



Proyecto integrador – primer semestre:

## **Modelo de predicción de Morosidad**

**Santiago Bedoya Rios**

[sbedoyar2@eafit.edu.co](mailto:sbedoyar2@eafit.edu.co)

**Juliana Andrea Peralta Jaramillo**

[japeraltaj@eafit.edu.co](mailto:japeraltaj@eafit.edu.co)

**Maestría en Ciencias de los Datos y Analítica**

**Universidad EAFIT**

9 de junio de 2022

# Contenido.

1. Introducción	6
2. Desarrollo metodológico	7
2.1. Pregunta de Negocio:	7
2.2. Análisis Exploratorio de Datos:	7
2.2.1. Entendimiento de los datos:	8
2.2.1.1. Atributos de calidad:	8
2.2.1.2. Distribución de valores:	9
2.2.2. Preparación de los datos	11
2.2.2.1. Tratamiento de valores faltantes:	12
2.2.2.2. Atributos desbalanceados:	16
2.2.2.3. Correlación entre atributos:	17
2.2.2.4. Atributos representativos de valores agregados:	17
2.2.2.5. Balanceo de atributos categóricos:	18
2.2.2.6. Representación de variables categóricas a variables numéricas:	19
2.2.2.7 Imputación de valores faltantes:	20
2.3. Modelado de los datos:	21
2.3.1. Escalado de los datos:	21
2.3.2. Reducción de la dependencia lineal entre atributos:	22
2.3.3. Detección de Outliers:	23
2.3.3.1. Mejorando el condicionamiento de la matriz de covarianzas:	23
2.3.4. Reducción de dimensionalidad:	24
2.3.5. Modelo de aprendizaje supervisado:	25
2.3.5.1. Métricas de evaluación de los modelos:	25
2.3.5.2. Modelos clasificadores:	25
2.3.5.3. Comparación de los modelos, por tipo:	31
3. Infraestructura e Ingeniería de Datos	33
3.1. Definición de arquitectura:	33
3.2. Lineamientos del proceso ELT:	33

3.2.1. Proceso de ingestión:	34
3.2.2. Almacenamiento de datos:	34
3.2.3. Transformación de datos:	34
3.2.4. Entorno de procesamiento:	35
3.2.5. Despliegue del modelo:	36
4. Conclusiones y consideraciones:	38
5. Referencias	39

## **Lista de Ilustraciones.**

Ilustración 1. Análisis y distribución de la variable objetivo 'TARGET'.	9
Ilustración 2. Ejemplo del reporte generado para una variable numérica.	10
Ilustración 3. Ejemplo del reporte generado para una variable categórica.	10
Ilustración 4. Gráficas generadas para comparar la distribución de algunas variables numérica significativas.	11
Ilustración 5. Cantidad de valores faltantes por variable. Atributos con más del 10% de valores faltantes.	12
Ilustración 6. Distribución de los atributos FLAG_OWN_CAR y OWN_CAR_AGE.	13
Ilustración 7. Distribución del atributo EXT_SOURCE_3 en función del TARGET.	14
Ilustración 8. Frecuencia del atributo OCCUPATION_TYPE.	14
Ilustración 9. Distribución de las proporciones del TARGET en función de OCUUPATION_TYPE.	15
Ilustración 10. Distribución de AMT_INCOME_TOTAL, en función de OCCUPATION_TYPE	15
Ilustración 11. Diagramas boxplot para AMT_INCOME_TOTAL	16
Ilustración 12. Mapa de calor con la correlación entre atributos relevantes.	17
Ilustración 13. Fragmento del resumen de la distribución de los datos una vez escalados.	21
Ilustración 14. Proceso iterativo para reducir la dependencia lineal en el data set.	22
Ilustración 15. Variabilidad explicada acumulada en función del número de componentes principales.	24
Ilustración 16 Evaluación árbol de decisiones, profundidad definida, todo el dataset.	26
Ilustración 17 Evaluación árbol de decisiones, profundidad definida, selección aleatoria para balancear.	26
Ilustración 18 Recall score obtenido en función de la profundidad.	27
Ilustración 19 Precision score obtenido en función de la profundidad	28
Ilustración 20 Recall score obtenido cuando se varía la selección de datos con TARGET igual a cero.	28
Ilustración 21 Precision score obtenido cuando se varía la selección de datos con TARGET igual a cero.	29
Ilustración 22 Resultados obtenidos con el modelo de regresión logística usando un dataset balanceado como entrenamiento.	29
Ilustración 23 Resultados obtenidos con el modelo de bosque aleatorio usando un dataset balanceado.	30
Ilustración 24 Resultados del modelo bosque aleatorios luego de optimizar hiperparámetros.	31
Ilustración 25 Comparación de las áreas bajo la curva para los modelos analizados.	31
Ilustración 26. Diagrama de Arquitectura.	33

Ilustración 27. Imagen de referencia, del data lake hospedado en AWS S3.	35
Ilustración 28. Imagen de referencia, entorno de procesamiento Google Colab.	36
Ilustración 29 Imagen de la aplicación desplegada para el usuario final.	36
Ilustración 30 Imagen de referencia del proceso de despliegue e implementación del reverse proxy.	
	37

---

## **Lista de Tablas.**

Tabla 1. Ficha general del dataset.	8
Tabla 2. Características de calidad del dataset en cuanto a su composición.	8

## 1. Introducción

La morosidad hace alusión al incumplimiento del pago de una obligación adquirida, y en función del tipo de mercado y el producto per se, esta es evaluada con distintos niveles de severidad, pues si el producto contempla prendas en garantía, respaldo asegurado obligatorio, estímulos de pago oportuno, o penalizaciones por incumplimiento según las condiciones acordadas. En todos los casos, es un comportamiento que ninguna entidad prestadora de servicios asociados desea tener presente en los hábitos de pago de su oferta a los usuarios finales. Ahora bien, determinar a priori si un cliente será o no moroso, tiene un punto de referencia fundamentado en los hábitos de pago y experiencia crediticia, todo consolidado en una métrica conocida como el score crediticio. Sin embargo, y al estar sólo centrado en un histórico de crédito, no contempla atributos adicionales propios de los clientes, relacionados directamente con características demográficas, sociales y laborales, las cuales, si bien no tienen relación directa con un producto financiero, si facilitan la caracterización y entendimiento del cliente, y así conocer en detalle la persona a la que se le están brindando estos productos y, por ende, tomar acciones que permitan ofrecer los servicios reduciendo el impacto negativo por la materialización de riesgos de morosidad.

En Colombia, para el tercer trimestre de 2021, la morosidad de 60 o más días, la más alta en la escala utilizada por el mercado financiero local, aumentó a 5,1% en todos los productos crediticios, menos en libranza. El hecho de que esto haya tenido un incremento, puede haber generado malestar en la oferta bancaria, pues el capital en riesgo es mayor y se deben ejecutar procesos de cobro de mayor envergadura, lo cual demanda un esfuerzo adicional a las operaciones normales.

Con esto en mente, y ante la oportunidad presente en kaggle, con un reto para la detección de la morosidad con datos capturados en India, se presenta el desarrollo de una solución para determinar si un cliente puede o no ser moroso, en función de los atributos propios del cliente, más allá de su historial crediticio. Se cuenta con un dataset puesto a disposición que contiene los datos de los clientes y cómo han sido clasificados dada su responsabilidad con los productos crediticios adquiridos. Al ser un dataset de alta dimensionalidad y con atributos de diversa naturaleza, la preparación de los datos deberá ser muy concienzuda y sustentada de tal manera que se logre reducir el sesgo propio de los datos, sin alterar la variabilidad explicada originalmente por los registros capturados. Finalmente, se analizarán los modelos clasificadores que briden un desempeño aceptable para realizar las predicciones y se desplegará toda la solución, de tal manera que sea fácilmente usable por un usuario final para que sea tomada como herramienta en las operaciones realizadas por una entidad real.

## **2. Desarrollo metodológico**

### **2.1. Pregunta de Negocio:**

En las entidades financieras cuando un deudor tiene atraso de pago por más de un día respecto a la fecha de vencimiento de la obligación adquirida, se le empieza a considerar como cliente moroso. Sin embargo, dependiendo del tipo de crédito y de las condiciones de este, cuando el deudor se empieza a atrasar en promedio por más de dos o tres meses, la entidad financiera puede considerar que su dinero está en riesgo de recaudo, no solo los meses adeudados, si no, la deuda total del crédito.

Se considera que el mayor porcentaje de empresas que tienen cartera vencida son las microempresas y en segundo lugar las pequeñas y medianas empresas debido a que estas no cuentan con una infraestructura o un personal adecuado para dar un seguimiento de cobranza por un largo periodo de tiempo. En el caso de las instituciones bancarias y grandes compañías suelen tercerizar estos procesos de cobranza que acarrean costos de operación más altos.

Es por esto por lo que se hace vital tener un mayor conocimiento del cliente y su comportamiento de pago antes de otorgar cualquier préstamo e incurrir en un riesgo financiero elevado para la compañía. Para mitigar este riesgo se requieren de mecanismos preventivos que permitan segmentar los clientes por probabilidad de recaudo y riesgo, comparar los hábitos de pago de un segmento o frente a otros segmentos.

Con esta solución se pretende brindar una herramienta que provean de estrategias oportunas y preventivas que permitan identificar los factores claves (variables) que promueven o caracterizan el incumplimiento en el pago del préstamo otorgado con el fin de clasificar posibles morosos antes de la otorgación del crédito y así lograr un impacto en la cartera y disminución del riesgo al que se exponen diariamente las entidades financieras.

### **2.2. Análisis Exploratorio de Datos:**

Los datos se recopilaron como parte del experimento social para proporcionar inferencias públicas sobre cómo una persona que solicita un préstamo puede completarlo en un tiempo mínimo. Además, adherirse a los hechos sobre qué tipo de clientes no pagan las cuotas o el préstamo

completo y proporcionar inferencia para que la persona que solicita el préstamo no entre en esa categoría.

### 2.2.1. Entendimiento de los datos:

El conjunto de datos utilizado para brindar solución al presente proyecto es generado a partir de los datos recolectados al registrar la información asociada a las solicitudes de crédito recibidas por la entidad, y que fueron clasificadas como de alto riesgo teniendo en cuenta si el cliente se atrasó más de X días en al menos una de las primeras Y cuotas del préstamo o para el resto de los casos.

*Tabla 1. Ficha general del dataset.*

Conjunto de datos: Application Data.csv	
Fuente:	<a href="#">Credit Card Fraud Detection   Kaggle</a>
Formato:	Comma Separated Value (.csv)
Cantidad de registros:	307.511
Cantidad de atributos:	122

#### 2.2.1.1. Atributos de calidad:

Inicialmente, para tener un panorama claro sobre la calidad de los datos en términos de completitud, validez e integridad, se identificaron características claves asociadas a la estructura del conjunto de datos:

*Tabla 2. Características de calidad del dataset en cuanto a su composición.*

Dataset statistics		Variable types	
Number of variables	122	Numeric	106
Number of observations	307511	Categorical	16
Missing cells	9152465		
Missing cells (%)	24.4%		
Total size in memory	286.2 MiB		
Average record size in memory	976.0 B		

De resaltar que se detectó que el 24,4% de los datos del conjunto corresponden a valores faltantes o campos vacíos. Esto es una característica que debe ser revisada con sumo detalle, pues cualquier modelamiento que se genere sin el debido tratamiento de esto, será complicado computacionalmente hablando y no muy funcional, dado el grado de incertidumbre presente en el conjunto de datos. En la siguiente sección se especifica cuáles fueron las acciones tomadas para analizar este atributo del conjunto de datos.

