Aplicación de técnicas de Machine Learning para la predicción del riesgo de default de un cliente en una compañía de filipinas

Manuela Ramírez Quiceno
Andrés Medina Báez

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Tutor
Efraín Alberto Oviedo Carrascal, Magíster (MSc)

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2022

Cita	(Ramírez Quiceno & Medina Báez, 2022)
Referencia **Estilo APA 7 (2020)**	Ramírez Quiceno, M., & Medina Báez, A. (2022). *Aplicación de técnicas de Machine Learning para la predicción del riesgo de default de un cliente en una compañía de filipinas* [Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte III.

Tabla de contenidos

1. Resumen ejecutivo

En el presente trabajo se realizó un modelo predictivo cuya respuesta es la clasificación de que una vez una empresa financiera en filipinas, Home Credit, le otorgue un crédito hipotecario a un cliente, éste caiga en default (1) o no (0). En la herramienta Python, y basados en la metodología de trabajo CRISP-DM, inicialmente se realizó la exploración de los datos, conformado por 7 dataset y un total de 220 variables de orden sociodemográfico y del historial crediticio de cada cliente, tanto en Home Credit como en el sector externo. Posteriormente, se prepararon los datos mediante la eliminación de duplicados y de variables irrelevantes o redundantes, tratamiento de atípicos y de missings, codificación de variables categóricas, revisión de correlación, análisis univariable y bivariable, y balanceo de los datos debido a que las clases están desbalanceadas: Solo el 8% de 295.221 clientes pertenecen a la categoría 1, es decir, clientes que caen en default. Finalmente, empleando KBest de SKlearn, se seleccionan las 15 variables más relevantes a la hora de predecir el default del cliente, probando diferentes técnicas de Machine Learning como Decision Tree Classifier, Support Vector Machine, Naive Bayes, Random Forest Classifier, y Logistic Regression siendo este último el ganador para la métrica ROC_AUC de 0.71. La validez del modelo se logró ratificar mediante Cross Validation, con un KFold de 10, cuyos resultados para la métrica fueron de 0.70 y 0.0073 para la media y la desviación estándar respectivamente. Finalmente, en la evaluación final de la técnica ganadora con la data de prueba, el modelo predice los casos en default con un Recall del 64%. Los resultados muestran que el modelo obtenido a partir de los datos dispuestos tiene un desempeño aceptable a la hora de predecir el default de un cliente.

Link Repositorio en GitHub: https://github.com/AMedinaBaez/Monografia_riesgo_default

2. Descripción del problema

En un mundo cada vez más globalizado e interconectado, que ha sufrido profundos cambios socioeconómicos desde el siglo XX como consecuencia de las guerras mundiales, la gran depresión y la guerra fría, pues esto ha repercutido en que el sistema financiero busque cerrar las vulnerabilidades ante

las cuales está expuesta ya que hay una mayor exposición al riesgo de contagio. Es aquí entonces donde nace la necesidad de toda empresa en gestionar los diferentes riesgos ante los cuales está expuesta, sobre todo, las entidades financieras, ya que desde su actividad económica de manera intrínseca se enfrentan a un gran riesgo, el cual corresponde al riesgo crediticio (Castillo Rodríguez & Pérez Hernández, 2008).

Definiendo entonces el riesgo de crédito, según el Comité de Supervisión Bancaria de Basilea (1999), es "la posibilidad de que un prestatario bancario o una contraparte no cumpla con sus obligaciones de acuerdo con los términos acordados". Por esta razón, es muy importante cuando una entidad financiera debe de establecer su apetito de riesgo en función del rendimiento esperado, y para garantizar su cumplimiento no solo acuden a la definición de políticas de otorgamiento de cartera, sino también al desarrollo de modelos predictivos que permiten desde la solicitud de un cliente predecir la probabilidad del buen pago o del incumplimiento (default) de la obligación (Ossa & Jaramillo, 2021).

Esta predicción se ha abordado ampliamente en la literatura, sobre cómo han solventado este riesgo basado en la matemática y la estadística: pasando de la econometría, como una propuesta tradicional, hacia modelos de machine learning (ML) como nuevas propuestas disruptivas (Cuenca, 2019). Estos modelos, principalmente los de ML, desempeñan cada vez un papel más importante en la toma de decisiones para mejorar el riesgo crediticio, lo cual lo confirman estudios que apuntan que "*cerca del 60% de las grandes compañías del sector utilizan técnicas de analítica avanzada de datos para tomar decisiones en el ámbito de la gestión de riesgos. Cifras que seguirán aumentando los próximos años.*" (decide, 2019).

Teniendo en cuenta dicho contexto, es por ello que, Home Credit, empresa filipina de financiamiento, no es ajena a la necesidad de cuidar su riesgo de crédito. Sin embargo, ellos buscan desde su objetivo social garantizar a su vez la inclusión financiera, es decir: buscan ofrecer créditos a personas con antecedentes crediticios insuficientes o inexistentes (Kaggle, 2018), sin exponer por ello peligrosamente su apetito de riesgo.

2.1 Problema de negocio

Home Credit es una empresa filipina que busca ofrecer créditos a personas con antecedentes crediticios insuficientes o inexistentes, buscando la inclusión financiera de la población no bancarizada. Para ello, disponen a la comunidad de Inteligencia Artificial mediante la plataforma de Kaggle, siete (7) datasets sobre información de sus clientes con el fin de buscar un modelo predictivo que cumpla con su objetivo: Lograr la inclusión financiera cuidando el riesgo de crédito.

El cumplimiento de dicho objetivo se apalanca mediante la herramienta que supone el modelo predictivo, gracias a la marcación personalizada de la probabilidad de que un cliente caiga en incumplimiento, dadas sus características particulares de índole sociodemográfica y de historial crediticio.

2.2 Aproximación desde la analítica de datos

Home Credit, al igual que las diferentes entidades financieras alrededor del mundo, acuden a este pronóstico que soporta la toma de decisiones cuando de riesgo crediticio se trata. Dicha necesidad parte entonces desde la disciplina del Machine Learning (ML), definida como la ciencia que se enfoca en los datos y en diferentes algoritmos con el fin de aprender patrones, y cuyos resultados pueden mejorar continuamente de forma autónoma, simulando así el aprendizaje del ser humano (IBM, 2020).

Esta rama de la Inteligencia Artificial es muy flexible, característica que lo ha llevado a popularizar su uso en el sistema financiero, principalmente para los modelos de otorgamiento de crédito (Ossa & Jaramillo, 2021). Dentro de las variantes para predicción en el ML, se encuentra con objetivo de regresión o de clasificación, siendo esta última de interés para la presente monografía, pues permite identificar la categoría (o clase) a la que pertenece una muestra nueva, gracias a que ya aprendió de un conjunto de datos cuya categoría es conocida (Villagrá, 2015). Es decir, este objetivo está alineado con lo que busca.

Home Credit, pues requieren de un modelo predictivo que relacione cliente a cliente cuál es su probabilidad de que pertenezca a la clase 0 ó 1, siendo 0 la clase que indica un cliente al día y 1, un cliente que cae en **incumplimiento** una vez se le otorgó el crédito hipotecario.

Finalmente, la solución propuesta del presente problema lo valida la búsqueda en la literatura, donde este tipo de ejercicios es ampliamente abordado, como en Dastile, Celik y Potsane (2020), donde evalúan la solvencia crediticia de los prestatarios y resaltan en el proceso la popularidad de la regresión logística a la hora de abordar este tipo de problemas; o como en Cuenca (2019), donde predicen la buena o mala solvencia que puede tener un cliente mediante una regresión logística, resaltando que la importancia de este tipo de herramientas en el riesgo de crédito ayuda a optimizar el tiempo de evaluación de aprobación y desembolso de un cliente, como también en las provisiones, las cuales tienen un efecto directo en los estados financieros.

2.3 Origen de los datos

Los datos dispuestos por la compañía financiera Home Credit están disponibles en la plataforma de Kaggle en el link: https://www.kaggle.com/c/home-credit-default-risk/overview.

