dataframe, específicamente el 0.9957 de la varianza (ver Figura 3).

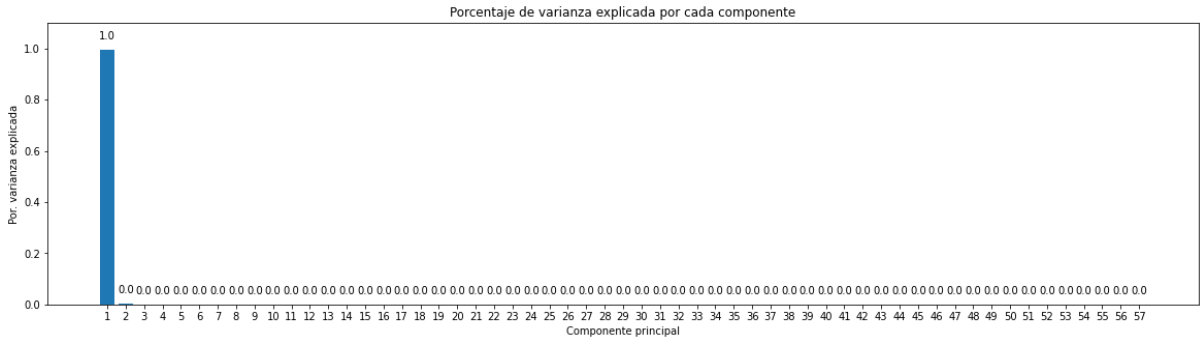


Figura 3: gráfico que muestra las componentes y la varianza explicada por cada uno. Se observa que ya el primer componente explica casi la totalidad de la varianza de los datos (0.9957)

Se realizó un modelo con la primera componente obtenida de PCA (ver sección Algoritmos de Clasificación).

Algoritmos de Clasificación

En esta instancia, se evalúan algunos algoritmos de clasificación con el fin de realizar una comparativa de desempeño y posteriormente elegir la mejor opción. Los algoritmos de clasificación evaluados en esta sección son los siguientes: K-Nearest Neighbor, Decision Tree, Random Forest y Logistic Regression.

Las métricas que se utilizan para evaluar los algoritmos de clasificación son las siguientes: Accuracy, Precision, F1-Score, Recall, Matriz de confusión.

A continuación, se detallan los parámetros utilizados para cada algoritmo y las métricas correspondientes para evaluar el desempeño del mismo. Hay que destacar que en todos los algoritmos se utilizó un Random State igual a 42.

K-Nearest Neighbor (KNN)

En este algoritmo, el parámetro más relevante es la cantidad de vecinos que se toman para la muestra. Para este caso, se adopta un k igual a 3.

Decision Tree

En este algoritmo, se utilizó una profundidad máxima de árbol igual a 7.

Random Forest

Para este algoritmo, se probaron diferentes parámetros, como max_features (cantidad de atributos a tener en cuenta en cada división del árbol) y cantidad de árboles. Para los cuatro modelos que se realizaron se ponderó las clases dado que el dataset presenta un desbalance.

- *Modelo 1:* En este caso, se probó un max_features de tipo logarítmico y una cantidad de árboles igual a 100.
- *Modelo 2:* En este caso, se probó un max_features de tipo logarítmico y una cantidad de árboles igual a 200.
- *Modelo 3:* En este caso, se probó un max_features de tipo raíz cuadrada y una cantidad de árboles igual a 200.
- *Modelo 4:* En este caso, se probó un max_features de tipo logarítmico y una cantidad de árboles igual a 100. Además, se colocan un mínimo de 1000 muestras para ser una hoja del árbol.

Random Forest con PCA

En esta ocasión, se procederá a realizar el mismo modelo random forest utilizando los mismos parámetros que en el modelo 3 de random forest, por el hecho de que es el mejor resultado obtenido, con la diferencia de que el entrenamiento previo se hará con la mejor

componente que arrojó el análisis de componentes. Hay que tener en cuenta que, como tenemos una sola componente de gran peso, se ha realizado el modelo con una variable.

Support Vector Machine (SVM)

Para este algoritmo, se procede a entrenar un modelo con todas las variables del dataset, utilizando un valor C de 100 y el kernel Radial Basis Function (RBF).

Con respecto al kernel lineal, se realizó un modelo con el mismo, pero se tuvo que parar debido al alto coste computacional. Además, en otras pruebas que se hicieron, las métricas dieron entre 0.5 y 0.6 de Accuracy y Precision.

Logistic Regression Multiclase

En este modelo, se utiliza el esquema one-vs-rest (OvR) para tratar la variable respuesta multiclase. Este esquema ajusta un problema binario para cada clase. Además, se utiliza el algoritmo de optimización “Newton-cg” que utiliza una función cuadrática.