Para la solución a desarrollar, la variable objetivo corresponde al atributo ‘TARGET’ del dataset. Corresponde a un valor binario, tratado como booleano con el fin de identificar si un cliente fue moroso (1) o no (0) con su producto financiero.



*Ilustración 1. Análisis y distribución de la variable objetivo 'TARGET'.*

Como se observa, en la variable objetivo se presenta un desbalance en función de la distribución de frecuencias de cada clase, pues la clase 0 corresponde al 91.9% de los registros en el dataset, equivaliendo a 282.686 registros.

#### *2.2.1.2. Distribución de valores:*

Se realiza un análisis de distribución de la frecuencia y magnitud de los valores de las variables numéricas con el fin de determinar la coherencia e integridad de los datos, puesto que, a partir de esto, es posible establecer preguntas respecto al rango ‘saludable’ en el cuál una variable debe estar. Este primer análisis es desarrollado con la librería **pandas-profiling**, permitiendo observar la distribución de todas las variables, de una manera amigable y enriquecedora:

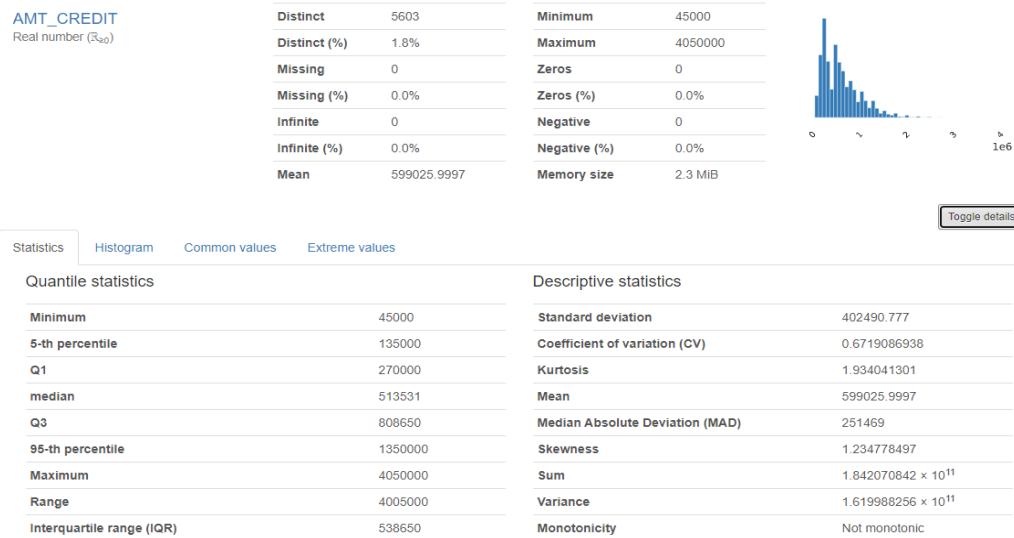


Ilustración 2. Ejemplo del reporte generado para una variable numérica.

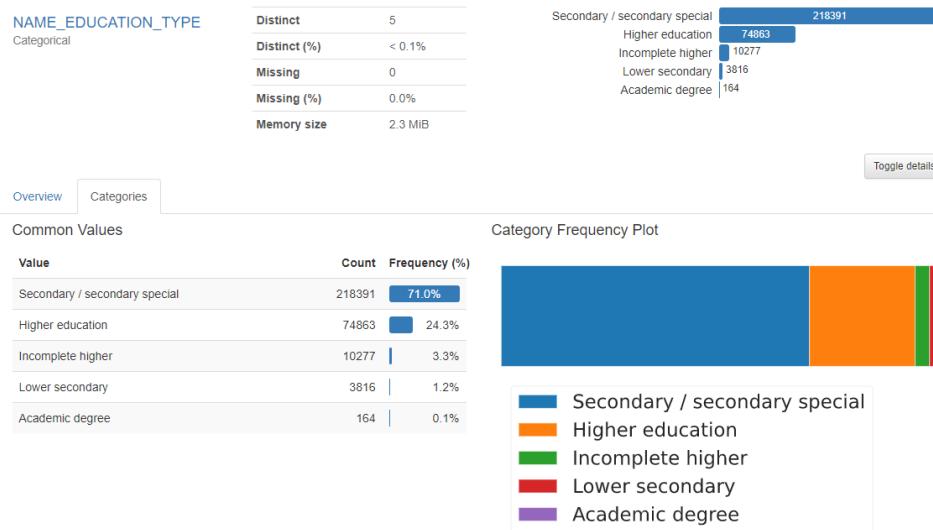
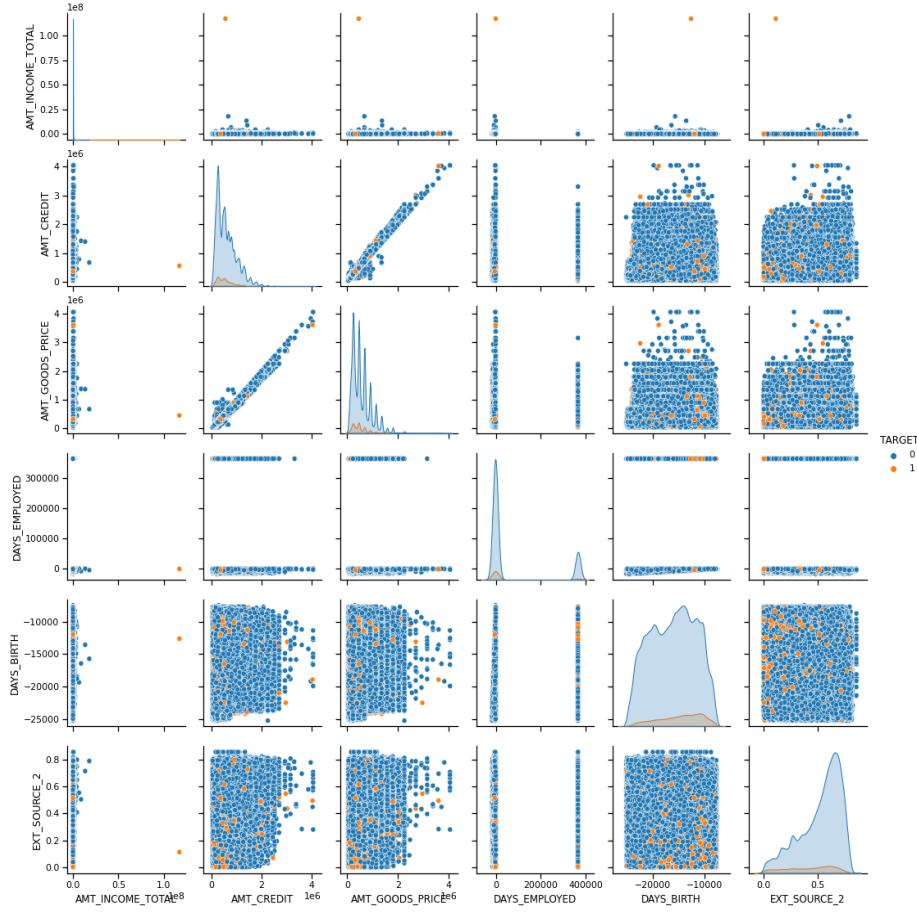


Ilustración 3. Ejemplo del reporte generado para una variable categórica.

Adicionalmente, se realiza también las distribuciones de las variables entre sí, con el objetivo de entender en mayor profundidad el fenómeno como tal, pues un valor extremo en un atributo podría ser explicado por lo demás sin precisar ser un outlier o valor atípico que amerite un tratamiento diferenciado.



*Ilustración 4. Gráficas generadas para comparar la distribución de algunas variables numérica significativas.*

### 2.2.2. Preparación de los datos

Una vez entendidos en términos de la composición del conjunto, los atributos de calidad y las distribuciones, por ahora, univariadas de los atributos, ya se cuenta con un panorama claro que permite implementar acciones de mejoramiento del dataset y tomar decisiones respecto a los atributos que serán contemplados para obtener la solución propuesta. Así entonces, se efectúan análisis y acciones en cuestión del tratamiento de valores faltantes, correlación entre las variables, análisis del balance de las variables categóricas y transformación de las variables categóricas a variables numéricas mediante algún método de representación de características.

### 2.2.2.1. Tratamiento de valores faltantes:

Como se mencionó en secciones anteriores, el conjunto de datos posee un 24,4% de valores faltantes o valores nulos. Al explorar con mayor minucia este fenómeno en el data set, se detectó que 57 atributos del dataset, poseen más del 10% de los registros con campos vacíos. En función de esto, se toma como primera medida descartar aquellos atributos que exhiben más del 20% de los valores faltantes, por lo que se descartan 50 columnas en el dataset, lo que corresponde al 40,9% del conjunto de datos.

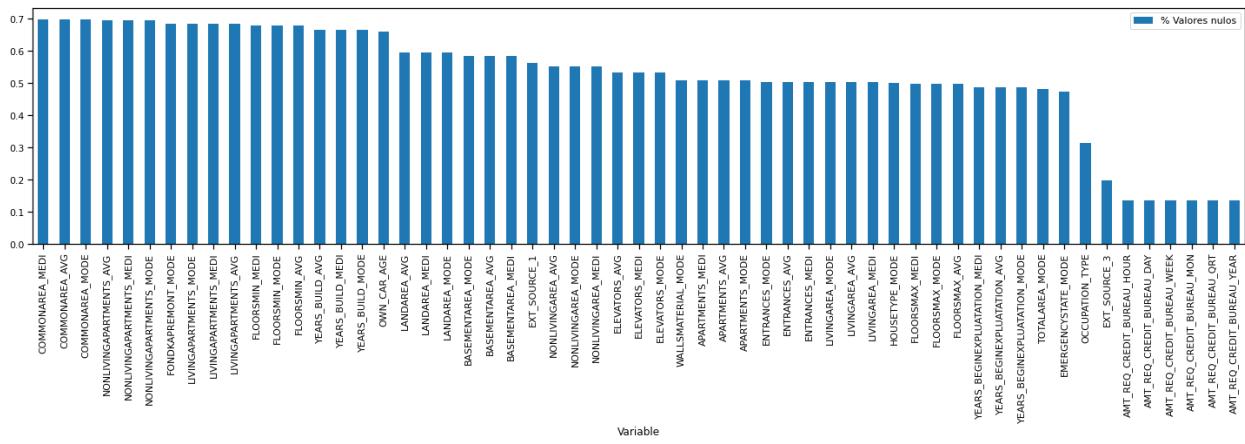
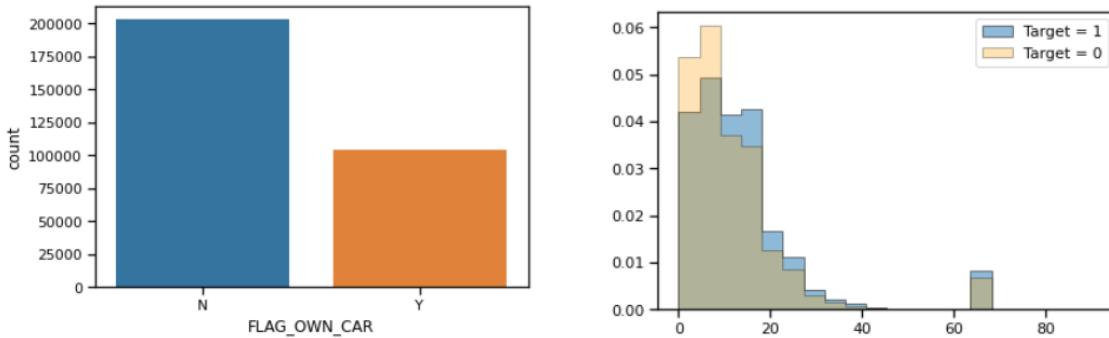


Ilustración 5. Cantidad de valores faltantes por variable. Atributos con más del 10% de valores faltantes.