Dicha data contiene si el cliente cayó o no el default, a partir de una marca de 0 y 1, y contempla información sobre:

- El historial crediticio del cliente si tuvo otro producto financiero dentro de la compañía previa a la solicitud actual,
- El historial crediticio compartido por el buró financiero de Filipinas,
- Información sociodemográfica del cliente

Los datos contemplan información de 307.511 clientes identificados cada uno de ellos por un SK_ID_CURR único. Por parte de Home Credit, en la plataforma proporcionaron un total de 7 tablas correspondientes a datos estructurados no procesados, que requieren la aplicación de funciones de agregación, y acciones de limpieza y transformación con el fin de obtener un dataset con buena calidad de la data para pasar al proceso del modelado.

El contenido de la data se abordará con mayor amplitud en la sección 3 del presente informe.

2.4 Métricas de desempeño

La medición del desempeño de un modelo de clasificación predictiva se hace a partir de ciertas métricas que permiten evaluar la correcta discriminación entre las clases 0 (Clientes al día o sin incumplimiento) y 1 (Clientes en incumplimiento o default). Las más usadas corresponden al área ROC_AUC, Accuracy, Precisión, Recall y F1-Score.

- **ROC_AUC:** La curva ROC relaciona la sensibilidad con el porcentaje de falsos positivos, es decir, la tasa de acierto y la tasa de falsas alarmas. Se espera que el desempeño del modelo de clasificación se encuentre por encima de un clasificador aleatorio. Finalmente, la curva viene acompañada de la métrica AUC, la cual indica el área bajo la curva (Ossa & Jaramillo, 2021) y se espera que su valor sea cercano a 1.
- **Accuracy:** Indica la proporción de observaciones bien clasificadas. Sin embargo, en la literatura se recomienda usar esta métrica acompañada de otras cuando las clases son desbalanceadas, pues puede mostrar un resultado muy bueno cuando en realidad el modelo solo aprendió a clasificar la clase más frecuente (Borja & Monleon, 2020).
- **Precisión:** Indica la proporción de positivos que verdaderamente lo son. Esta métrica es ampliamente usada cuando el costo de los falsos positivos es elevado.
- **Recall:** Indica la proporción de casos positivos que fueron clasificados correctamente. En un modelo perfecto, se espera que esta métrica para cada clase tienda a 1 (Borja & Monleon, 2020). Esta métrica es ampliamente usada cuando el costo de los falsos negativos es elevado.
- **F1-Score:** Esta métrica es una mezcla entre el accuracy y el recall. Es usada en problemas de clases desbalanceadas, donde el "coste" de los falsos positivos y los falsos negativos es diferente (Abella & González, 2021)

Entonces, a partir de la definición de las diferentes métricas de desempeño a tener en cuenta para medir la capacidad discriminatoria del modelo, finalmente se identifica que para el presente problema se deberá revisar especialmente el ROC_AUC y el Recall.

1. **ROC_AUC**: Toda vez que es la métrica que Home Credit exige a la hora de evaluar el desempeño del modelo, y el

2. **Recall**: A la luz de lo que este reto supone, se identifica que el error que más le cuesta a Home Credit, y por lo tanto se debe de trasladar al modelo, son aquellos clientes que se marcan como 0, es decir, que no caerá en default, pero en realidad son un 1 (sí caen en default). Por esta razón, se busca minimizar el error correspondiente a los Falsos Negativos (FN), puesto que tiene un mayor costo en términos de capital que no retorna a la entidad prestadora de servicios financieros: capital por el cual Home Credit debe de provisionar y responder; vs a un falso positivo, que si bien es rechazar un cliente que no cae en default, pues el modelo captura que por las variables de entrada de dicho cliente, está más cercano a un default, por lo cual se está respetando intrínsecamente lo que la empresa está dispuesta a asumir en sus políticas de riesgo crediticio.

Por esta razón, se selecciona de manera complementaria al ROC_AUC, el Recall como las métricas ideales a la hora de seleccionar el modelo con el mejor desempeño.

Ahora bien, ya que se definió las métricas bajo las cuales se medirá las diferentes técnicas que se usarán para resolver el presente problema de clasificación, sin embargo, se debe profundizar de cara a Home Credit cuáles son los resultados esperados al aplicar este tipo de herramientas en la toma de decisiones.

Dentro del Riesgo Crediticio, definido anteriormente como el riesgo de que un prestatario incumpla con el pago de su obligación, existen tres (3) métricas muy usadas en toda entidad financiera para monitorear su estado actual frente a dicho riesgo. Estas tres (3) métricas son de acuerdo con el Banco BBVA, 2021:

La Tasa de Cobertura: Que son las provisiones acumuladas para dar respuesta a un posible impago derivado de un préstamo o crédito fallido (INCP, 2020)

El Coste de Riesgo: Asociado al costo anual de generar las provisiones.

y el índice de Mora: Que es una relación entre el volumen de clientes morosos y el tamaño de la cartera crediticia de la entidad. Este indicador particularmente cobrará una alta importancia para el presente problema, pues el objetivo de Home Credit es otorgar más créditos que permitan inclusión financiera, pero guardando un balance con la *salud* de su cartera.

Al ser un ejercicio con fines netamente académicos y con una data abierta al público, no se tiene acceso a la información del índice de mora actual y esperado de la entidad. Por esta razón, se pondrá como referencia la tasa actual de morosidad que reportan países de Europa y Asia, toda vez que Home Credit tiene presencia en los países de China, India, Indonesia, Vietnam, Filipinas, Rusia, Kazakstán, Estados Unidos de América, República Checa y Eslovaquia.

Desde la burbuja financiera del 2008, la tasa promedio de morosidad bancaria tocó niveles del 8%. Sin embargo, las entidades financieras año a año han hecho esfuerzos con el fin de disminuir dicha tasa y de esta forma ubicarse en un 4,3% promedio para el año 2022 (Crédito y Caución, 2022) (El Economista, 2022). Por esta razón, se esperaría que Home Credit al incluir la herramienta analítica propuesta en su proceso de otorgamiento de préstamos, pues esto les permita llegar a niveles similares a los que a hoy la industria financiera reporta, o por lo menos a una tasa de morosidad más baja que presentan actualmente.

Por esta razón, desde el ejercicio académico se sugiere no solo revisar el ROC_AUC y el RECALL como métricas desde la parte técnica que permitirán concluir la capacidad del modelo de reconocer efectivamente los clientes que pueden caer en incumplimiento, sino también tener presente el Índice de Mora actual vs la meta, pues se esperaría que pasados 90 días de una puesta en producción de este tipo de soluciones, se tenga una primera visual de si se está cumpliendo el objetivo de disminuir el deterioro de la cartera vencida de la entidad.

3. Datos

En los siguientes numerales se hará una descripción de los datos dispuestos en la plataforma de Kaggle por la empresa Home Credit, con el fin de profundizar en el entendimiento del problema a resolver.

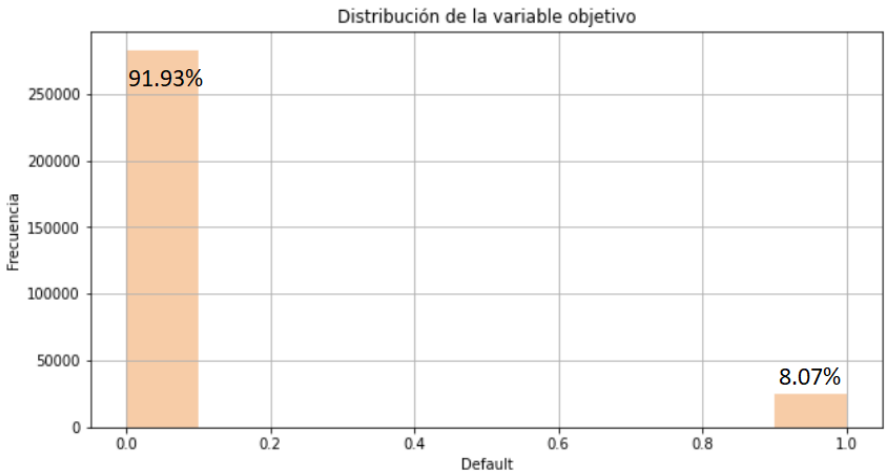
3.1 Datos originales

Para el reto propuesto por Home Credit, se contó con data estructurada distribuida en un total de siete (7) datasets.