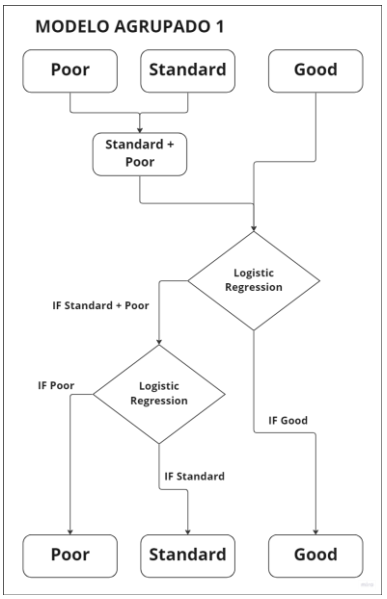
Resultados de modelos multiclase

Tabla de resultados obtenidos en los distintos modelos multiclase:

Algoritmo	Modelo	Accuracy	Precision	Recall	F1 Score
KNN	1	0.65500	0.64927	0.65500	0.65052
Decision Tree	1	0.71903	0.72461	0.71903	0.72076
Random Forest	1	0.77580	0.77524	0.77580	0.77544
	2	0.77921	0.77884	0.77921	0.77898
	3	0.78486	0.78457	0.78486	0.78469
	4	0.68729	0.74226	0.68729	0.69251
	PCA	0.73069	0.73056	0.73069	0.73063
SVM	Kernel: RBF	0.58649	0.49568	0.58649	0.52586
Logistic Regression	Multiclase	0.68003	0.72124	0.68003	0.68551

De acuerdo a las métricas obtenidas, se observa que el modelo multiclase con mejores resultados es el modelo 3 de Random Forest, con métricas cercanas a 85%, medida propuesta en objetivo. El modelo de PCA realizado no supera al anterior, por lo que se descarta la utilización de componentes principales.

Pipelines

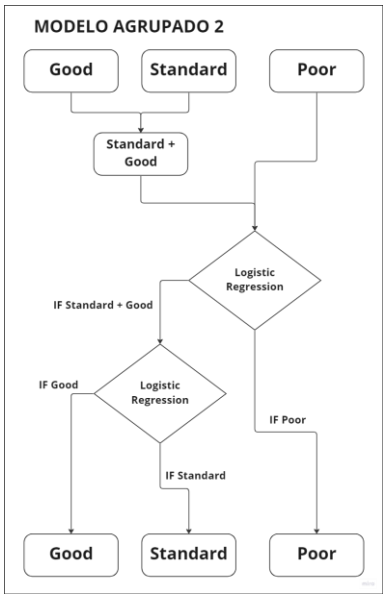


Con el fin de buscar optimizar el modelo de regresión logística, se decide realizar pruebas de agrupamiento de la variable target. Esto se debe a que la variable target contiene tres clasificaciones, y que entre la clasificación del tipo good y poor se distinguen bien en los gráficos de dispersión. Sin embargo, la clase standard se encuentra dispersa entre las dos anteriores.

Para ello, se procederá a realizar dos modelos agrupados: uno agrupará la clasificación poor con standard para comparar con la clase good y el otro modelo agrupará standard con good y se comparará con la clase poor. Luego, para ambos modelos se procede a realizar un filtro de la clase agrupada y realizar una nueva regresión con las mismas. Hay que destacar que para todos los modelos de regresión lineal de los pipelines se mantuvo el solver “newton-cg” como algoritmo de minimización.

Modelo Agrupado 1

Tal como se mencionó anteriormente, la variable target tiene tres posibles valores, por lo que se decide hacer una agrupación de valores y dos modelos distintos, tal como se muestra en el esquema **Modelo Agrupado 1**.



Modelo Agrupado 2

Para el segundo modelo, la agrupación se realizó diferente, pero se mantuvieron los mismos parámetros. El flujograma de este modelo se observa en el esquema **Modelo Agrupado 2**.

Resultados Obtenidos

En la siguiente tabla se observan los resultados obtenidos en los distintos modelos de ambos pipelines:

Algoritmo	Modelo	Accuracy	Precision	Recall	F1 Score
Pipeline 1	1	0.83309	0.89018	0.83305	0.84963
	2	0.78600	0.79817	0.78600	0.79011
Pipeline 2	1	0.78600	0.79817	0.78600	0.79011

	2	0.67848	0.67788	0.67848	0.67600

Si bien el pipeline 1 presenta buenas métricas en ambos modelos de regresión, esta arquitectura es más compleja que los otros algoritmos. Esto quiere decir que requiere un proceso de investigación más profundo y un sistema de métricas diferente al utilizado para el resto de los algoritmos.

Por otro lado, el algoritmo de regresión logística se suele utilizar para problemas de dos clases, por lo que sería conveniente analizar en profundidad otros algoritmos que sí se utilicen para problemas multiclase.

Dado que los modelos realizados con Random Forest obtuvieron buenos resultados, y tienen posibilidad de mejorarse, se decide descartar estas arquitecturas.

Selección de modelo

Dado los resultados obtenidos en la sección anterior, se decidió elegir el algoritmo Random Forest - modelo 3 para seguir trabajando en la obtención de mejores métricas. Además, se probará el algoritmo XGBClassifier, de la librería XGBoost, dado que es uno de los más utilizados para clasificación con datos estructurados.