Una aclaración importante, es que, desde el punto de vista de negocio, antes de descartar variables con una proporción significativa de valores faltantes, para algunas de ellas se hizo un análisis del impacto directo sobre la variable objetivo, esto con el fin de determinar el causal de la cantidad de valores faltantes. Es decir, si corresponde a un mecanismo MCAR (missing completely at random, faltante totalmente al azar), MAR (missing at random, faltante al azar) o MNAR (missing not at random, faltante no al azar).

- Atributos FLAG\_OWN\_CAR y OWN\_CAR\_AGE:

El atributo FLAG\_OWN\_CAR es un valor booleano que indica si el usuario posee o no un vehículo. Por su parte, OWN\_CAR\_AGE corresponde a la antigüedad del vehículo en caso de poseer. De ahí que, naturalmente, estos dos atributos se hayan analizado en conjunto.



*Ilustración 6. Distribución de los atributos FLAG OWN CAR y OWN CAR AGE.*

Teniendo en cuenta lo observado, es posible notar que el atributo asociado a la antigüedad del vehículo no tiene impacto o variación en cuanto a si ese cliente fue clasificado o no como moroso. Adicionalmente, se identificó que la cantidad de valores faltantes en este atributo en función del TARGET, mantienen una proporción similar a la presentada por la distribución del TARGET.

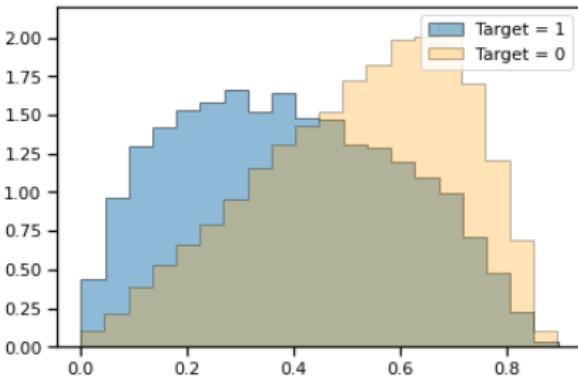
Proporción de valores faltantes cuando TARGET es 1 = 8,5%

Proporción de clientes calificados como morosos, respecto al total = 8,1%

En este orden de ideas, se infiere no sólo que no tiene impacto significativo sobre la variable objetivo, sino que adicionalmente, es posible visualizar que el comportamiento de los valores faltantes no depende tampoco de la variable objetivo, puesto que la proporción de estos es similar. Así entonces, los atributos son descartados del modelo, dado que, al no aportar variabilidad ni explicación, generarán ruido en el dataset a la hora de tratar de ajustar algún modelo, inyectando un sesgo, claramente no deseado.

#### - Atributo EXT\_SOURCE\_3:

Este atributo corresponde a una calificación que se obtiene de fuentes de información externas al negocio. Dentro del dataset se identifican 3 de esta naturaleza, siendo descartadas las dos primeras dado que poseen más del 50% de sus valores faltantes. En cambio, para la tercera fuente de calificaciones, la cantidad de valores faltantes es menor al 20%, motivo por el cual se revisa en detalle, evaluando el impacto sobre la variable objetivo.

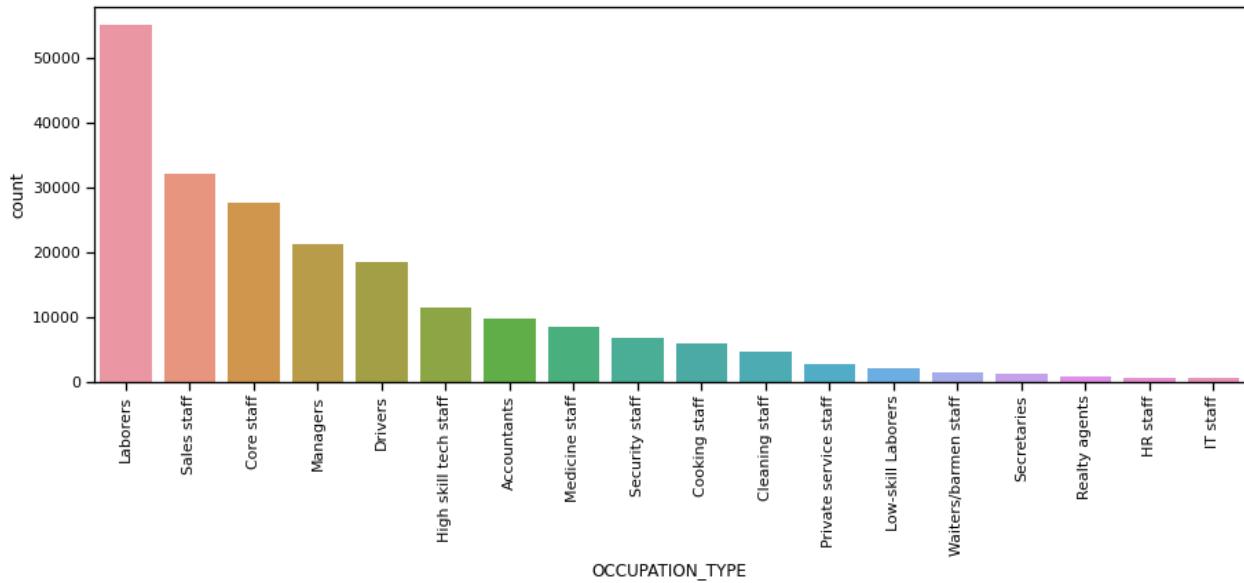


*Ilustración 7. Distribución del atributo EXT\_SOURCE\_3 en función del TARGET.*

El atributo 'EXT\_SOURCE\_3' si puede tener impacto significativo sobre la variable objetivo, teniendo en cuenta que la distribución de este cambia en función de ella. Por eso, no se descarta en función de sus valores faltantes, y en cambio se realizará la imputación de valores faltantes, para mejorar la completitud del atributo.

- Atributo OCCUPATION\_TYPE:

Este atributo hace alusión a una clasificación generada para segmentar la actividad económica en la que el usuario desempeña sus funciones laborales.



*Ilustración 8. Frecuencia del atributo OCCUPATION\_TYPE.*

Con el fin de identificar el impacto que puede haber en función del TARGET, y así optar por la decisión más adecuada en cuanto al tratamiento de los valores faltantes, se analiza cómo es la distribución de las proporciones del TARGET en función de OCCUPATION\_TYPE:

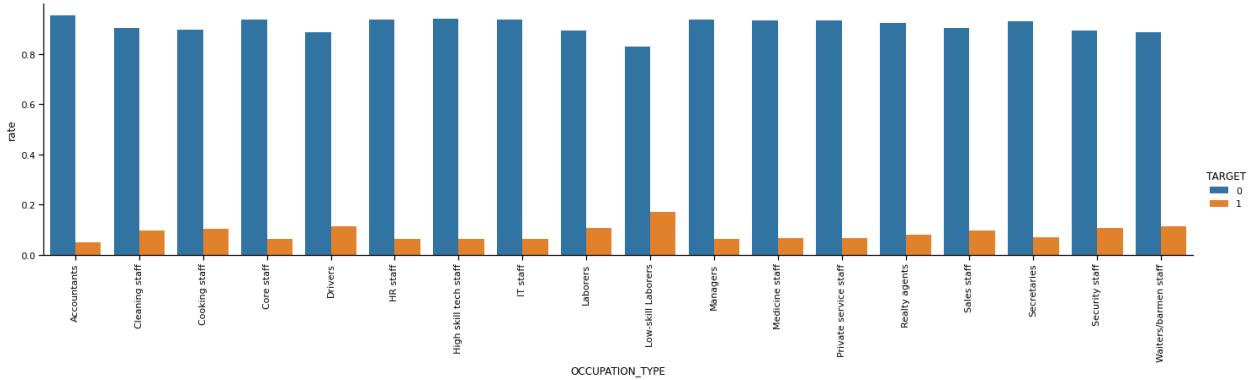


Ilustración 9. Distribución de las proporciones del TARGET en función de OCCUPATION\_TYPE.

A primera vista, no se ve una diferencia significativa en cuanto a las proporciones, a excepción de la clase ‘Low-skill Laborers’, pues en esta es mayor la proporción de los usuarios que fueron catalogados como morosos. En este orden de ideas, a pesar de que no se está obteniendo una medida de impacto cuantitativa, optar por descartar este atributo no es del todo viable, pues por el fenómeno per se, este atributo es clave para identificar al usuario y, por ende, tener una aproximación más real con la predicción a efectuar.

Ahora bien, para realizar la imputación de los valores faltantes, el atributo AMT\_INCOME\_TOTAL, que corresponde a los ingresos totales del usuario, es analizado para en función de OCCUPATION\_TYPE, y así determinar por rango de ingresos, cuál es la ocupación de trabajo más frecuente en el dataset y utilizarla como clase de imputación. Importante mencionar que sólo se imputaron las clases cuyo valor de ingresos totales estuviera en el intervalo [50.000, 250.000] debido a que se presentan valores extremos en este atributo y estos si bien pueden ser explicativos del fenómeno, están en menor proporción y puede inyectar un sesgo en el conjunto de datos.