1. **Application Train**: Contiene información principal de los clientes al momento de la solicitud del crédito hipotecario, asociados a un ID único. El tamaño de la tabla es de 307.511 instancias por 122 columnas. Dichas columnas traen información de las características del crédito solicitado, como también información propia del cliente como sus ingresos y egresos, otros créditos, edad, profesión, educación, residencia e información del comportamiento crediticio del círculo social del cliente. Este dataset trae la columna de TARGET, donde indica si el ID cayó en default (1) o no (0). En la Figura 1 se muestra la distribución de dicha variable, lo cual permite precisar que el reto a resolver tiene su clase objetivo desbalanceada.

Figura 1

Distribución de la variable Target.



Fuente: Elaboración propia.

2. **Previus Application**: Consolida información de los créditos que tuvo el cliente previamente en Home Credit. Contiene información del tipo de crédito que tuvo, los montos tanto de la aplicación como los desembolsados, la tasa de interés, como también el tipo de bien para el cual iba destinado el crédito: si era de índole de tecnología, electro-digital, medicinas, etc. Este dataset relaciona información de ID por ID del préstamo, por lo cual tiene un total de 1.670.214 instancias por 37 columnas.

3. **Installments Payments**: Detalla el comportamiento de pago que tuvo el cliente para los ID de préstamos relacionados en el dataset *Previos Application.* Tiene un total de 13.605.401 instancias por 8 columnas.

4. **Credit Card Balance**: Trae un detallado del comportamiento del cliente para el producto financiero correspondientes a créditos rotativos, es decir, tarjetas de crédito. Asocia por ID cliente y el ID de producto de tarjeta, el cupo, los saldos, montos mínimos de pago, entre otros. Tiene 3.840.312 instancias por 23 columnas.

5. **Pos Chash Balance**: Compuesta por 10.001.358 instancias por 8 columnas. Trae un detallado del comportamiento del cliente para el producto financiero de desembolso de dinero. Asocia por ID cliente y el ID de producto, el mes a mes del estado del préstamo, y un balance del comportamiento de pago.

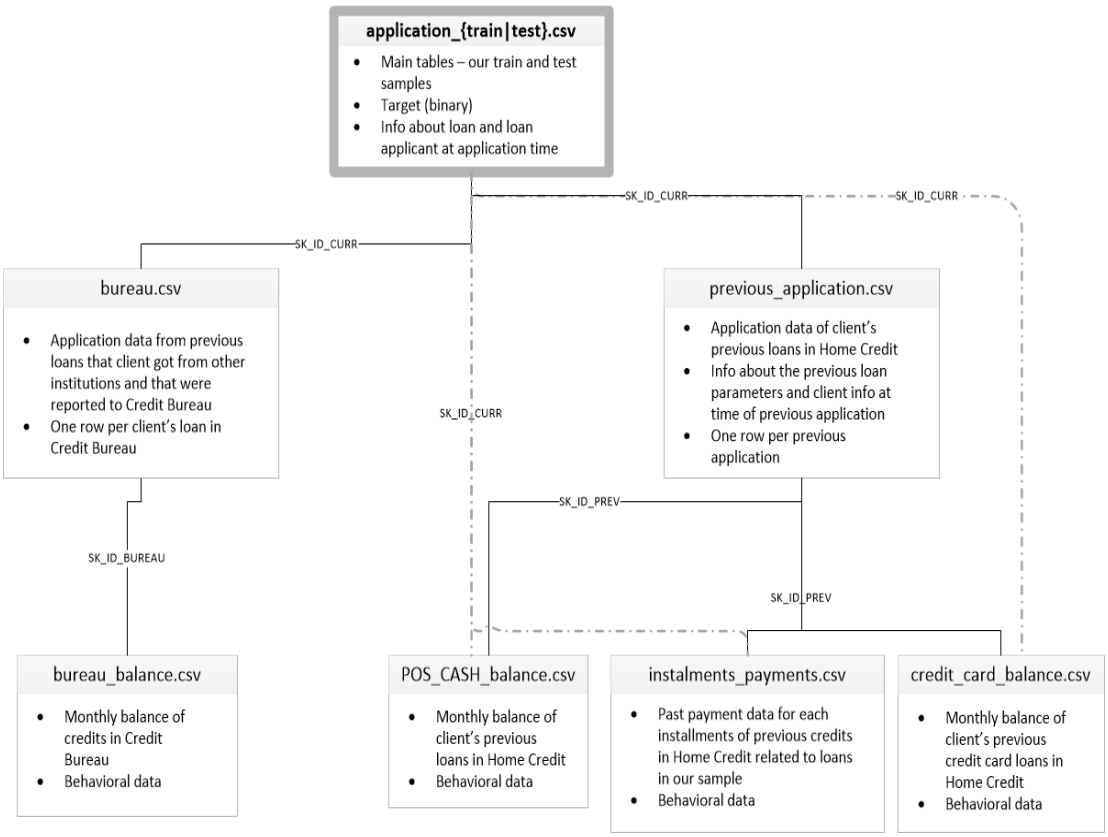
6. **Bureau**: Contiene la información proporcionada por un buró financiero del país de filipinas. Esta data es estructurada, cuyo contenido es de 1.716.428 instancias y 17 columnas. En sus columnas se describen otros créditos que el cliente ha tenido en el sector financiero, sin incluir Home Credit: Créditos activos, tipo de créditos, hace cuántos días el cliente solicitó dicho crédito, si el crédito fue prolongado, montos, entre otros.

7. **Bureau Balance**: Al igual que Bureau, éste dataset contiene la información proporcionada por el buró financiero, sin embargo, trae un desglose de cuál era el estado de cada crédito por mes, es decir: activo, cerrado, con alguna altura de mora. Tiene un total de 27.299.925 instancias y 3 columnas.

En este caso, se toma el histórico de 2 años hacía atrás por cliente y por crédito. En la Figura 2, se muestra la relación que tiene cada uno de los datsets descritos anteriormente.

Figura 2

Fuentes de datos.



Fuente: (Kaggle, 2018).

3.2 Datasets

A partir del entendimiento de cada columna por dataset, se realizó un resumen cliente a cliente de cada variable mediante diferentes funciones de agregación como suma, máximo, mínimo, y relaciones mediante raciones.

Un ejemplo de las funciones de agregación aplicadas fue en el dataset de credit_card_balance donde se obtiene una velocidad de pago del cliente a partir del promedio obtenido de AMT_INST_MIN_REGULARITY (cuota mínima de la tarjeta de crédito) sobre AMT_PAYMENT_CURRENT

(monto real pagado por el cliente). Esta variable obtenida es más poderosa a la hora de explicar el comportamiento real del pago del cliente vs al revisar de manera independiente las dos columnas fuente. (Ver Tabla 1).

Tabla 1

Ejemplo de función de Agregación.