Mejora del modelo - Random Forest

El proceso de mejora (ver Figura 4) consiste en varios pasos, desde análisis y selección de variables hasta un balance de muestras. Este último paso se debe a que, como se observó en la sección de EDA, la variable respuesta presenta un desbalance entre sus valores posibles, siendo el valor Standard el de mayor frecuencia.

Figura 4: esquema de los pasos a seguir para mejorar el modelo de clasificación

Evaluar variables numéricas

Se procede a entrenar el modelo con todas las variables numéricas y con los mismos parámetros que tenía el modelo 3 de Random forest de la sección anterior. Se obtuvieron las siguientes métricas al evaluar con los datos de testeo:

Algoritmo	Modelo	Accuracy	Precision	Recall	F1 Score
Random Forest	3	0.77213	0.77143	0.77213	0.76970

Selección de variables numéricas según relevancia

Se evalúa la importancia de las variables mediante el atributo “feature_importances_” del algoritmo de Random Forest de la librería scikit-learn. Un valor alto de este atributo indica una mayor importancia de la variable para el modelo de clasificación. Se observó que las variables **Outstanding Debt**, **Interest Rate** y **Delay From Due Date** son las 3 variables con mayor importancia (ver Figura 5). Además, se decidió agregar la variable **Annual Income** dado su importancia a la hora de pedir un préstamo.

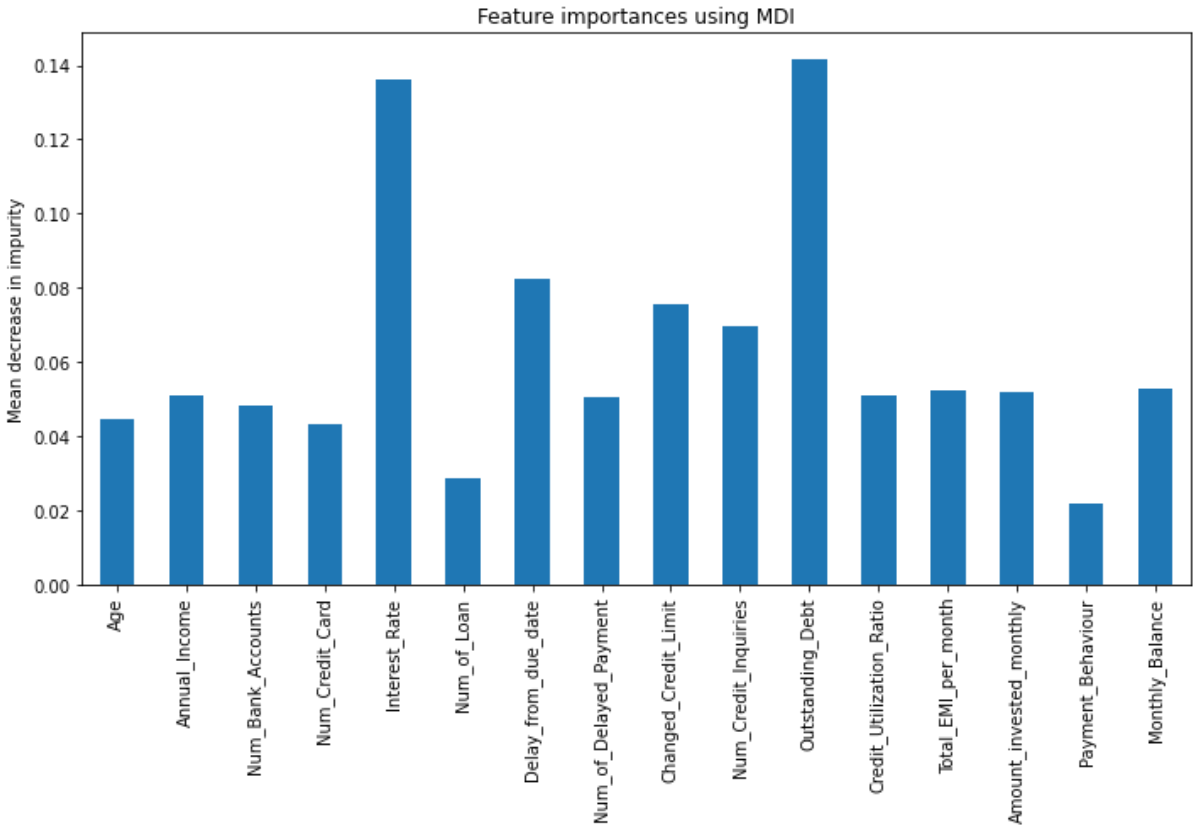


Figura 5: Gráfico de Importancia de variables numéricas

Al entrenar el modelo con estas cuatro variables numéricas seleccionadas, se obtuvieron las siguientes métricas con el conjunto de datos de testeo:

Algoritmo	Modelo	Accuracy	Precision	Recall	F1 Score
Random Forest	3	0.76365	0.76509	0.76365	0.76389

Si bien las métricas son muy similares a aquellas obtenidas al evaluar con todas las variables numéricas, se elige este modelo ya que presenta una menor cantidad de variables (modelo reducido).