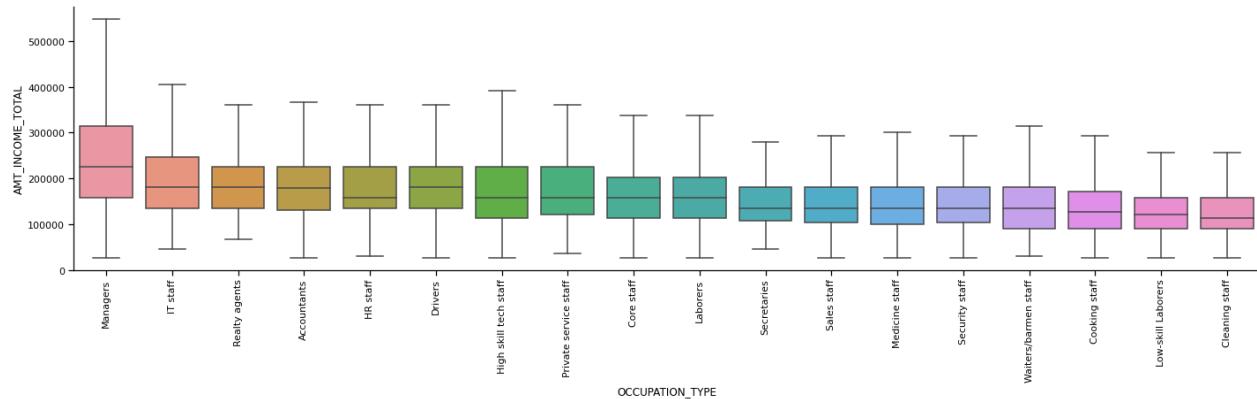
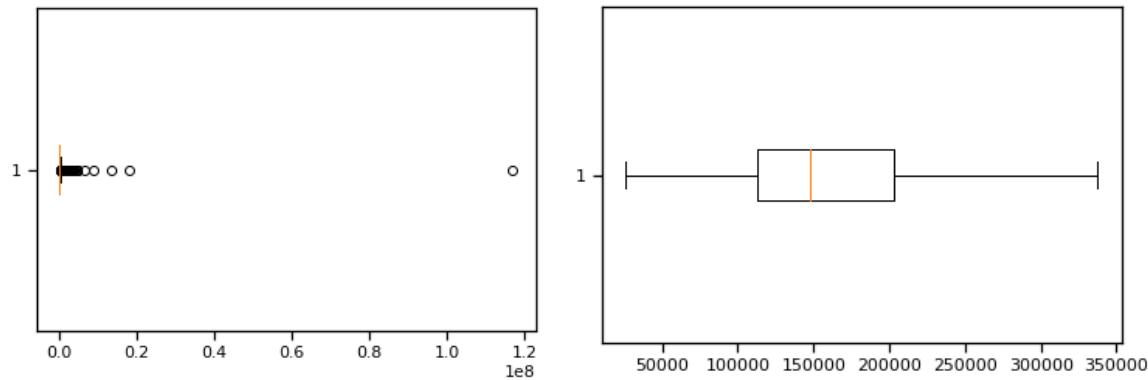


Ilustración 10. Distribución de AMT\_INCOME\_TOTAL, en función de OCCUPATION\_TYPE



*Ilustración 11. Diagramas boxplot para AMT\_INCOME\_TOTAL*

Con lo aplicado, se logran imputar 66% de los registros para el atributo OCCUPATION\_TYPE, pasando de 88.759 a 29.876, que serán analizados de forma integral en el conjunto de datos.

#### *2.2.2.2. Atributos desbalanceados:*

En el dataset, dado el análisis exploratorio y de entendimiento realizado para cada atributo, se determinó que los siguientes atributos serían descartados del conjunto de datos, dado que la distribución de sus clases tenía clase dominante en por lo menos 98%, por lo que, al no tener variabilidad, no aportan explicación al modelo.

- FLAG\_MOBIL
- FLAG\_CONT\_MOBILE
- FLAG\_DOCUMENT\_2
- FLAG\_DOCUMENT\_4
- FLAG\_DOCUMENT\_7
- FLAG\_DOCUMENT\_10
- FLAG\_DOCUMENT\_12
- FLAG\_DOCUMENT\_17
- AMT\_REQ\_CREDIT\_BUREAU\_HOUR
- AMT\_REQ\_CREDIT\_BUREAU\_DAY
- AMT\_REQ\_CREDIT\_BUREAU\_WEEK
- AMT\_REQ\_CREDIT\_BUREAU\_MON
- AMT\_REQ\_CREDIT\_BUREAU\_QR

#### 2.2.2.3. Correlación entre atributos:

Una vez más, fruto del análisis exploratorio de los datos, se detectó que los atributos AMT\_CREDIT, que corresponde al monto cuándo el producto financiero es un crédito y AMT\_GOODS\_PRICE, indicativa al monto del producto financiero en general, presentan una correlación de 0.99, lo que permite concluir que la variabilidad de uno explica directamente la del otro. Para estos era de esperarse, pues el dataset hace alusión a un problema, donde los productos financieros son todos créditos.

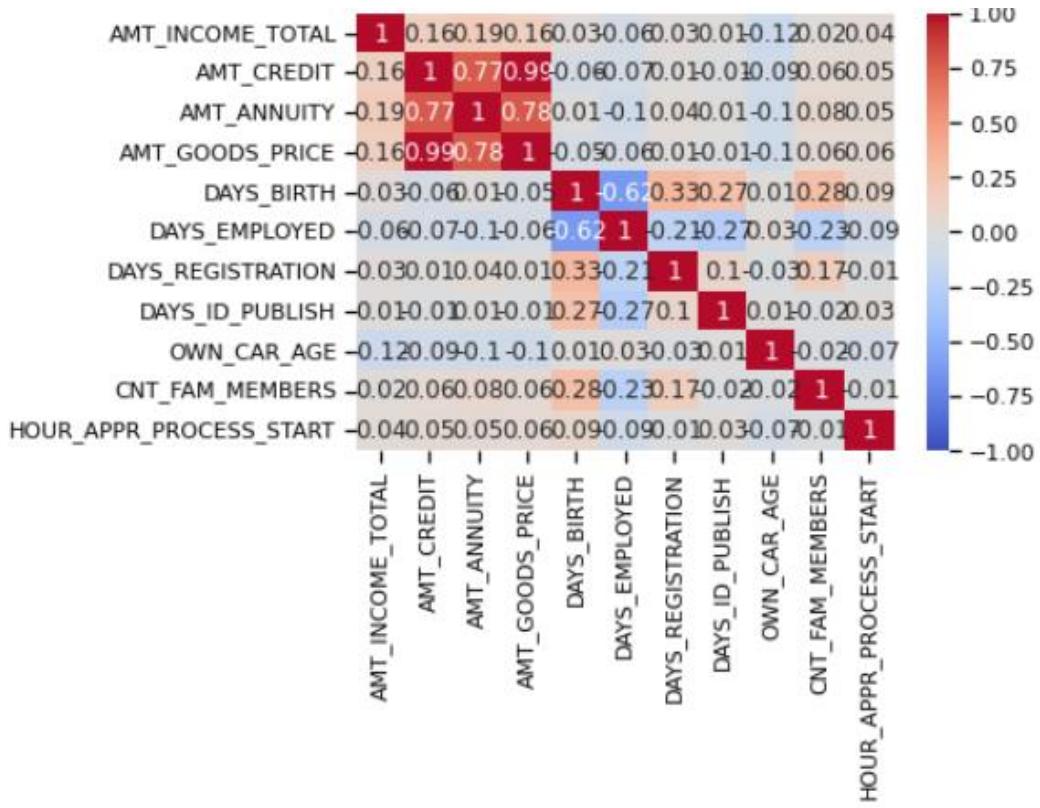


Ilustración 12. Mapa de calor con la correlación entre atributos relevantes.

#### 2.2.2.4. Atributos representativos de valores agregados:

En el dataset, se detectaron 47 atributos que corresponden a valores agregados que explican para segmentos de usuarios información asociada a las características de las viviendas de estos. Dado que, para estos puntualmente, la menor proporción de valores faltantes es de aproximadamente el 50%, se opta por descartarlos del conjunto objeto de modelamiento.

#### *2.2.2.5. Balanceo de atributos categóricos:*

Hasta ahora, todas las operaciones que se han realizado para la preparación del dataset a partir del entendimiento obtenido del negocio, y de la estructura misma de los datos, su naturaleza, tipología y distribución, han sido a nivel de columnas o atributos. Esto se refleja en qué el dataset inicial o crudo, está compuesto por 307.511 registros y 122 columnas, mientras que, con las acciones implementadas, se tiene un data set de 307.511 registros y 59 columnas. A continuación, se empiezan a implementar ciertas estrategias para continuar con la preparación de los datos, pero esta vez, a nivel de registros en función de la distribución y composición de las variables tanto numéricas como categóricas.

##### - Atributo NAME\_INCOME\_TYPE:

De este atributo, que corresponde a la clasificación que un usuario obtiene según el origen de sus ingresos, se descartan las clases ‘Unemployed’, ‘Student’, ‘Businessman’ y Maternity leave, pues representan menos del 0,001% de los registros asociados. Así entonces, estas son las clases que son conservadas y su respectiva frecuencia:

- » Working: 158774
- » Commercial associate: 71617
- » Pensioner: 55362
- » State servant: 21703

Tamaño del dataset: (307.456, 59)

##### - Atributo NAME\_FAMILY\_STATUS:

Hace alusión al estado civil o estado familiar del usuario del crédito. Se descarta la clase ‘Unknown’ dado que sólo aparece en dos ocasiones en todo el conjunto de datos. Las clases conservadas son las siguientes:

- » Married: 196401
- » Single / not married: 45433
- » Civil marriage: 29767
- » Separated: 19768
- » Widow: 16085

Tamaño del dataset: (307.454, 59)

- Atributo NAME\_EDUCATION\_TYPE:

Corresponde a la clasificación asignada al grado de educación que tiene el usuario al momento de adquirir el producto financiero con la entidad. Para esta, se descarta la clase ‘Academic degree’ debido a su muy baja presencia en el dataset. Adicionalmente, las clases ‘Lower secondary’ y ‘Incomplete higher’ son agrupadas en una única representada por ‘Incomplete higher’, quedando entonces las siguientes clases:

- » Secondary / secondary special: 218365
- » Higher education: 74837
- » Incomplete higher: 14088

Tamaño del dataset: (307.290, 59)

- Atributo NAME\_TYPE\_SUITE:

Indica si el usuario fue acompañado o no, y si sí, por quién fue acompañado al momento de haber hecho la solicitud para adquirir el producto con la entidad. En este caso, se opta por agrupar todas las clases en sólo dos, acompañado o no acompañado:

- » Unaccompanied: 248335
- » Accompanied: 58955

Tamaño del dataset: (307.290, 59)

#### *2.2.2.6. Representación de variables categóricas a variables numéricas:*

El siguiente paso considerado durante la preparación de los datos, es representar las variables categóricas como variables numéricas según su naturaleza, ya que en función de esto se debe definir la técnica más apropiada.

- Label Encoding:

Para los atributos binarios desde su definición, se aplicó la técnica de Label Encoding, que consiste en asignar un valor numérico a cada clase. En este caso se asignaron 0's y 1's dado que, como se menciona, sólo se aplica para los atributos numéricos. Estos fueron NAME\_CONTRACT\_TYPE, CODE\_GENDER, FLAG\_OWN\_CAR, FLAG\_OWN\_REALTY.

- One-Hot Encoding:

Consiste en representar los atributos como combinación lineal de nuevos atributos numéricos binarios, donde se crean  $k - 1$  nuevas variables, conocidas como variables dummies, donde  $k$  corresponde a la totalidad de clases del respectivo atributo. Mediante esta técnica, los siguientes atributos fueron representados: NAME\_TYPE\_SUITE, NAME\_INCOME\_TYPE, NAME\_EDUCATION\_TYPE, NAME\_FAMILY\_STATUS, NAME\_HOUSING\_TYPE, OCCUPATION\_TYPE, WEEKDAY\_APPR\_PROCESS\_START.

- Frequency Encoding:

Esta técnica consiste en asignar un valor numérico a cada clase, el cuál corresponde a la frecuencia relativa de cada clase dentro del conjunto de datos. Cabe mencionar que, este valor siempre será un número contenido en el intervalo (0, 1). Se aplicó para el atributo ORGANIZATION\_TYPE, puesto que este contaba con bastantes clases sin existir una o varias altamente dominantes, por lo que es importante contemplarlas todas, pues hacen parte de la explicación de la variabilidad del conjunto de datos.

Con estas transformaciones, el dataset pasa a tener el siguiente tamaño: (307.290, 81).