SK ID CURR	AMT_INST_MIN_REGULARITY	AMT_PAYMENT_TOTAL_CURRENT	INDEX_PAY	INTERPRETACIÓN
100048	1.458	1.458	1	Cliente que paga a la misma velocidad que lo acordado
100188	18.000	1.294	13,908	Cliente que se está "colgando" en su deuda, pues su pago real está por debajo del pago mínimo exigido
100235	4500	410.715	0,011	Cliente prepagador, es decir: Paga más rápido que lo que pactó pagar

Elaboración: Fuente Propia

Por otra parte, para poder consolidar por cliente la información en un único registro, en el caso de las variables categóricas se les aplicó la función get_dummies disponible en la librería de Python, esto con el fin de realizar de igual forma operaciones de agregación. Un ejemplo de esta acción fue la realizada en la variable CREDIT_TYPE del dataset bureau. Los posibles valores que podía tomar esta variable son: Consumer credit, Credit card, Mortgage, Car loan, Microloan, Loan for working capital replenishment, Loan for business development, Real estate loan, Unknown type of loan, Another type of loan, Cash loan (non-earmarked), Loan for the purchase of equipment, Mobile operator loan, Interbank credit, Loan for purchase of shares (margin lending). A partir de la función get_dummies, se obtuvo un uno (1) de acuerdo con el tipo de préstamo al que perteneciera el ID del préstamo. Finalmente, para obtener un único registro por ID del cliente, se suman de manera independiente las nuevas 15 columnas. De esta manera, variables de tipo categórico pasaron a ser cuantitativas en los diferentes dataset.

Finalmente, el dataset obtenido para pasar a la fase de preprocesamiento está compuesto por diferentes columnas. Ver Tabla 2. El join entre dichos dataset lo permitió la columna llave de un SK ID único por cliente. Para los dataset que eran un balance mes a mes como Bureau Balance, éste traía consigo un

ID único de préstamo, que, al cruzarla con el dataset de Bureau, esto permitía obtener nuevamente el SK ID único por cliente y de esta forma hacer un merge frente a la tabla principal, correspondiente a application train.

Tabla 2

Filas x Columnas por DataSet.

DataSet	**Filas Después del Tratamiento**	**Columnas que aportó**
Application Train	307.511	119
Previus Application		52
Installments Payments		9
Credit Card Balance		14
Pos Cash Balance		12
Bureau		35
Bureau Balance		7
CANTIDAD FINAL	**307.511**	**248**

Fuente: Elaboración Propia

Uso de memoria por este dataset: **584.2MB**.

Para mayor detalle del proceso de colección de los datos y creación del dataset final, revisar el notebook 06-Carga_datos_Final.ipynb.

3.3 Descriptiva

El dataset final obtenido corresponde a data estructurada, donde por ID se relaciona información del cliente en aspectos sociodemográficos, y de su comportamiento con créditos de diferente índole, sea dentro de Home Credit o de otra empresa. Ver Figura 3.

Figura 3

Estructura del DataSet Final.

	Unnamed: 0	SK_ID_CURR	TARGET	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT_x	AMT_ANNUITY_x	DAYS_BIRTH	...	Active_y	Amortized debt
0	0	100002	1	0	1	0	202500.0	406597.5	24700.5	26.0	...	19.0	0.0
1	1	100003	0	0	0	0	270000.0	1293502.5	35698.5	46.0	...	9.0	0.0
2	2	100004	0	1	1	0	67500.0	135000.0	6750.0	52.0	...	0.0	0.0
3	3	100006	0	0	1	0	135000.0	312682.5	29686.5	52.0	...	18.0	0.0
4	4	100007	0	0	1	0	121500.0	513000.0	21865.5	55.0	...	25.0	0.0

5 rows × 249 columns

Fuente: Elaboración Propia.

De las 248 columnas finales, una vez eliminado “Unnamed: 0”:

- (1) SK_ID_CURR: Correspondiente a la identificación única por cliente.
- (1) TARGET: Indica la variable de clase, siendo 1 un cliente que cae en default y 0 un cliente que no.
- (188) Columnas que hacen referencia a variables numéricas, en formato int64 o float64
- (59) Columnas que hacen referencia a variables categóricas, en formato Category.

En la sección 4. PROCESO DE ANALÍTICA se entrará a mayor detalle de los hallazgos encontrados en la parte descriptiva de la información.

4. Proceso de analítica

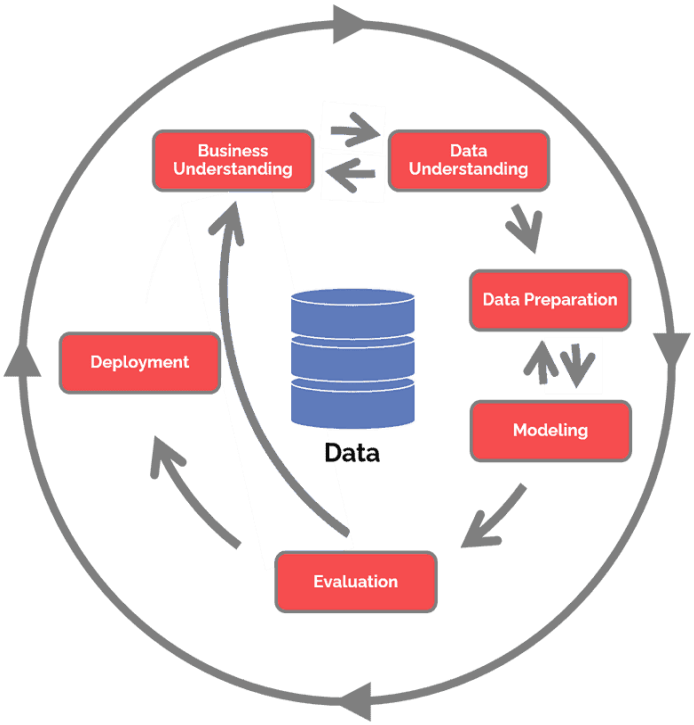
4.1 Pipeline principal

Este problema de negocio aborda el desarrollo de proyectos de minería de datos empleando técnicas supervisadas. Para aportar al desarrollo de esta aplicación financiera, se busca obtener predicciones que sean confiables, pues ésta constituirá una herramienta de apoyo en la toma de decisión si un cliente debe ser aceptado o rechazado, a partir de su probabilidad de caer en default una vez otorgado el producto.

Por lo tanto, se propone seguir la metodología de trabajo más impulsada en el momento por IBM, correspondiente a CRISP-DM (Cross Industry Standard Process for Data Mining) (IBM, n.d.), cuyo proceso se observa en la Figura 4.

Figura 4

Proceso CRISP-DM.



Fuente: (Data Science Process Alliance, s.f)

En las siguientes secciones del numeral 4 se hará zoom en cada una de las etapas.

4.2 Comprensión del negocio y de la data

4.2.1 Comprensión del negocio

Si bien en la sección 2. DESCRIPCIÓN DEL PROBLEMA se revisa a profundidad de la necesidad que surge desde la visión de negocio, antes de comenzar a abordar la solución propuesta, es necesario responder a las siguientes preguntas:

- **Cuál es el objetivo de negocio:** Home Credit busca cuidar su riesgo de crédito a través de la identificación anticipada de posibles clientes que caerán en un incumplimiento del pago de su préstamo hipotecario.

- **Evaluar la situación actual:** Como se mencionó anteriormente, Home Credit como entidad financiera de filipinas debe de cuidar su riesgo de crédito que puede ser medido mediante el índice de morosidad. Al ser una data abierta al público, no se tiene información al detalle del índice actual para la empresa, sin embargo, un buen punto de partido es establecer los puntos porcentuales que se busca disminuir respecto al estado actual, o por lo menos, definir que tan cerca o por debajo se desean situar frente al índice promedio que reportan para el sector financiero en el 2022, el cual ronda el 4,3% (sección 2.4).

- **Fijar los objetivos a nivel de minería de datos:** Se propone para alcanzar el objetivo de negocio obtener un modelo predictivo de clasificación que relacione cliente a cliente cuál es su probabilidad de que pertenezca a la clase 0 ó 1, siendo 0 la clase que indica un cliente al día y 1, un cliente que cae en incumplimiento una vez se le otorgó el crédito hipotecario. Se espera que dicho modelo en el ROC_AUC tenga un mejor desempeño que un predictor aleatorio, es decir, superior a 0.5.

- **Obtener un plan de proyecto:** El plan de proyecto para la consecución del objetivo a nivel de minería de datos será continuar con el paso a paso propuesto por la metodología CRIP-DM.

4.2.2 Comprensión de la data

Para el cumplimiento de esta fase se tuvo en cuenta:

- **Recolección de los datos**: En la sección 3. DATOS se expone la data que Home Credit pone a disposición en la plataforma de Kaggle, los retos enfrentados para consolidarla y descripción breve del contenido del dataset final.