Agregar variables categóricas

Al modelo de 4 variables se le agregaron todas las variables categóricas en formato dummy. Esto se hizo para ver si hay variables categóricas con gran relevancia para el modelo. Al evaluar este modelo con el conjunto de testeo, se obtuvieron las siguientes métricas:

Algoritmo	Modelo	Accuracy	Precision	Recall	F1 Score

Random Forest	3	0.77522	0.77737	0.77522	0.77598

Selección de variables categóricas según relevancia

Nuevamente, se evaluó las importancias de estas variables categóricas. Se observó que las variables asociadas a **Type of Loan**, **Credit Mix** y **Payment of Min Amount** son las variables categóricas con mayor importancia (ver Figura 6).

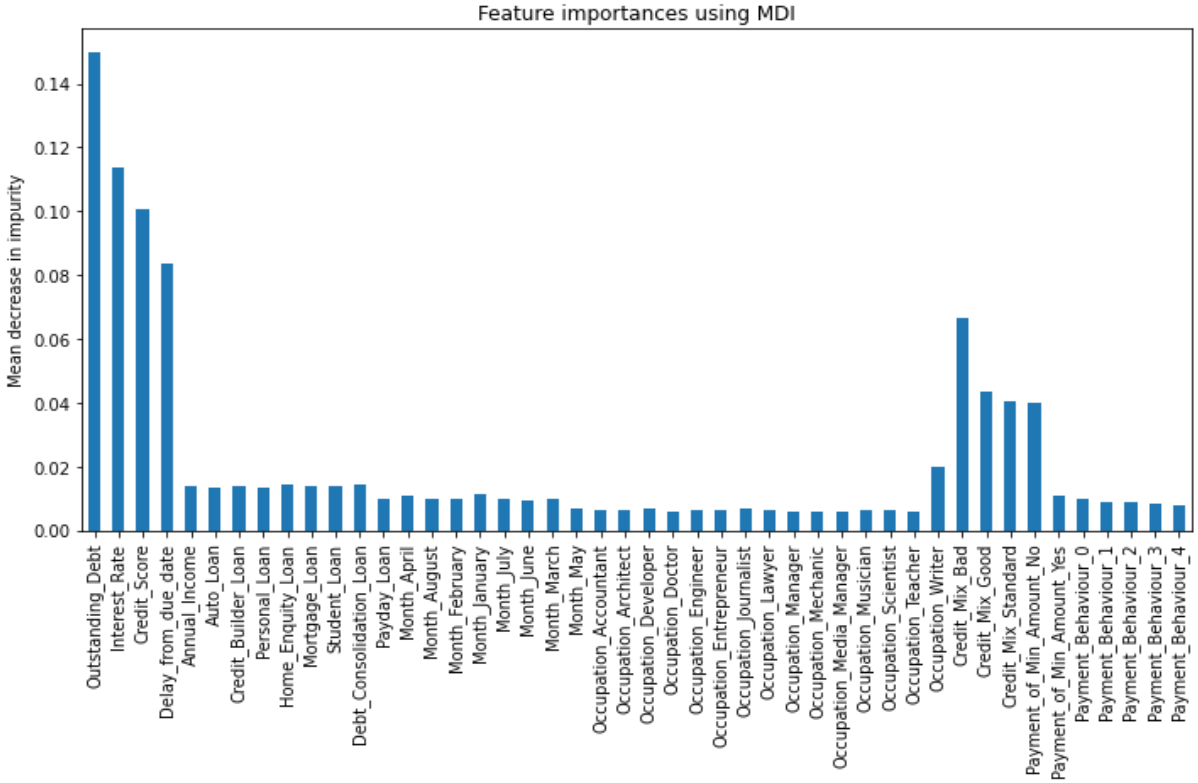


Figura 6: Gráfico de Importancia de variables categóricas

Al entrenar el modelo con las variables numéricas seleccionadas anteriormente y sumando las cuatro variables categóricas seleccionadas recientemente, se obtuvieron las siguientes métricas con el conjunto de datos de testeo:

Algoritmo	Modelo	Accuracy	Precision	Recall	F1 Score
Random Forest	3	0.77670	0.78065	0.77670	0.77721

Balanceo

Recordando que la variable respuesta presenta un desbalance, se procedió a balancear los datos mediante la función **SMOTE**, de la librería **Imbalanced Learn**. Con este balance y con las variables seleccionadas, se obtuvieron las siguientes métricas:

Algoritmo	Modelo	Accuracy	Precision	Recall	F1 Score
Random Forest	3	0.87543	0.87781	0.87543	0.87564

Grid Search

Una vez obtenido un modelo con buenos resultados, se procedió a realizar un ajuste de hiperparámetros. Para esto, se realizó un **Grid Search** para obtener la mejor combinación de parámetros del algoritmo de Random Forest. Los parámetros y valores analizados fueron los siguientes:

- n_estimators: 50, 100, 200, 300, 400, 500
- max_depth: 1, 3, 5, 10, 15, 25, 35, 50
- min_samples_split: 1, 3, 5, 7, 9
- max_features: sqrt, log2

Según los resultados obtenidos en el grid search, la mejor combinación de métricas es: max_depth: 30, max_features: sqrt, min_samples_split: 6, n_estimators: 200. Esta combinación obtuvo un Score de 0.85118.