#### *2.2.2.7 Imputación de valores faltantes:*

Con las transformaciones efectuadas para contar con todo un conjunto de datos numéricos, es posible ahora implementar alguna técnica analítica para terminar la preparación de los datos. En este caso, se opta por realizar una imputación de los valores faltantes mediante un imputador KNN en todo el dataset. Para esto, se define que el imputador estará evaluando distancias euclídeas en el data set sin asignar ponderación en función de las distancias.

Con esto entonces, se tiene un dataset de tamaño (307.290, 81) sin valores faltantes dentro de su estructura, y compuesto en su totalidad por atributos numéricos, dando por terminada y satisfactoria la preparación de los datos para los posteriores modelamientos.

## 2.3. Modelado de los datos:

Una vez los datos han sido analizados en profundidad y se han implementado acciones para aumentar la calidad del dataset, además de obtener atributos numéricos solamente, se tiene el punto de partida para empezar a implementar acciones analíticas que permitan obtener un predictor de calidad y así dar respuesta al requerimiento que el problema expone.

### 2.3.1. Escalado de los datos:

Con el fin de eliminar el impacto por las diferentes magnitudes que son evaluadas en el dataset, y que claramente no son comparables entre sí, todos los atributos fueron escalados mediante normalización o tipificación, siendo representados por su valor Z luego de centrar y escalar los datos.

	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	REGION_POPULATION_RELATIVE	DAYS_BIRTH	DAYS_EMPLOYED
count	3.072900e+05	3.072900e+05	3.072900e+05	3.072780e+05	3.072900e+05	3.072900e+05	3.072900e+05
mean	5.248552e-16	-8.371633e-16	8.472627e-17	-1.294447e-16	6.593855e-16	-1.625640e-16	1.098160e-15
std	1.000002e+00	1.000002e+00	1.000002e+00	1.000002e+00	1.000002e+00	1.000002e+00	1.000002e+00
min	-5.775839e-01	-6.034443e-01	-1.376533e+00	-1.759587e+00	-1.488091e+00	-2.106244e+00	-5.784584e-01
25%	-5.775839e-01	-2.371970e-01	-8.174006e-01	-7.303881e-01	-7.853935e-01	-8.352081e-01	-4.712008e-01
50%	-5.775839e-01	-9.487294e-02	-2.122186e-01	-1.519491e-01	-1.457624e-01	6.576812e-02	-4.602499e-01
75%	8.072493e-01	1.423338e-01	5.211613e-01	5.172010e-01	5.639505e-01	8.304064e-01	-4.537091e-01
max	2.573425e+01	4.926785e+02	8.576017e+00	1.594151e+01	3.734985e+00	1.958689e+00	2.133810e+00

Ilustración 13. Fragmento del resumen de la distribución de los datos una vez escalados.

En la Ilustración 13 se presenta un fragmento del resumen de la distribución de los datos escalados. Como se resalta, la media de todos estos valores es un número muy cercano a cero, mientras que la desviación estándar, por su parte, es muy cercana a 1, comportamiento finalmente esperado con la normalización aplicada. Ahora, todos los atributos están representados como la razón de la distancia del atributo a la media, en términos de la desviación estándar.

### 2.3.2. Reducción de la dependencia lineal entre atributos:

Dado que el objetivo es poder contar con atributos independientes entre sí, de tal manera que puedan explicar la variabilidad del fenómeno analizado y cómo esta impacta la variable objetivo, mitigando el sesgo lo máximo que sea posible, se evalúa la explicabilidad de los atributos del conjunto de datos, en función de los demás. Esto se hace, utilizando el coeficiente de determinación  $R^2$  para cada atributo en función de los demás, explicados como una regresión lineal.

Se obtuvo que 16 atributos presentaron un valor de  $R^2$  en función del resto del dataset, mayor a 0.9, es decir reflejan una explicabilidad sumamente alta y, por ende, una dependencia lineal presente en el dataset. Para tratar este comportamiento, se implementó un proceso iterativo con el fin de ir eliminando los atributos de mayor dependencia en función del dataset, evaluar el cambio, y repetir el procedimiento hasta que el valor del coeficiente de determinación presente en el data set no superara el umbral definido, que para este caso fue de 0.9.

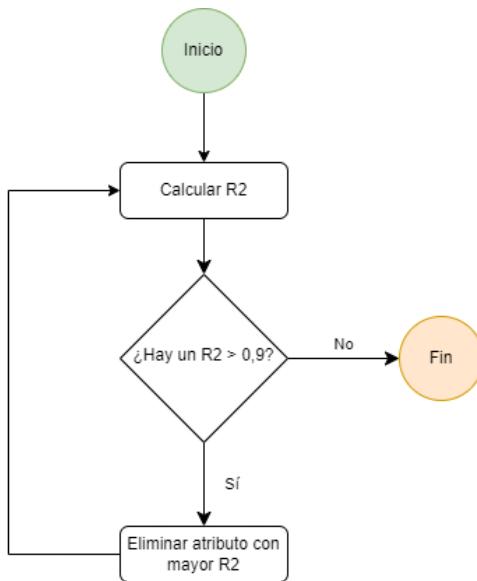


Ilustración 14. Proceso iterativo para reducir la dependencia lineal en el data set.

Como resultado final de este proceso, se eliminaron 10 atributos que estaban induciendo la mayor dependencia lineal en el conjunto de datos. Así entonces, el dataset procesado queda con un tamaño de (307.290, 71).

### 2.3.3. Detección de Outliers:

Hasta ahora, todo el análisis que se ha efectuado ha sido en mayor medida univariante, a excepción de este último paso, pues se evaluó la explicabilidad de un atributo en función de los demás. El paso siguiente será analizar la distribución de los registros, pero tomando cada uno de ellos como un punto en  $R^n$ . En mayor medida, este ejercicio se realiza calculando la distancia de cada uno de los puntos con respecto a la media, sin embargo, para obtener una aproximación más robusta, se tendrá como punto de referencia o comparación, el vector de medianas. Adicionalmente, se aclara que la distancia que se utiliza en este caso es la distancia de Mahalanobis pues es más adecuada para el análisis de datos multivariantes, pues no sólo considera la magnitud de la diferencia, sino que también considera la variabilidad de los puntos, por lo que logra una mejor identificación del comportamiento de los datos.

$$d_{mah}(X_i, Me) = (X_i - Me)^T \cdot S^{-1} \cdot (X_i - Me)$$

Dónde,

- $X_i$  es el punto analizado
- $Me$  es el vector de medianas
- $S^{-1}$  es la inversa de la matriz de covarianzas

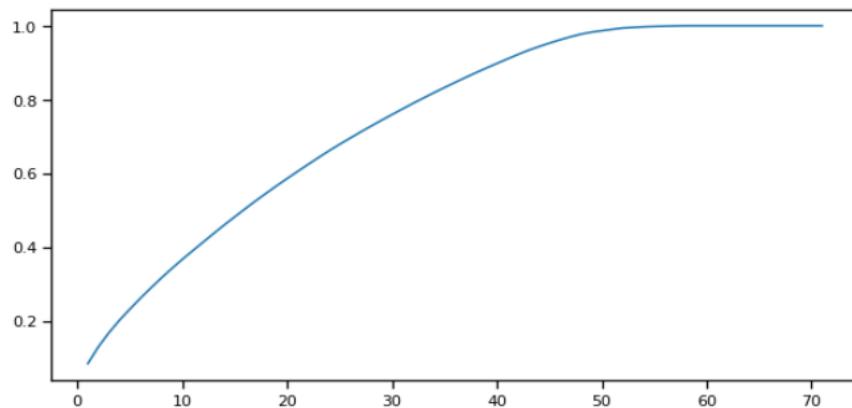
Para este caso puntual, teniendo en cuenta que la matriz de covarianzas es considerada para el cómputo de la distancia de Mahalanobis, se analiza el número de condición obtenido cuando se realiza la estimación habitual, obteniendo un valor de 117.74. Teniendo en cuenta que este número es alto, es un indicativo de que a la hora de computar la matriz inversa se presentará incertidumbre en este cálculo, dado que la matriz está mal condicionada y por ende es altamente susceptible a pequeñas variaciones.

#### 2.3.3.1. Mejorando el condicionamiento de la matriz de covarianzas:

Ante lo previamente mencionado, es menester utilizar alguna alternativa que permita reducir el número de condición de la matriz de covarianzas y por ende reducir el impacto de la sensibilidad inducida por pequeñas variaciones. En este caso, se procede con realizar la estimación de la matriz de covarianza implementando el encogimiento de **Ledoit and Wolf**. El número de condición de esta matriz es de 70.00, valor que si bien es alto también cuando de un número de condición de una matriz se trata, es menor al obtenido con la estimación habitual. Estableciendo un umbral asociado al percentil 85 de las distancias obtenidas, el dataset queda ahora con 261.196 registros.

#### 2.3.4. Reducción de dimensionalidad:

Remarcando que el actual dataset, una vez implementadas los pasos anteriores cuenta con 71 atributos, es muy valioso buscar la forma de reducir la dimensionalidad del conjunto de datos, puesto que el costo computacional incrementa a medida que la cantidad de atributos también aumenta. Para esto, se efectúa un análisis de componentes principales, dando validez a esta técnica, el hecho de que en operaciones anteriores se redujo la dependencia lineal de los atributos. Para la decisión correcta, se tomó la transformación de los atributos a un número equivalente de componentes principales, y así evaluar la variabilidad explicada acumulada por las componentes principales obtenidas, como se muestra en la siguiente gráfica:



*Ilustración 15. Variabilidad explicada acumulada en función del número de componentes principales.*

Como es posible identificar, a partir de las componentes principales obtenidas, el 90% de la variabilidad del dataset es posible explicarlo con 39 componentes, valor inferior a la dimensionalidad original. Así entonces, para las siguientes operaciones, se trabajará con el dataset equivalente obtenido a partir de la transformación de los datos. Este, es un dataset de tamaño (261.196, 39).

### **2.3.5. Modelo de aprendizaje supervisado:**

Con el contexto del problema en cuestión, una solución factible es la implementación de un modelo de aprendizaje supervisado que permita clasificar a los usuarios como morosos o no morosos. Una vez los datos preparados, es posible pasar a entrenar los modelos, con el fin de determinar la mejor configuración de éste y seleccionar el de mejor desempeño.

#### *2.3.5.1. Métricas de evaluación de los modelos:*

Para este caso puntual, teniendo en cuenta que el atributo TARGET, el cuál es el objetivo, presenta un desbalance en las clases, siendo mayoritaria la presencia de usuarios no morosos, es decir, etiquetados con 0, se tendrán en cuenta las siguientes métricas para la selección y optimización del modelo:

- Precisión: Número de elementos identificados correctamente en una clase, con respecto al total de los elementos predichos para esa clase.
- Recall o sensibilidad: Número de elementos identificados correctamente en una clase, con respecto al total de los elementos reales para esa clase.
- F1-Score: Media armónica de la Precisión y la Sensibilidad.

$$F1\ Score = \frac{2 \cdot Precision \cdot Sensibilidad}{Precision + Sensibilidad}$$

- Área bajo la curva ROC (Receiver Operating Characteristic): Representación de la sensibilidad en función de los falsos positivos para distintos puntos de corte. El área bajo la curva puede ser interpretada como la probabilidad de que, ante un par de observaciones de distinta clase, el predictor los clasifique correctamente.