- **Descripción de los datos:** El dataset final está compuesto por 307.511 instancias por 248 columnas, para un uso de memoria de 584.2MB.

- **Exploración de los datos:** Respecto a la variable objetivo, correspondiente a la columna TARGET, se identifica que solo el 8% de los clientes pertenecen a la clase de default o incumplimiento.

Para las variables explicativas, a continuación, se eligen 5 columnas con el fin de profundizar en la descripción de cada una de ellas. Ver Tabla 3 y Figura 5.

Tabla 3

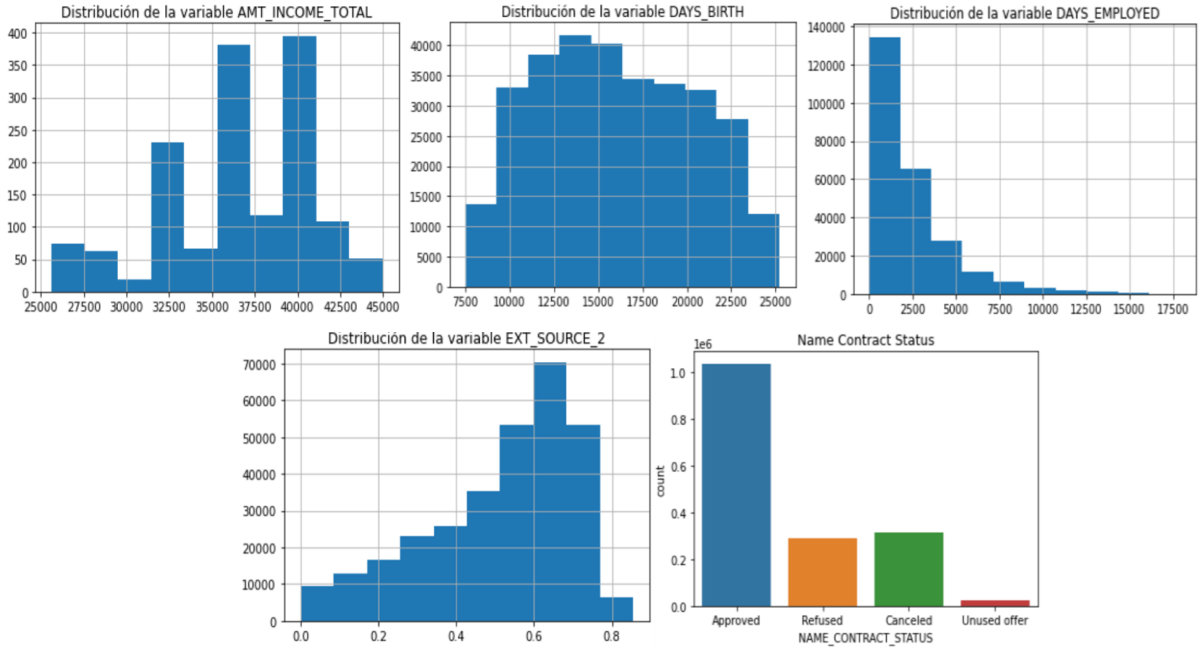
Ejemplo de variables categóricas y cuantitativas previas al preprocesamiento.

VARIABLE	**DESCRIPCIÓN**	**TYPE**	**MEDIA**	**DESVIACIÓN ESTÁNDAR**
AMT_INCOME _TOTAL	Corresponde a los ingresos del cliente	Float64	168797.9192	237123.1462
DAYS_BIRTH	Edad del cliente en días en el momento de la aplicación al préstamo hipotecario.	Float64	16036.99	4363.9886
DAYS_ EMPLOYED	Días transcurridos que el cliente lleva trabajando en el actual empleo en el momento de la aplicación del préstamo hipotecario.	int64	63815.0459	141275.7665
EXT_ SOURCE_2	Calificación tipo 2 que el sector externo le da al cliente.	Float64	0.5143	0.1910
VARIABLE	**DESCRIPCIÓN**	**TYPE**	**DISTRIBUCIÓN**	
NAME_CONTRACT _STATUS	Tipo de Contrato que el cliente presenta en aplicaciones pasadas en Home Credit	Category	Approved Canceled Refused Unused offer	1036781 316319 290678 26436

Elaboración: Fuente Propia

Figura 5

Distribución de 5 variables.



- Verificar calidad de los datos: En la exploración de los datos se identifica que la data requiere de preprocesamiento, toda vez que cuenta con variables con muy baja variabilidad, datos atípicos, y una importante calidad de missing. En la fase de PREPROCESAMIENTO se entrará en profundidad cómo se trató cada una de estas situaciones.

4.3 Preprocesamiento

Para un mayor detalle del preprocesamiento realizado, revisar el notebook:

07-Limpieza_Datos_62Cols.ipynb

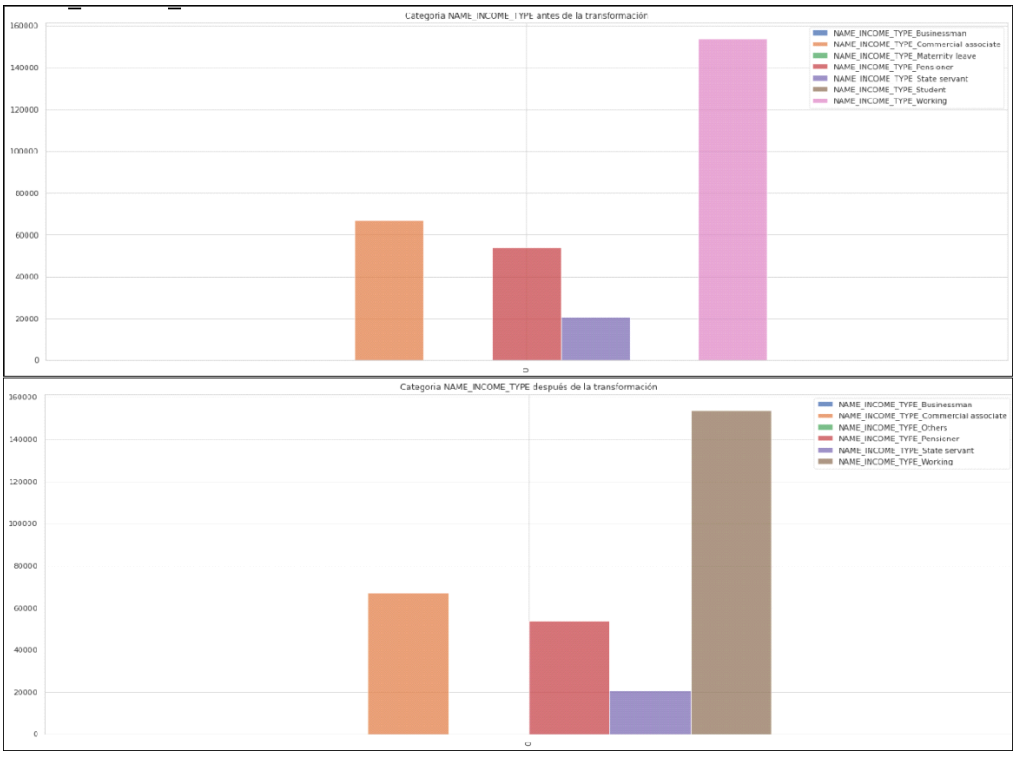
4.3.1 Transformaciones

Se transforman las variables que Python por defecto lee en formato int64 o float64 por Category, mediante la función .astype(). De acá se obtiene el dataset a trabajar compuesto por 58 variables categóricas y 188 numéricas.

Hay tres (3) variables categóricas que tienen más de 7 etiquetas, como por ejemplo se observa en la Figura 6 para la variable Name_Income_Type. Por ello, se realizó entonces una agrupación en “OTROS”, que son la suma de aquellas etiquetas que presentaban una baja tasa de eventos, garantizando que esta nueva categoría no tomará mayor relevancia vs aquellas que por sí solas ya tenían una tasa de eventos importante.