Luego, se entrenó el modelo con las variables seleccionadas y los parámetros obtenidos del Grid Search. Se obtuvieron las siguientes métricas:

Algoritmo	Modelo	Accuracy	Precision	Recall	F1 Score
Random Forest	3	0.86965	0.87242	0.86965	0.86989

Como se observa en las medidas obtenidas, estas son levemente peores que las del modelo anterior con datos balanceados por lo que quedaría descartada esta combinación. Por lo tanto, se decide mantener los parámetros del paso anterior.

Matrices de confusión

En la Figura 7 se observa las matrices de confusión obtenidas del proceso de mejora. Hay que destacar que los valores están normalizados por fila, es decir, por los valores reales de la variable respuesta. Esto se debe a que es una mejor manera de visualizar y comparar valores entre celdas. La matriz del paso 5, que se encuentra enmarcada, corresponde al modelo que obtuvo mejores métricas.

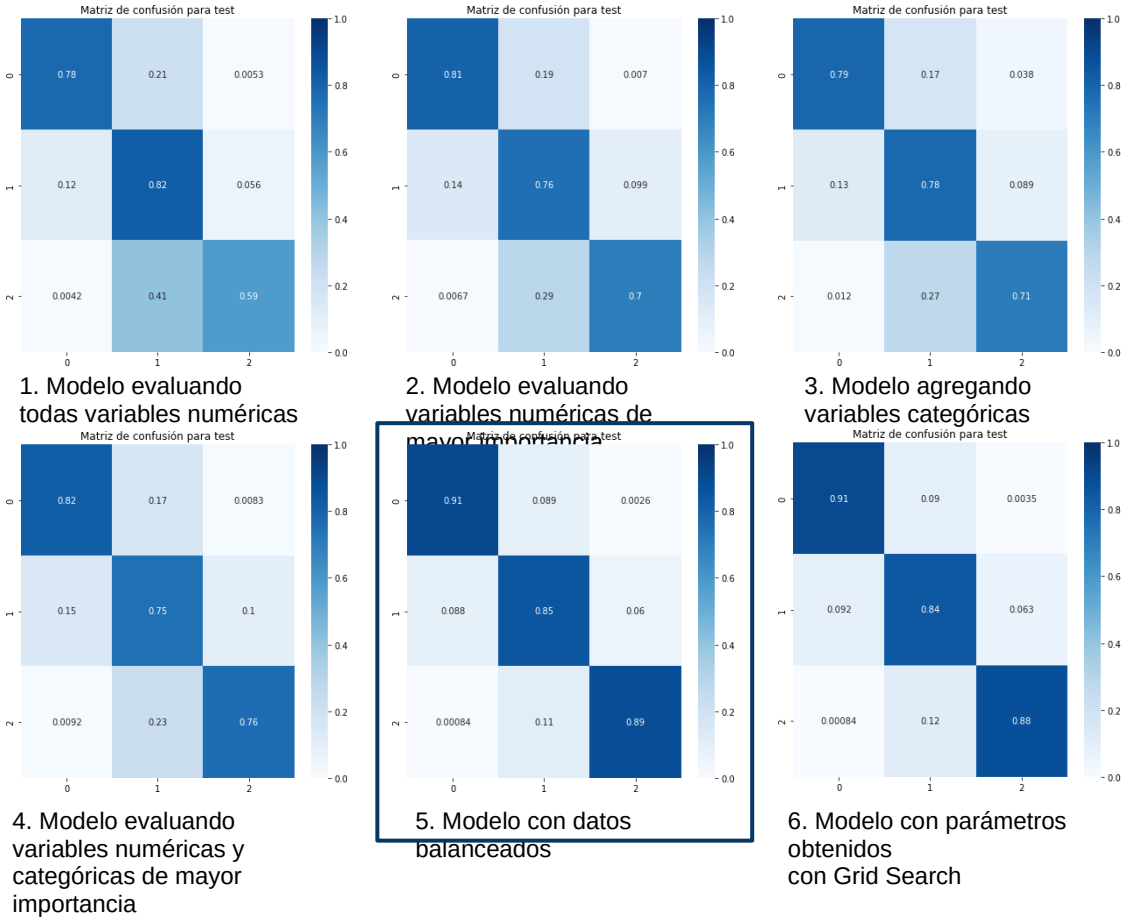


Figura 7: matrices de confusión de los pasos de mejoras para el algoritmo de Random Forest

Mejora del modelo - XGBClassifier

Teniendo en cuenta el proceso aplicado en la mejora del Random Forest, se decidió realizar los mismos pasos para XGBClassifier, con la diferencia de que se analizará la importancia de todas las variables al mismo tiempo, tanto numéricas como categóricas.