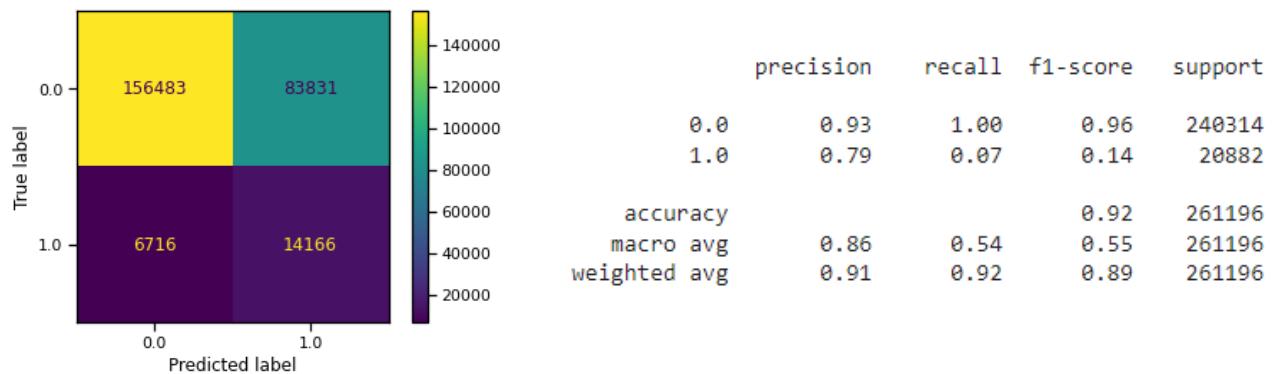
#### *2.3.5.2. Modelos clasificadores:*

Considerando la naturaleza del problema, y que la variable a predecir es un atributo binomial, se analizan en primera instancia los modelos árbol de decisión, regresión logística y bosques aleatorios.

- Árbol de decisión:

Se presentan a continuación los escenarios evaluados con el modelo en cuestión, dadas las características del dataset y del problema analizado.

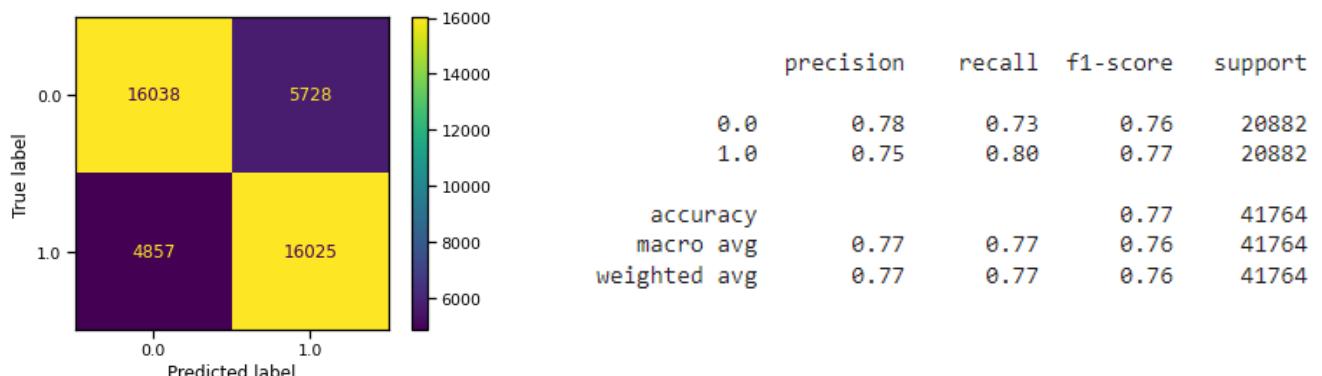
» Árbol de decisión, máxima profundidad 10, todo el dataset para el entrenamiento:



*Ilustración 16 Evaluación árbol de decisiones, profundidad definida, todo el dataset.*

En general, se obtuvo un accuracy de 0,92 lo que indica que el modelo, en términos generales el modelo tiene una alta tasa de clasificación. Sin embargo, al revisar en detalle el recall y el F1-Score para la clase 1, es decir, que sea moroso, el desempeño es bastante bajo. Comportamiento esperado en función del desbalance del conjunto de datos.

» Árbol de decisión, máxima profundidad 10, seleccionando aleatoriamente registros con TARGET igual a cero, de tal manera que se obtenga un dataset de entrenamiento balanceado.



*Ilustración 17 Evaluación árbol de decisiones, profundidad definida, selección aleatoria para balancear.*

En este caso, es evidente la mejora en cuanto a la capacidad del modelo para clasificar la clase objetivo. Sin embargo, el accuracy disminuye en comparación con el anterior. Sin embargo, como se mencionó previamente, este no será criterio para medir la calidad del modelo clasificador.

Para este caso, también fue implementado el entrenamiento y la validación del desempeño del modelo aplicando la técnica de validación cruzada, con el fin de analizar cómo es la clasificación cuando se valida con datos que no fueron objetos de entrenamiento, además de revisar si hay un sesgo puntual por el subconjunto de datos que se usa para entrenar.

Se obtuvieron los siguientes resultados:

F1 score medio en Train: 0.77 - F1 score medio en Test: 0.68

Precisión obtenida en Train: 0.77 - Precisión obtenida en Test: 0.67

Como era de esperarse, al momento de realizar la predicción con los subconjuntos de validación, las métricas fueron menores. Sin embargo, revisando los subconjuntos generados, no hubo variación significativa entre los modelos comparados, por ende, se puede inferir que en el entrenamiento no hay sesgo asociado al subconjunto seleccionado para entrenar el modelo.

» Análisis de la sensibilidad del árbol de decisiones en función de la profundidad.

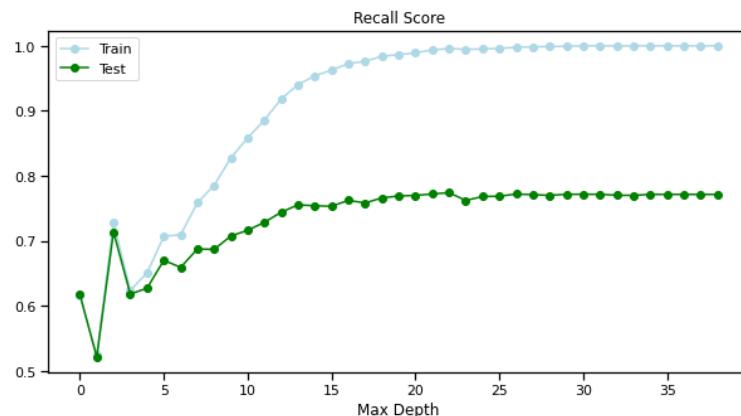


Ilustración 18 Recall score obtenido en función de la profundidad.

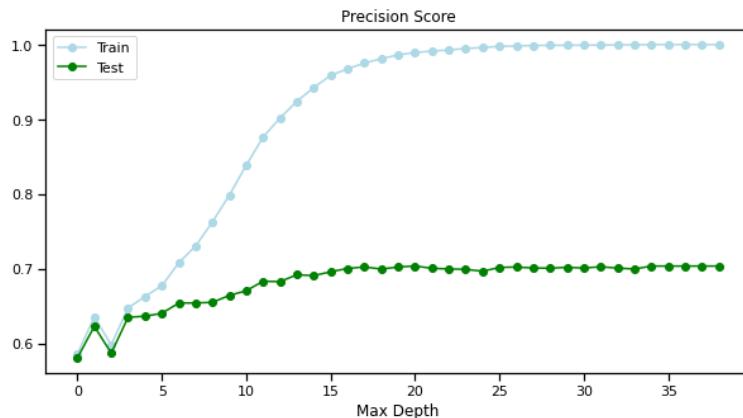


Ilustración 19 Precision score obtenido en función de la profundidad

Es claramente evidente cómo el desempeño del modelo aumenta hasta obtener incluso valores de las métricas iguales a 1,0. En este caso, este comportamiento no es deseado, dado que refleja que el modelo está quedando sobreajustado. Así pues, un valor adecuado para el hiperparámetro de la profundidad de árbol de decisiones es de 10.

- » Análisis de la sensibilidad del árbol de decisiones, evaluando varios grupos de registros con TARGET igual a cero, cubriendo todo el dataset y así evaluar el sesgo por tasa.

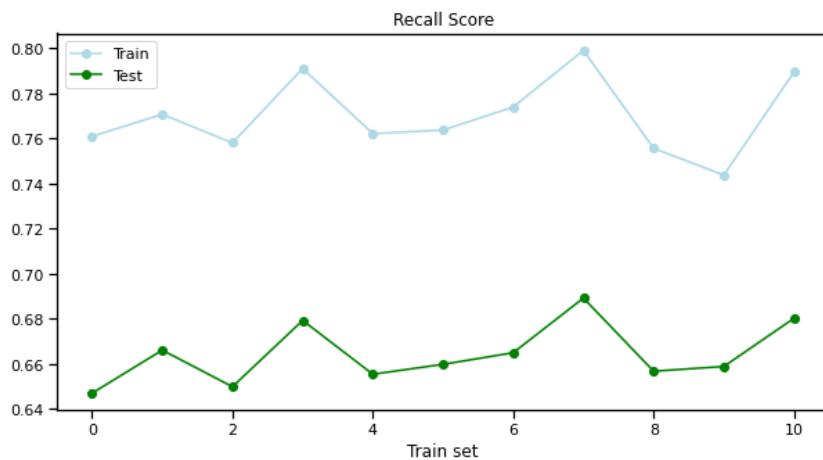


Ilustración 20 Recall score obtenido cuando se varía la selección de datos con TARGET igual a cero.

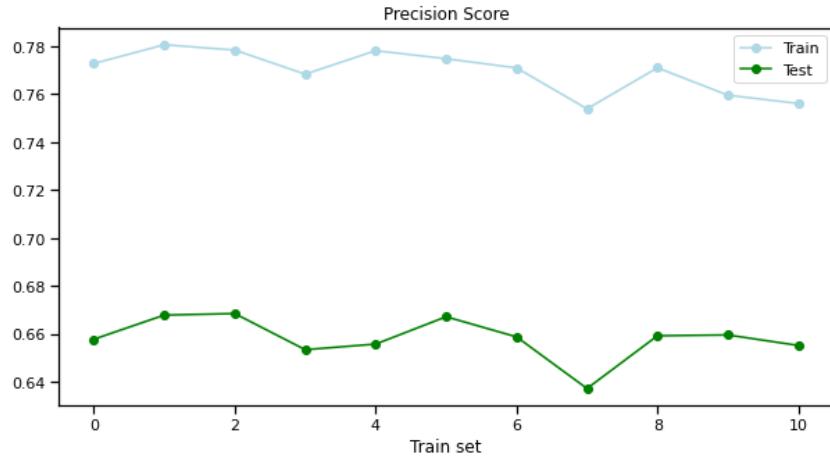


Ilustración 21 Precision score obtenido cuando se varía la selección de datos con TARGET igual a cero.

Teniendo en cuenta que la variación de las métricas, cuando se cambia el conjunto de datos con TARGET igual cero aleatoriamente, son pequeñas, del orden de 0,02 tanto para el recall score como para el precision score, se confirma que el entrenamiento del modelo se puede realizar con la selección aleatoria de un conjunto de datos de tamaño tal que equipare la cantidad de datos con TARGET igual a uno.