Figura 6

Transformación en variables Categóricas.



Fuente: Elaboración Propia.

4.3.2 Eliminar duplicados y variables redundantes

- No hay presencia de datos duplicados por ID, lo cual se validó mediante la función .duplicated().
- Por otra parte, se identifica las siguientes variables redundantes, las cuales se procedieron a hacerles un .drop(): NUM_INSTALMENT_VERSION, NUM_INSTALMENT_NUMBER, DAYS_INSTALMENT, DAYS_ENTRY_PAYMENT, AMT_INSTALMENT, AMT_PAYMENT, toda vez que éstas fueron empleadas para el cálculo de nuevas columnas más dicientes, como índices de velocidad de las cuotas del crédito, velocidad de pago, velocidad en el pago total de la deuda, y como index_pay que se describe en la sección 3.2.

4.3.3 Tratamiento de nulos

Primero, se revisó el porcentaje de instancias nulas por columna, donde se identificó que 90 de las columnas (Ver Figura 7) presentaban un porcentaje superior al 15% de datos nulos, por lo cual se tomó la decisión de realizar un drop de dichas variables.

Figura 7

Muestra de columnas con su % de datos nulos.

```
missing= pd.DataFrame(list(zip(x,y)), columns = ['feature','missing'])
missing[missing['missing']>15]
```

	feature	missing
12	EXT_SOURCE_1	56.3811
14	EXT_SOURCE_3	19.8253
15	APARTMENTS_AVG	50.7497
16	BASEMENTAREA_AVG	58.5160
17	YEARS_BEGINEXPLUATATION_AVG	48.7810
...	...	...
227	Demand_y	20.1869
228	Returned to the store	20.1869
229	Signed_y	20.1869
230	MAX_SK_DPD_y	45.9889
231	MAX_SK_DPD_DEF	45.9889

90 rows × 2 columns

Fuente: Elaboración Propia.

A partir de esta acción, se pasó de 18.136.837 valores nulos a 1.954.325, para una reducción del 89.22% en la cantidad de valores nulos.

Posteriormente, se identificó que la variable que asocia la anualidad del crédito presenta una alta variabilidad, por lo cual se reemplazó sus valores nulos por la mediana. Para el resto de las variables que aún presentaban nulos, se realizó el reemplazo por la media.

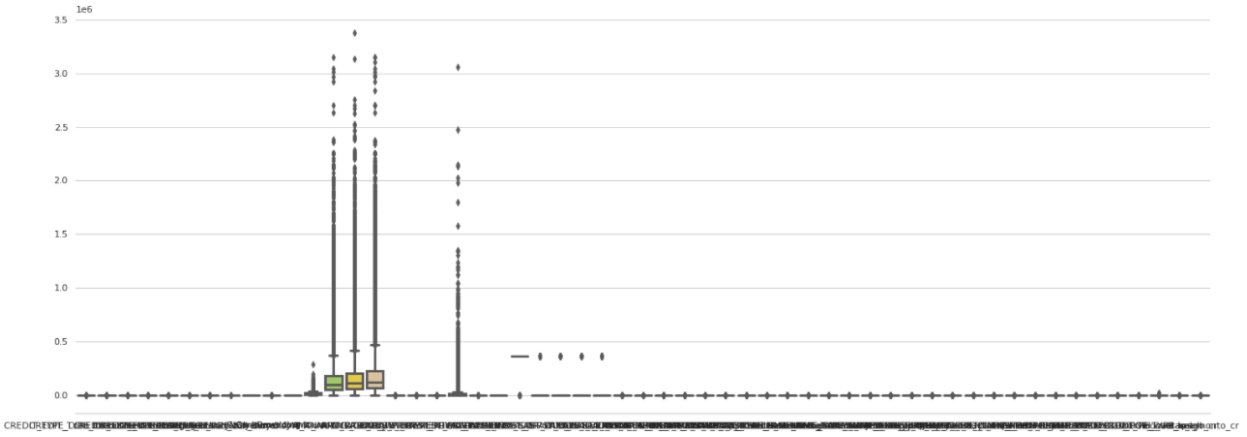
4.3.4 Tratamiento de valores atípicos

Primero, se realiza el drop de todos los valores que sean inferiores al percentil 1 y superiores al percentil 99 por columna, eliminando así un total de 6.121 instancias respecto al dataset inicial.

Posteriormente, mediante la librería de seaborn, se realizó el gráfico de Caja y Bigotes, toda vez que con el bigote fácilmente se logra identificar datos atípicos, como también la diferencia de magnitud entre las variables (Ver Figura 8).

Figura 8

Caja y Bigotes para Datos Atípicos.



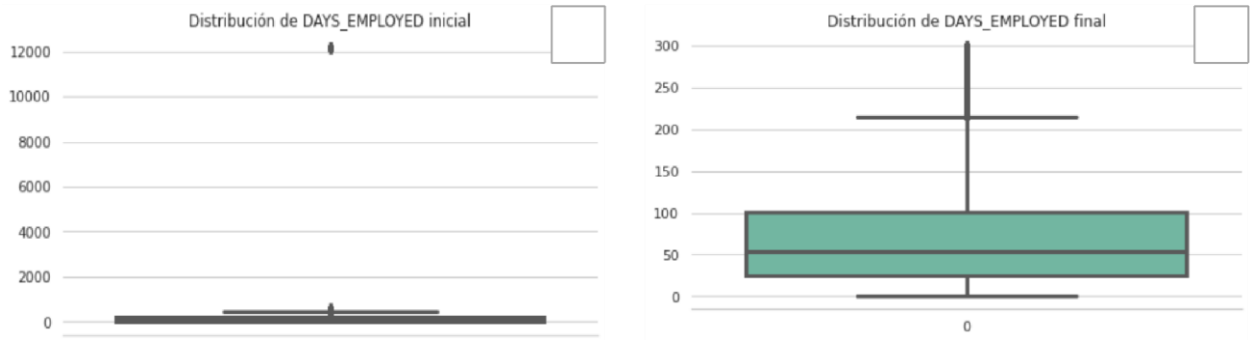
Fuente: Elaboración Propia

Un claro ejemplo del tratamiento de atípicos realizado es para la variable DAYS_EMPLOYED. El primer tratamiento fue pasar de días a meses dicha variable. Posteriormente, al revisar su distribución se encuentra que hay clientes que registran 12 mil meses empleados, lo cual corresponde a un error ya que esto equivale a 1.000 años laborando (Ver Figura 9 - 1). El tratamiento que se le da a esta variable es

reemplazar todos aquellos valores superiores a 300 meses por 300, lo cual corresponde a 25 años aproximadamente en un mismo puesto de trabajo. Este valor se obtiene a partir de revisar la distribución de la variable, donde todos los valores a partir del percentil 80 son reemplazados con impacto en términos de clientes del 20% del total de la base (Ver Figura 9 - 2).

Figura 9

Tratamiento de Atípicos en DAYS_EMPLOYED.



Elaboración: Fuente propia.

A partir del tratamiento de datos atípicos, se obtuvo una reducción de -3.98% en el número de muestras.