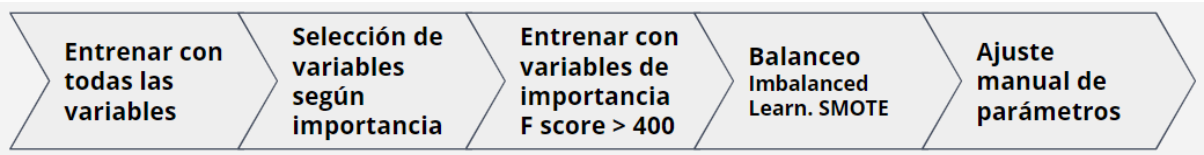


Figura 8: esquema de los pasos a seguir para mejorar el modelo de clasificación

Evaluar todas las variables

Se procede a entrenar el modelo con todas las variables, tanto numéricas como categóricas. Se obtuvieron las siguientes métricas al evaluar con los datos de testeo:

Algoritmo	Accuracy	Precision	Recall	F1 Score

XGBClassifier	0.76648	0.76671	0.76648	0.76658

Selección de variables según relevancia

Se evalúa la importancia de las variables mediante el gráfico obtenido del método plot_importance del algoritmo XGBClassifier (ver Figura 9). Para la selección de las variables, se establece un umbral en 400 de F Score. Por ende, las variables seleccionadas, en orden de importancia, son:

1. Outstanding_Debt
2. Annual_Income
3. Total_EMI_per_month
4. Changed_Credit_Limit
5. Credit_Utilization_Ratio
6. Amount_invested_monthly
7. Monthly_Balance
8. Delay_from_due_date
9. Age
10. Interest_Rate
11. Num_of_Delayed_Payment
12. Num_Credit_Inquiries
13. Num_Bank_Accounts

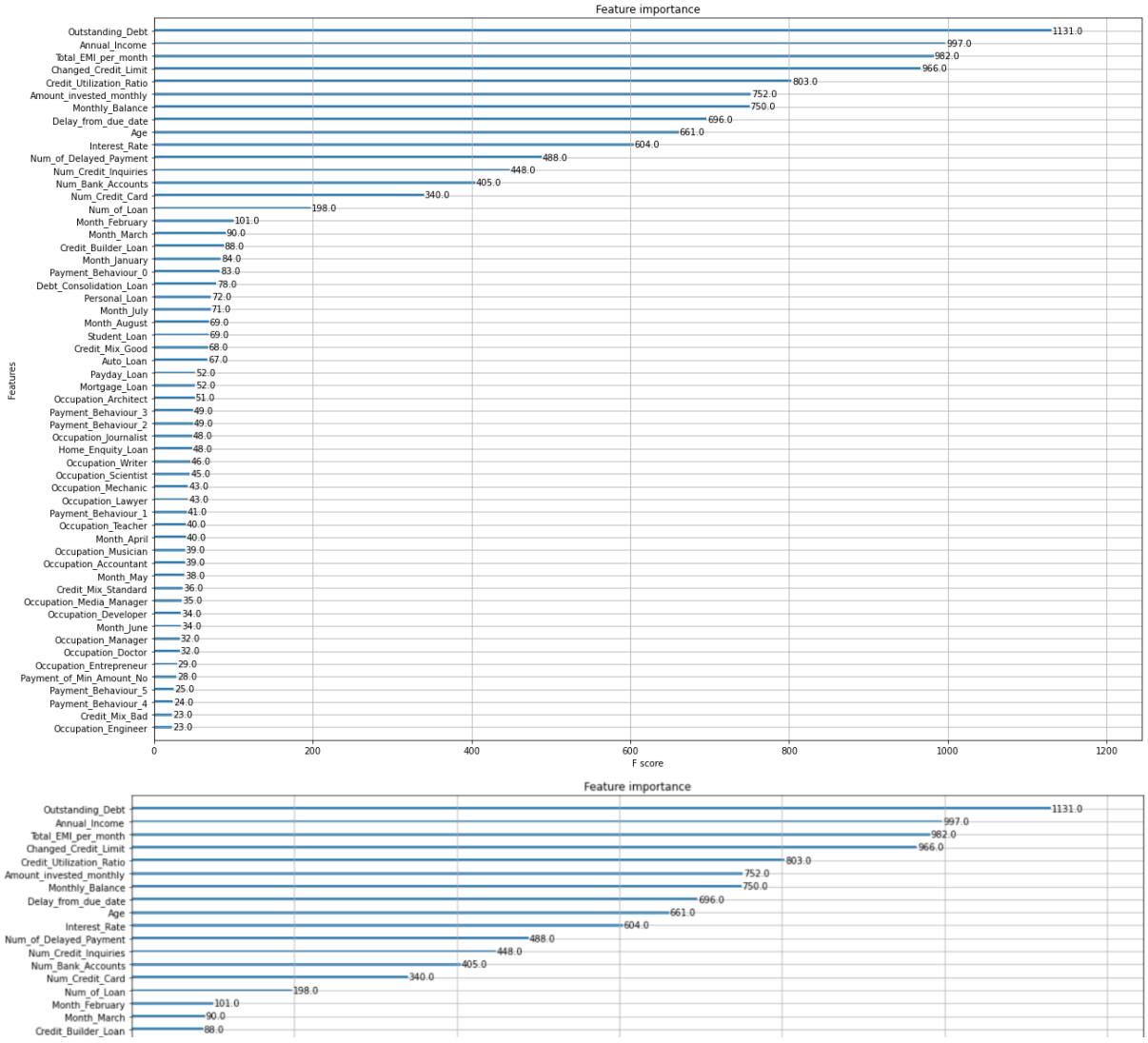


Figura 9: En la parte superior, se observa el gráfico de importancia de variables para el modelo de XGBClassifier. En la parte inferior se hace un zoom en las variables más importantes.