- Regresión Logística:

Para el caso de la regresión logística, se tomarán parte de las conclusiones obtenidas con el modelo anterior. Es decir, es factible realizar el entrenamiento del modelo seleccionando aleatoriamente un conjunto de datos con TARGET igual a 0 para balancear el dataset de entrenamiento.

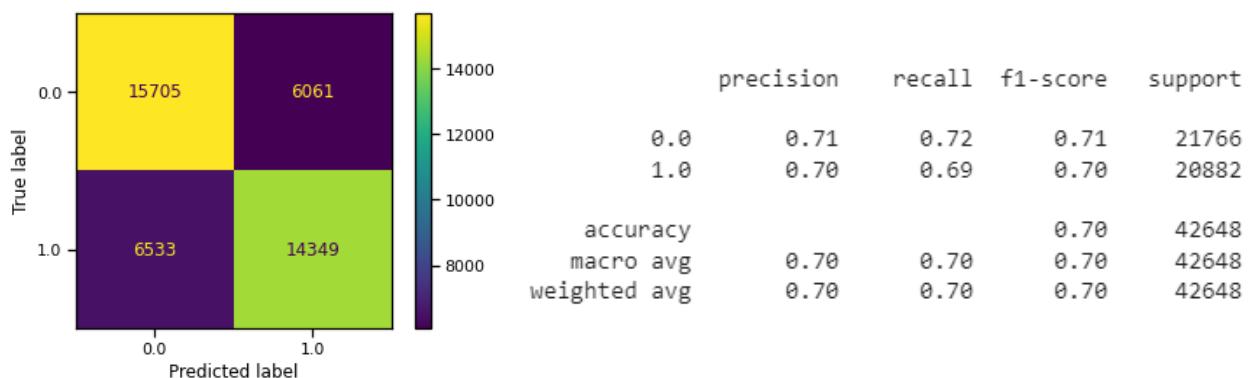


Ilustración 22 Resultados obtenidos con el modelo de regresión logística usando un dataset balanceado como entrenamiento.

Al aplicar validación cruzada:

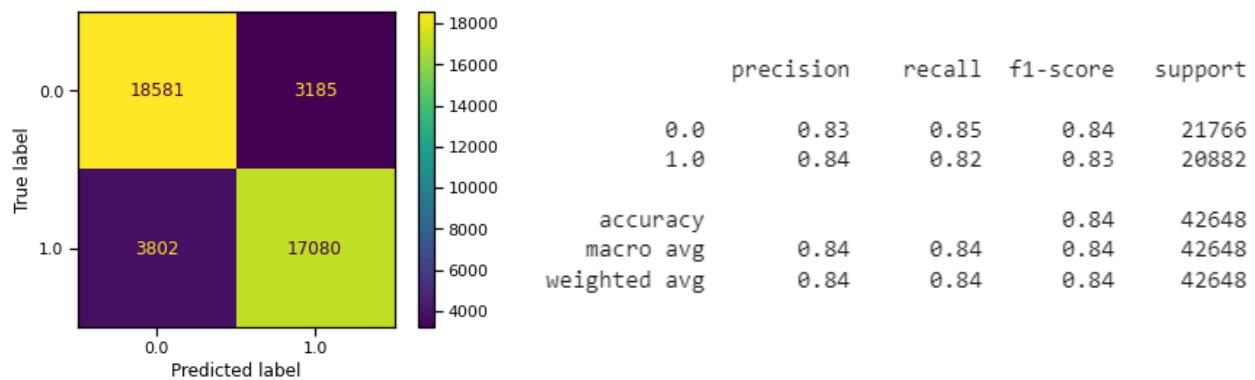
F1 score medio en Train: 0.71 - F1 score medio en Test: 0.71

Precisión obtenida en Train: 0.71 - Precisión obtenida en Test: 0.71

Así entonces, el modelo de regresión logística obtenido también puede ser considerado un buen predictor para entregar la solución al problema. Una desventaja del modelo de regresión logística es que para buscar un mejor desempeño se debería ajustar el proceso iterativo para hallar los pesos asociados.

- Bosques aleatorios:

Siguiendo la línea, se instancia un modelo clasificador de bosques aleatorios para un dataset balanceado, con 25 estimadores y una máxima profundidad de 10.



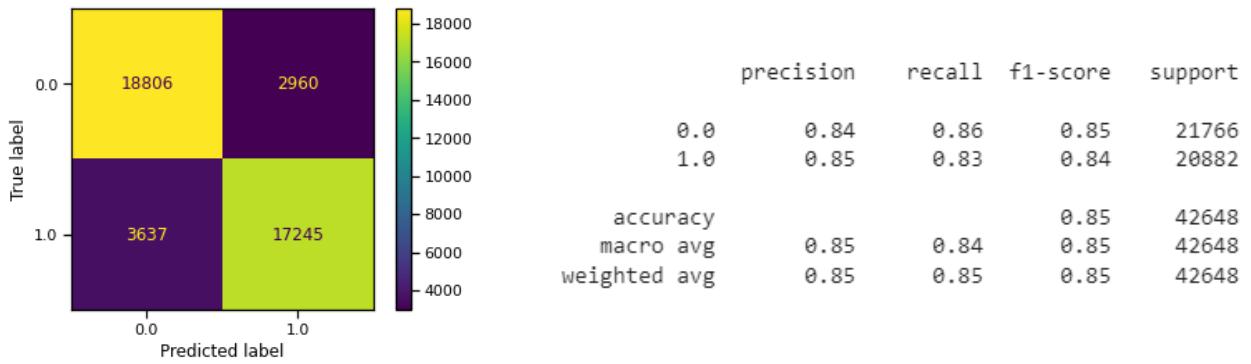
*Ilustración 23 Resultados obtenidos con el modelo de bosque aleatorio usando un dataset balanceado.*

Al aplicar validación cruzada:

F1 score medio en Train: 0.84 - F1 score medio en Test: 0.72

Precisión obtenida en Train: 0.85 - Precisión obtenida en Test: 0.73

Con estos resultados, este modelo muestra que no está sesgado por el conjunto de datos utilizado para el entrenamiento. Ahora bien, menester buscar cómo optimizar los hiperparámetros pues en este caso puntual, se puede ajustar la profundidad y el número de estimadores. Esto fue realizado con la técnica de GridSearchCV, que corresponde a iterar sobre un conjunto de posibles combinaciones para intervalos de hiperparámetros. En este caso, la mejor combinación sin entrar en riesgo de que el modelo quede sobreajustado a los datos de entrenamiento, fue un número de estimadores igual a 50 y máxima profundidad de 10.



*Ilustración 24 Resultados del modelo bosque aleatorios luego de optimizar hiperparámetros.*

Al aplicar validación cruzada:

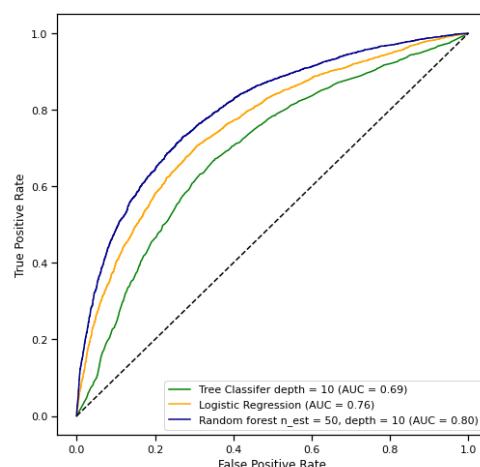
F1 score medio en Train: 0.85 - F1 score medio en Test: 0.73

Precisión obtenida en Train: 0.86 - Precisión obtenida en Test: 0.74

En este caso, las métricas mejoran en comparación con los modelos anteriormente mencionados. Sin embargo, para tener un criterio único de comparación, se analizará el área bajo la curva de los mejores modelos obtenidos para cada tipo de clasificador.

#### *2.3.5.3. Comparación de los modelos, por tipo:*

Se obtiene el área bajo la curva ROC para cada modelo de los anteriormente expuestos, esto con el fin de tener el criterio definitivo para realizar la selección del clasificador más adecuado para la solución en curso.



*Ilustración 25 Comparación de las áreas bajo la curva para los modelos analizados.*

Finalmente, se nota que el modelo de bosque aleatorio tiene un mejor desempeño para realizar las clasificaciones y brindar a los usuarios finales una herramienta que permita determinar si cliente será o no moroso.

### 3. Infraestructura e Ingeniería de Datos

#### 3.1. Definición de arquitectura:

Una vez entendido el problema de negocio, la conclusión de los objetivos generales y específicos y la composición del conjunto de datos fruto del análisis; se define la infraestructura con la que va a ser desarrollado, en este caso usando dos clouds con el fin de resaltar los beneficios de cada una de ellas, además de lograr una interfaz entre las dos que permita el flujo de datos de acuerdo a las definiciones funcionales establecidas.

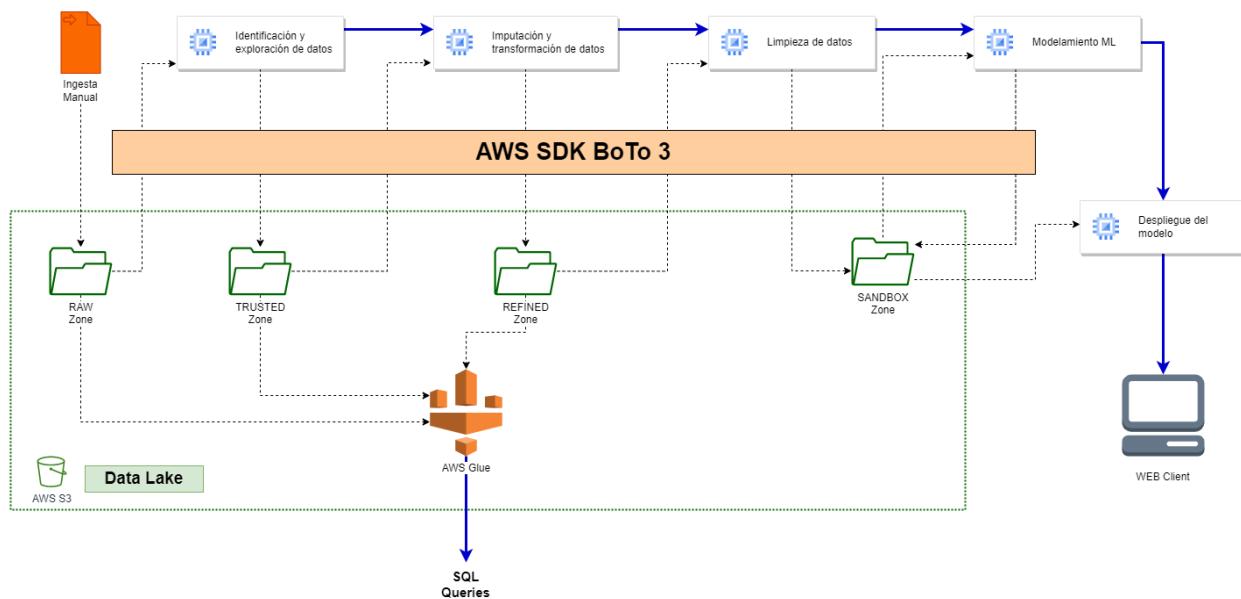


Ilustración 26. Diagrama de Arquitectura.