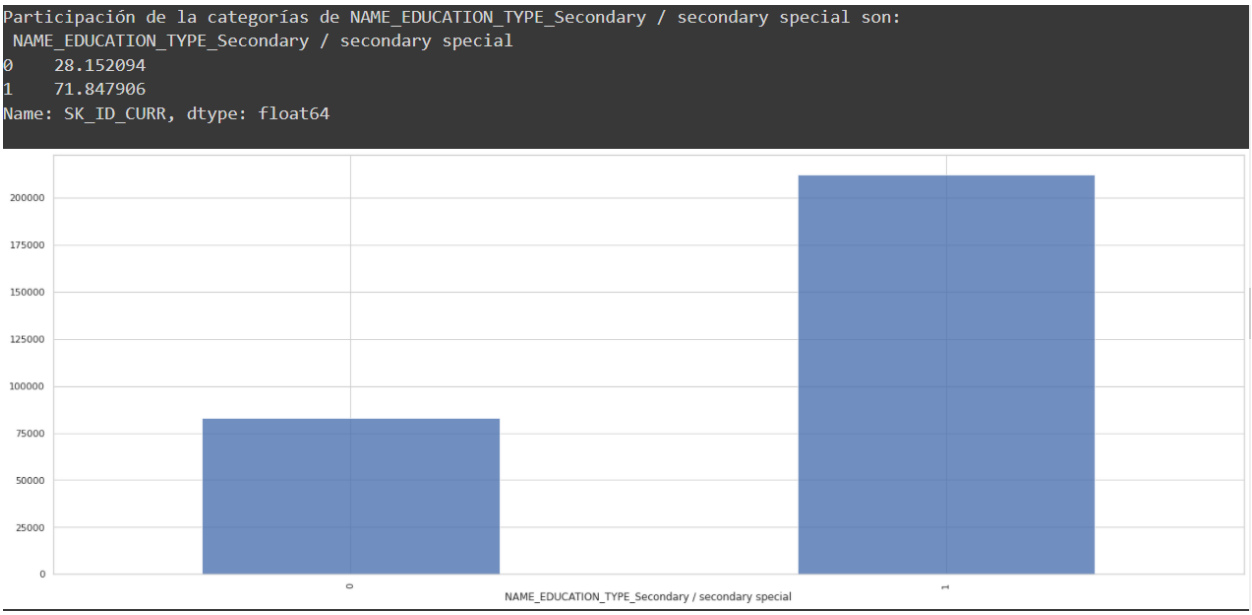
4.3.5 Exploración univariable y bivariable

- CATEGORICAS

Se identifica la distribución que presenta cada variable categórica, según los posibles valores que puede tomar ésta. En la Figura 10 se muestra el caso de la columna NAME_EDUCATION_TYPE_Secondary/secondary special, donde 0 son clientes que no tienen este nivel de escolarización, correspondiente al 28% del total de la población, y 1 personas que sí, correspondiente a un 72%. En una exploración univariable esto da un buen indicio que para esta variable en particular la técnica a utilizar aprenderá de ambos valores posibles en nivel educativo de secundaria.

Figura 10

Distribución univariable para Name education type Secundary



Fuente: Elaboración Propia

Posteriormente, se realiza un análisis bivariable, es decir, se valida la hipótesis de si cada una de las variables categóricas es importante para la predicción de nuestras clases 0 y 1 mediante una prueba de chi-cuadrado. Esto es posible gracias a stats y chi2_contingency disponible en scipy.stats en Python, que con un nivel de confianza del 95% y mediante una tabla de contingencia se realiza el test de independencia, donde la hipótesis nula (H0) corresponde a que La variable TARGET es independiente de la variable explicativa, o la hipótesis nula (H1): La variable TARGET tiene un grado de asociación o relación frente a la variable explicativa.

Bajo esta prueba, se detectan un total de 4 variables que se comprueba la H0, es decir, no son relevantes a la hora de predecir nuestro TARGET. Dichas variables corresponden a ['NAME_HOUSING_TYPE_Co-op apartment', 'HOUSETYPE_MODE_terraced house', 'NAME_HOUSING_TYPE_Municipal apartment', 'NAME_INCOME_TYPE_Unemployed']. Finalmente, se realiza un reagrupamiento de las variables Occupation_type, wallsmaterial_mode y name_income_type

con el fin de eliminar esas posibles etiquetas dentro de la variable categórica con baja participación, pues posiblemente la técnica de ML no aprenderá información relevante de dicha variable.

- CUANTITATIVA

Primero, se realiza una cuantificación de las variables que presentan una desviación estándar inferior a 0.05: 'CREDIT_ACTIVE_Bad debt', 'CREDIT_CURRENCY_currency 3', 'CREDIT_CURRENCY_currency 4', 'RATE_INTEREST_PRIMARY', 'NAME_PORTFOLIO_Cars', 'ind_vel_tpago_cr', las cuales fueron eliminadas del set de variables explicativas ya que por su baja variabilidad, nuevamente nos enfrentamos al reto de que la técnica de ML que se use posiblemente no aprenderá información relevante de éstas variable.

Finalmente, se revisa la correlación presente entre las variables explicativas, como también de dichas variables frente al target. La regla empleada para evitar problemas de multicolinealidad entre variables explicativas fue eliminar las correlaciones menores a -0.7 o superiores a 0.7. Bajo esta regla, hay una reducción de únicamente 20 variables numéricas entre las explicativas. A continuación, en la Tabla 4 se hace un zoom de algunas variables con correlación importante, y la descripción de la decisión tomada.

Tabla 4

Análisis de Correlación entre variables explicativas.

VARIABLE 1	**DESCRIPCIÓN V1**	**VARIABLE 2**	**DESCRIPCIÓN V2**	**CORRELACIÓN**	**DESCRIPCIÓN PROCESO**
AMT_ CREDIT_x	Monto del crédito del préstamo	AMT_ ANNUITY_x	Anualidad préstamo	0,7728	Se elimina el monto de la anualidad, ya que el monto del crédito refleja lo grande/pequeño que era el

					crédito con mayor precisión
REGION_ RATING _CLIENT	Nuestra calificación de la región donde vive el cliente	REGION_RA TING_ CLIENT_W_C ITY	Nuestra calificación de la región donde vive el cliente teniendo en cuenta la ciudad	0,9506	Se toma la calificación de la región donde vive el cliente y que tiene en cuenta la ciudad, ya que va más allá
OBS_30_C NT_ SOCIAL_CI RCLE	Cuantas observaciones del entorno social del cliente con incumplimiento observable de 30 DPD (días vencidos)	OBS_60_CN T _SOCIAL_CIR CLE	Cuantas observaciones del entorno social del cliente con incumplimiento observable de 60 DPD (días vencidos)	0,9986	Es natural que una persona que rueda a mora60, tuvo que pasar primero por mora 60. Ambas variables son importantes, por lo cual, no se elimina ninguna de las dos
AMT_CRE DIT _SUM	Monto de crédito actual para el crédito de Buró de Crédito	AMT_CREDI T _SUM_DEBT	Deuda vigente en crédito Buró de Crédito	0,7103	Ambas variables se le hacen drop toda vez que se ven reflejadas en la columna calculada: indice_falta_pagar
CREDIT_AC TIVE _Active	num de de los créditos activos reportados del Buró de Crédito	CREDIT_TYP E _Creditcard	Num de creditos en tarjetas credito	0,7327	Se toma el número de créditos activos reportados del Buró, ya que dentro de ellas contiene el nro que

					corresponde a tarjetas de crédito.

Elaboración: Fuente Propia

Por otra parte, para el caso de las variables frente el target, se logra concluir que no hay una correlación lineal fuerte entre las variables explicativas y la variable a predecir, pues la correlación más alta obtenida en valor absoluto es de 0,159260136, correspondiente a la variable EXT_SOURCE_2, que es un score que le otorga el cliente el sector externo. Por esta razón, se hace un análisis para identificar a partir de cuántas variables Xi, ordenadas de la correlación mayor a menor con la variable TARGET, dejan de aportar una correlación significativa. Ver Figura 11.

Figura 11

Cambio en el valor promedio de la correlación.

Fuente: Elaboración Propia

Del gráfico anterior, surgen 2 posibles dataset para el entrenamiento final:

- El primero, para un punto de corte de correlación en valor absoluto superior a 0,005 (Punto2 en la Figura 11), para un total de 82 variables. (08-Limpieza_Datos_82Cols.ipynb)

- El segundo, para un punto de corte de correlación en valor absoluto superior a 0.017 (Punto1 en la Figura 11), para un total de 62 variables. (11-Limpieza_Datos_62cols.ipynb).

En el numeral 5 de la presente monografía se profundizará sobre cuál fue el dataset final elegido para entrenar.