Al entrenar el modelo con las variables seleccionadas, se obtuvieron las siguientes métricas con el conjunto de datos de testeo:

Algoritmo	Accuracy	Precision	Recall	F1 Score
XGBClassifier	0.74078	0.73884	0.74078	0.73895

Balanceo

Finalmente, recordando que la variable respuesta presenta un desbalance, se procedió a balancear los datos mediante la función **SMOTE**, de la librería **Imbalanced Learn**. Con este balance y con las variables seleccionadas, se obtuvieron las siguientes métricas:

Algoritmo	Accuracy	Precision	Recall	F1 Score
XGBClassifier	0.79903	0.80329	0.79903	0.80010

Ajuste de hiperparámetros

Una vez obtenido un modelo con buenos resultados, se procedió a realizar un ajuste de hiperparámetros. Para esto, primero se intentó realizar un **Grid Search** para obtener la mejor combinación de parámetros. Sin embargo, el coste computacional era tal que las corridas duraban más de 24 horas. Por este motivo, se decidió realizar un ajuste manual de los parámetros hasta obtener las mejores métricas. Los parámetros con los que se obtuvieron buenas métricas fueron los siguientes:

- n_estimators: 1000
- max_depth: 10
- learning_rate: 0.3
- max_leaves: 100

Según el análisis realizado, con estos parámetros se entrenó el modelo del paso anterior y se obtuvieron las siguientes métricas:

Algoritmo	Accuracy	Precision	Recall	F1 Score
XGBClassifier	0.93489	0.93508	0.93489	0.93493

Se observa que las medidas obtenidas son mejores que las del modelo anterior con datos balanceados. Por lo tanto, se decide mantener estos parámetros para el modelo. Además, se observa que es el mejor modelo obtenido en el trabajo.

Matrices de confusión

En la Figura 10 se observa las matrices de confusión obtenidas del proceso de mejora. Nuevamente, los valores están normalizados por fila para mejorar la visualización y comparación. La matriz del paso 4, que se encuentra enmarcada, corresponde al modelo que obtuvo mejores métricas.