#### 3.2. Lineamientos del proceso ELT:

Dada la arquitectura definida, a continuación, se detallan los lineamientos para el flujo de datos en las etapas de extracción, almacenamiento, carga, transformación y exposición a los usuarios finales.

### **3.2.1. Proceso de ingestión:**

Dada la naturaleza de los datos y su respectivo origen, se define un proceso de ingestión o cargue de datos manual, a un data lake hospedado en infraestructura de AWS, puntualmente usando el servicio de S3 Buckets. Adicionalmente, cabe aclarar que, por la naturaleza del negocio, los datos son puestos a disposición en forma histórica, y por ende la arquitectura propuesta se plantea para el procesamiento de datos en **batch**.

### **3.2.2. Almacenamiento de datos:**

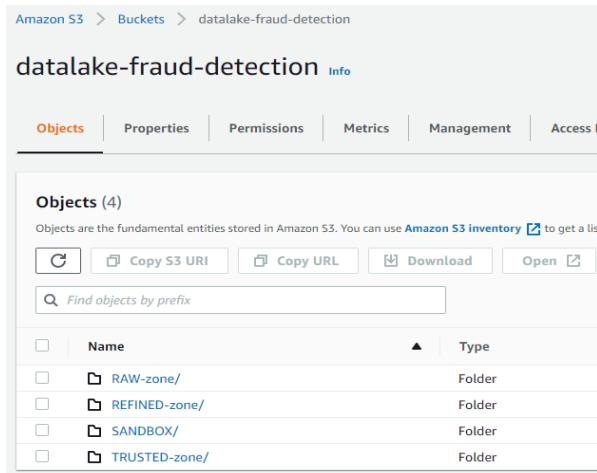
Para el almacenamiento de los datos objeto de la solución en curso, se configura en el servicio S3 Buckets de AWS, un data lake, teniendo en cuenta los beneficios de éste, como lo es la esquematización de los datos on-read, es decir, cuándo estos van a ser consumidos por un usuario final, además de la posibilidad de almacenar datos tanto estructurados, como semi-estructurados y no estructurados.

### **3.2.3. Transformación de datos:**

Se define el pipeline de preparación y transformación de datos, para ser alojados en las zonas RAW (Crudos), TRUSTED (de confianza), REFINED (Refinados) y SANDBOX (modelamiento), de la siguiente manera:

- Zona RAW - Datos crudos: Los datos se disponen tal cuál son extraídos desde el origen, que para este caso son obtenidos manualmente desde un repositorio de kaggle.
- Zona TRUSTED - Datos de confianza: En esta zona se disponen los datos una vez su esquema y estructura, adicionalmente estos se exponen vía SQL utilizando AWS Glue.
- Zona REFINED - Datos refinados: Los datos han pasado por múltiples transformaciones, incluyendo el tratamiento de datos faltantes desde su entendimiento e imputación, ingeniería de atributos para pasar las variables categóricas a una representación numérica y estandarización y escalado de los datos, con el fin de reducir el sesgo por la magnitud de los atributos.

- Zona SANDBOX - Modelamiento: En esta zona se dispone una réplica de los datos de la zona refined, pero acá, estos siguen siendo transformados, pues el objetivo principal es identificar un modelo de aprendizaje supervisado que permita predecir si un cliente es potencialmente identificado como moroso o no, y la probabilidad de dicha predicción.



*Ilustración 27. Imagen de referencia, del data lake hospedado en AWS S3.*

### 3.2.4. Entorno de procesamiento:

El entorno seleccionado para la transformación, procesamiento y modelamiento de los datos fue Google Colab (capa gratuita) teniendo en cuenta las bondades ofrecidas para estas labores. De destacar que en este entorno ofrece capacidades de cómputo aceptables para el procesamiento requerido, que en este caso es de 12Gb de RAM, y almacenamiento en disco de 100Gb. Importante mencionar que tiene la opción de habilitar una GPU para aumentar la velocidad de procesamiento.

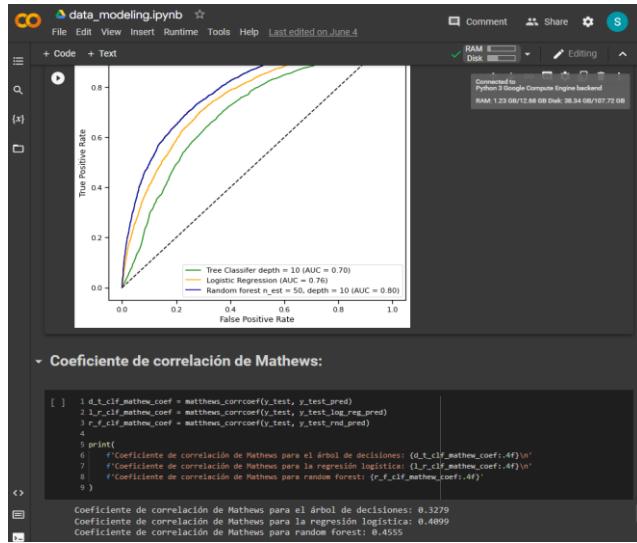


Ilustración 28. Imagen de referencia, entorno de procesamiento Google Colab.

### 3.2.5. Despliegue del modelo:

Para exponer el modelo de tal manera que un usuario final pueda realizar las predicciones respectivas y utilizarlas en el día a día de las operaciones y así tomar las decisiones oportunas, se utiliza la librería **streamlit** de python, dado que esta facilita la creación de aplicaciones web, con funcionalidad muy adoptadas para modelos de aprendizaje automático.

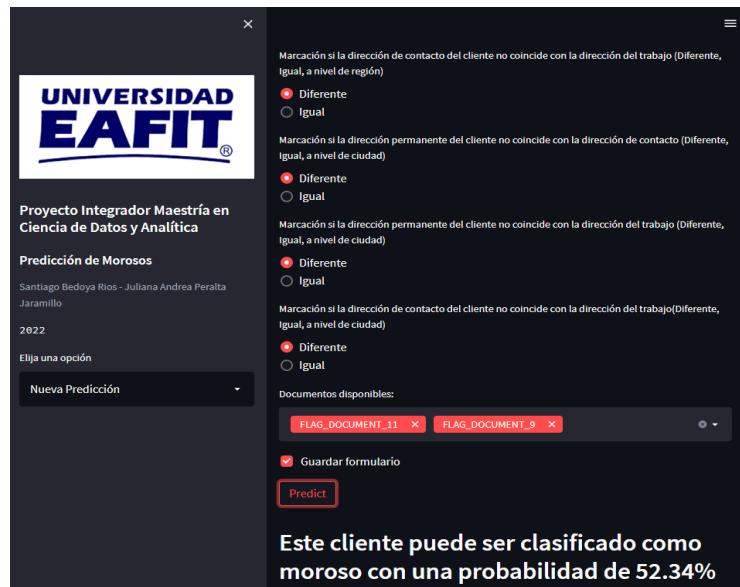


Ilustración 29. Imagen de la aplicación desplegada para el usuario final.

Esta es desplegada en producción mediante una instancia de Google Colab. Para exponerla en la web, se utiliza la librería de reverse proxy **ngrok**, también de Python con el fin de generar una URL para que los usuarios finales puedan acceder a los resultados.



```
✓ 0 # from pyngrok import ngrok
# Setup a tunnel to the streamlit port 8501
public_url = ngrok.connect(port='8501')
public_url

↳ <NgrokTunnel: "http://d82b-34-74-239-20.ngrok.io" -> "http://localhost:80">

✓ [41] tunnels = ngrok.get_tunnels()
tunnels

[<NgrokTunnel: "https://d82b-34-74-239-20.ngrok.io" -> "http://localhost:80">,
 <NgrokTunnel: "http://d82b-34-74-239-20.ngrok.io" -> "http://localhost:80">]

⌚ #!nohub streamlit run app.py
# !streamlit run app.py &>/dev/null&
!streamlit run --server.port 80 app.py >/dev/null
```

Ilustración 30 Imagen de referencia del proceso de despliegue e implementación del reverse proxy.

## **4. Conclusiones y consideraciones:**

El problema de la morosidad es un ejercicio que puede estar ligado a muchas características. Sin mostrar contradicción a esta premisa, al partir de un dataset de alta dimensionalidad, y en las que los atributos entre sí tienen muchas diferencias por la naturaleza y la tipología, se hizo necesario implementar muchas técnicas de interpretación y preparación de los atributos, así como ingeniería de características para lograr obtener un conjunto de datos numéricamente de calidad para así poder evaluar modelos con bajos niveles de sesgo propios del dataset como tal. Esto se vio reflejado en las métricas de los modelos evaluados, dado que la selección de los hiperparámetros fue relativamente sencilla en comparación con otros ejercicios propios de las ciencias de los datos. El criterio decisivo para escoger el modelo más adecuado fue la curva ROC, principalmente por la cobertura no sólo en las predicciones correctas, sino también porque tiene en cuenta la sensibilidad y la exhaustividad de los modelos.

El modelo logra ser desplegado en una aplicación amigable para que los usuarios de este pudiesen utilizarlo en las operaciones del día a día obteniendo resultados ágiles e interpretables de cara a la toma de decisiones en lo concerniente a la asignación o no de un producto financiero. Uno de los retos más importantes para lograr esto, fue replicar todas las transformaciones y artefactos implementados en la preparación de los datos, al formulario de la aplicación, esto con el fin de mantener la coherencia entre los datos usados como entrenamiento y validación, y los inputs nuevos para obtener la predicción.

Finalmente, es válido resaltar que a medida que se continúen capturando datos del comportamiento de la morosidad, el modelo debe ser reentrenado, y las consideraciones mencionadas en el presente desarrollo deben ser interpretadas de nuevo, puesto que la sensibilidad del modelo a datos nuevos no es posible determinarla completamente. Un factor importante para esto, y fruto de trabajos posteriores, es el comportamiento de la morosidad en función del tiempo, así como sumarle las características propias del producto financiero adquirido y las características de la entidad bancaria.

## 5. Referencias

- [Credit Card Fraud Detection | Kaggle](#)
- Reporte de la situación de credito en Colombia  
<https://repositorio.banrep.gov.co/bitstream/item/827c82f9-87f1-4e32-9986-6b7a109cce8d/RSCC.pdf?sequence=1&isAllowed=y>
- Morosidad en Colombia <https://www.portafolio.co/economia/finanzas/aumento-morosidad-en-todos-los-creditos-menos-en-el-de-libranza-segun-transunion-558702>
- Detección de outliers multivariantes en python  
<https://towardsdatascience.com/multivariate-outlier-detection-in-python-e946fc843b3>
- El peligro de ingorar los valores faltantes  
<https://www.sciencedirect.com/science/article/abs/pii/S0169534708002772>
- Imputación con KNN <https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html>
- Random forest <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- Validación de modelos predictivos Cross-validation, OneLeaveOut, Bootstraping  
[https://www.cienciadedatos.net/documentos/30\\_cross-validation\\_oneleaveout\\_bootstrap](https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap)
- Algoritmos de clasificación <https://medium.com/datos-y-ciencia/aprendizaje-supervisado-introducci%C3%B3n-a-la-clasificaci%C3%B3n-y-principales-algoritmos-dadee99c9407#:~:text=Clasificaci%C3%B3n%20Binaria%3A%20Es%20un%20tipo,%E2%86%92%20etiquetado%20con%20un%200.>
- Pandas profiling <https://pypi.org/project/pandas-profiling/>