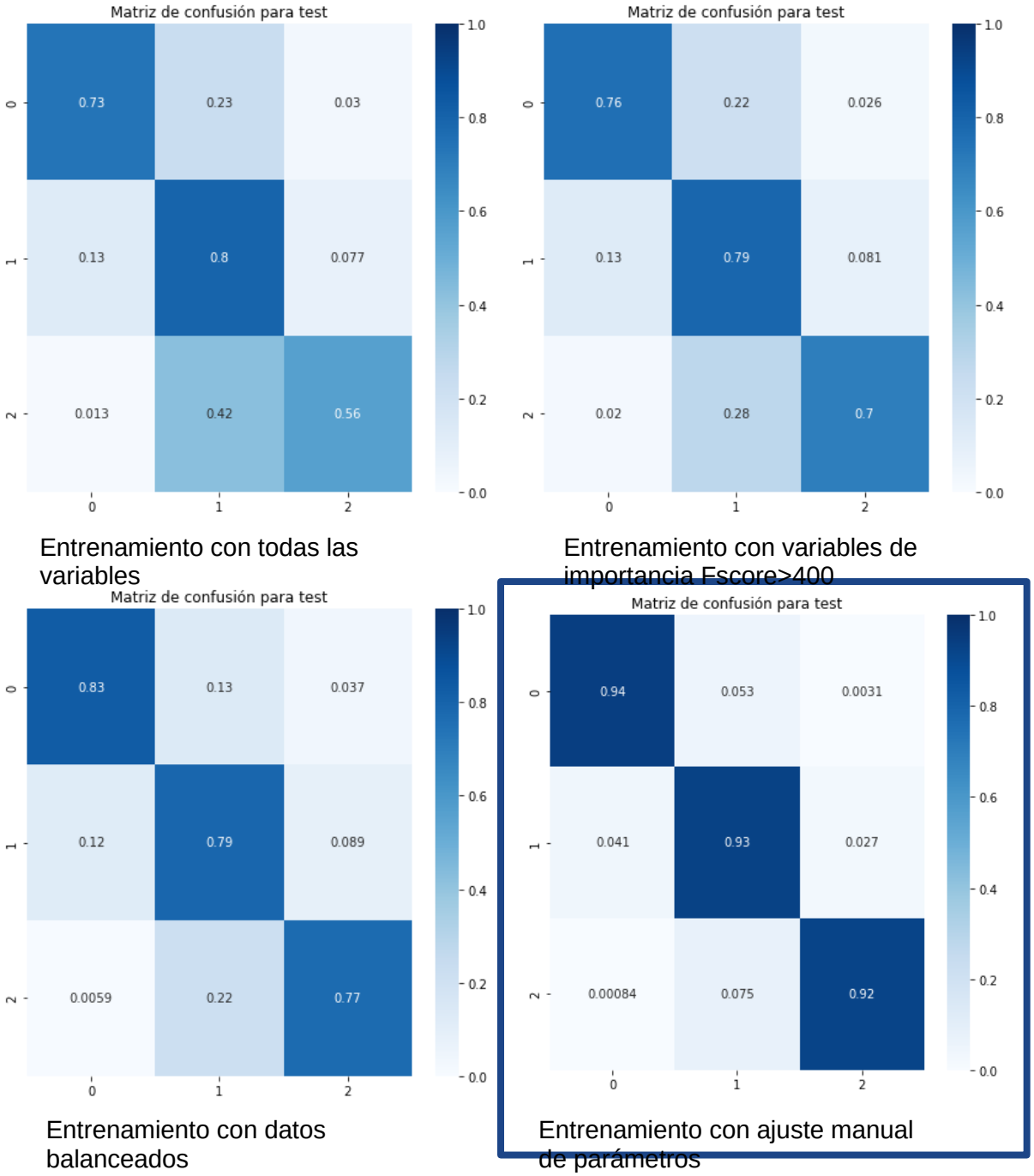


Figura 10: matrices de confusión de los pasos de mejoras para el algoritmo XGBClassifier

Futuras líneas

En primer lugar, se podría hablar con personal de la empresa financiera para implementar mejoras en el sistema de entrada de datos para reducir la cantidad de datos faltantes o valores incorrectos. Una vez realizado este plan de mejora y obtenido una buena cantidad de datos nuevos, habría que realizar un re-entrenamiento del modelo. Al contar con mayor cantidad de datos, es probable que se obtengan dataset de mejor calidad y, por ende, se obtengan mejores resultados.

En cuanto a los pipelines realizados, si bien se descartaron por presentar arquitecturas más complejas que los demás modelos, se podría investigar más sobre estos. Esto se debe a que el primer modelo realizado obtuvo las mejores métricas iniciales (cercano al 0.8 de accuracy al promediar ambos modelos del pipeline 1).

Con respecto al modelo final obtenido (de XGBClassifier), se podría ajustar aún más los hiperparámetros. La razón por la que no se probó algoritmos como Grid Search o Random Grid Search es que tardan mucho en correr para este caso. Por lo que se necesitaría más tiempo para investigar y realizar estos análisis.

Por último, se podría probar algoritmos de ML no supervisados, tales como K-means o HDBSCAN, y ver si la cantidad de cluster de la variable respuesta es efectivamente 3. Además, se debería buscar relaciones de esos clusters con el fin de identificar a qué clase target pertenece.

Conclusión

Al realizar el EDA y el data Wrangling, se observa que el perfil de una personas con alto riesgo de impago corresponde a una persona joven (media de edad de 32.68), con bajos ingresos anuales (inferior a 55000), con tasas de interés superiores al 15%, con una mezcla crediticia mala y con retrasos desde la fecha de vencimiento de pago que tienden a ser mayores a 20 días.

Teniendo en cuenta el perfil de las personas en relación con el riesgo de impago, se realizó un estudio de modelos de clasificación en busca de clasificar los perfiles. Para la selección definitiva del modelo se tuvo en cuenta el coste computacional y el valor de las métricas obtenidas. En base a esto, se observó que el modelo que se destaca es el obtenido en el cuarto paso del proceso de mejora del algoritmo XGBClassifier. Este modelo demora 3 minutos y 30 segundos en entrenarse (depende de la capacidad de la computadora) y obtuvo un accuracy de 0.93489, el cual cumple con el objetivo propuesto para el trabajo (accuracy mayor a 0.85). Por lo tanto, es posible predecir si una persona tiene alto riesgo de impago mediante un modelo de clasificación.

Para la obtención de este modelo, se realizó un análisis de la importancia, en donde se obtuvo que las variables más importantes para el modelo de clasificación son (en orden de mayor a menor importancia): *Outstanding_Debt, Annual_Income, Total_EMI_per_month, Changed_Credit_Limit, Credit_Utilization_Ratio, Amount_invested_monthly, Monthly_Balance, Delay_from_due_date, Age, Interest_Rate, Num_of_Delayed_Payment, Num_Credit_Inquiries y Num_Bank_Accounts*. Estas variables presentaron un valor superior a 400 de F Score. El resto de las variables resultaron no importantes para el modelo. Además, se realizó un balance de datos debido a la desproporción en la cantidad de registros de cada clase de la variable respuesta.

Con respecto al modelo obtenido del proceso de mejora de Random Forest, este modelo también cumple con el objetivo del modelo y tiene un tiempo de entrenamiento de 33.5 segundos. Sin embargo, el accuracy dio como resultado 0.87543. Por ende, como se observa que el accuracy es inferior al resultado obtenido con el algoritmo XGBClassifier y como los tiempos de ejecución no son grandes, se decide optar por el último algoritmo expuesto.