4.3.6 Revisión de duplicados

Se realiza una revisión nuevamente de posibles instancias duplicadas mediante. duplicated().sum() disponible en la librería de pandas. El resultado de 0 indica que no hay presencia de registros completamente idénticos una vez se elimina el ID único, lo cual favorece al dataset finalmente obtenido, pues evitará que las técnicas a usar para la modelación "aprenderán" más sobre cierto grupo de clientes.

4.3.7 Verificación final del procesamiento

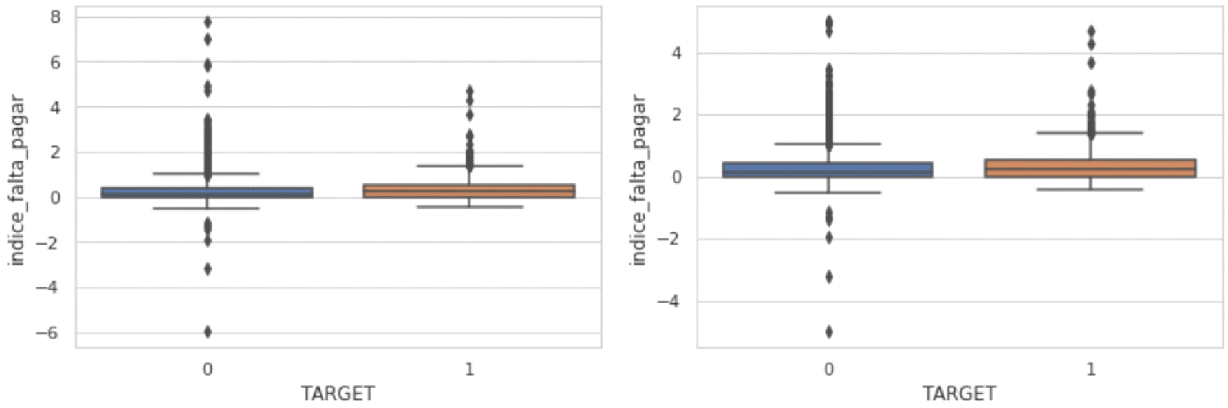
Para revisar con mayor profundidad el procedimiento llevado a cabo en este numeral, consultar el notebook: 14-Preprocesamiento_Data_Elegida.ipynb.

Para el dataset obtenido en 4.3.5, se realiza nuevamente una revisión de las variables con el fin de garantizar la eliminación o reemplazo de atípicos, evitar datos nulos y altamente correlacionados.

Entre las transformaciones realizadas, está la aplicada a la variable "Indice_falta_pagar" (Ver Figura 12), donde se redujo la variabilidad que estaba presentando dicha variable a partir de reemplazar los valores atípicos de las colas por valores de -5 o 5.

Figura 12

Cambio en la variable indice_falta_pagar.



Elaboración: Fuente Propia

Una vez realizada la etapa de preprocesamiento, el dataset obtenido finalmente es de 295.221 instancias por 60 columnas. El uso de memoria por este dataset: 74.3 MB, para una reducción de 509.9MB respecto al dataset inicial descrito en 3.2.

4.4 Modelos

Teniendo en cuenta el reto actual, se abordaron cinco (5) técnicas para poner a competir. Todas las técnicas usadas están disponibles en la librería de sklearn en Python. A continuación, una leve descripción de cada una de ellas.

- Regresión Logística. Como se describió en el numeral 2 de la descripción del problema, en la literatura y bajo la práctica, de los modelos que han dado mejores desempeños históricamente a la hora de responder a problemas de riesgo de crédito es la regresión logística.

En Python: **from** sklearn.linear_model **import** LogisticRegression

- Árbol de Regresión de Clasificación. Si bien es una técnica poco robusta y sofisticada, es de las más empleadas cuando la predicción requiere ser de alta interpretabilidad, aspecto que es altamente demandado en el sector financiero al ser supervisados por entes regulatorios. Uno de los retos más grandes a enfrentar con esta técnica es que es vulnerable cuando la clase a predecir está desbalanceada.

En Python: **from** sklearn.tree **import** DecisionTreeClassifier

- Naive Bayes: Bajo la técnica GaussianNB, esta técnica basada en probabilidades se usa toda vez que popularmente se concibe sus resultados como “buenos” frente a problemas donde el target presenta desbalanceo en sus clases.

En Python: **from** sklearn.naive_bayes **import** GaussianNB

- Support Vector Machine (SVM): Técnica robusta que se incluye teniendo en cuenta la alta dimensionalidad que presentan los dos datasets obtenidos y descritos en la sección 4.

En Python: **from** sklearn.svm **import** SVC

- Random Forest: Técnica de ensamble, donde se busca que los n árboles capturen una relación levemente distinta durante el proceso de bagging, posibilitando a la técnica aprender de la clase minoritaria, en este caso, la clase 1 (cae en default/incumplimiento).

En Python: **from** sklearn.ensemble **import** RandomForestClassifier

4.5 Evaluación

Como se precisó en la sección 2.4, para la medición del desempeño de este modelo de clasificación se hará mediante:

- **ROC_AUC.**

En Python: from sklearn.metrics **import** roc_curve, auc. Permitirá dar juicio sobre la capacidad del modelo de distinguir entre las 2 clases que se están buscando predecir.

- **RECALL.**

En Python: from **sklearn.metrics** import **classification_report.** Mediante la Matriz de Confusión se podrá obtener la ratio de tp: de true positive y fn: false negative, y de esta manera obtener tp/(tp+fn), que será la métrica final que dará indicios sobre la capacidad del modelo de clasificar bien aquellos clientes que caen en incumplimiento.

4.6 Despliegue

La solución propuesta para este reto no contará con el paso de despliegue, toda vez que su abordaje y resultados son únicamente con fines educativos. Sin embargo, en la sección 6.3 se detallan sugerencias a la hora de implementar este tipo de soluciones analíticas.

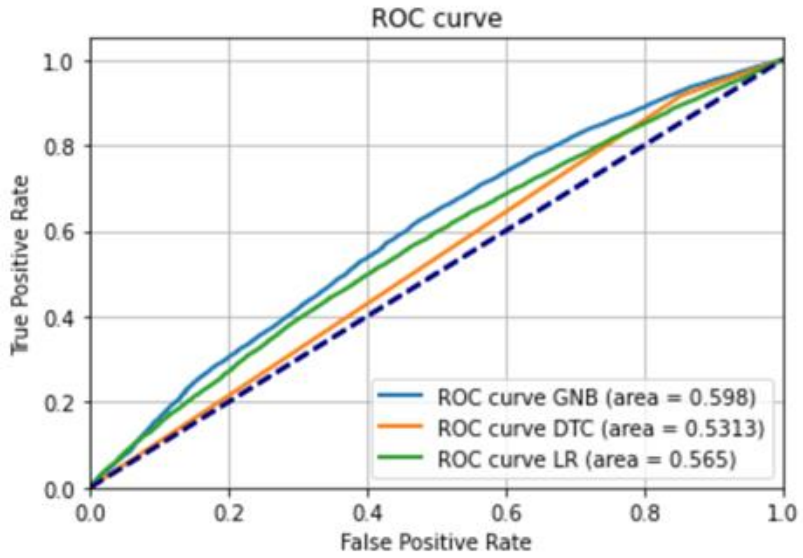
5. Metodología

5.1 Baseline

La primera iteración de este reto se realizó con tres de los modelos descritos en la sección 4.4, correspondientes a la regresión logística, árbol de decisión de clasificación y Naive bayes, bajo los hiperparametros que vienen por default en Sklearn. El resultado de la métrica ROC_AUC en general fue muy bajo, pues los tres (3) resultados tienden a 0.50 como un modelo aleatorio (Ver Figura 13).

Figura 13

ROC_AUC para la primera iteración.



Fuente: Elaboración Propia.

Por otra parte, al revisar el desempeño en las métricas de Accuracy Train vs Test, se identifica que la técnica de la regresión logística y naive bayes presentan un buen desempeño, mientras que el árbol de decisión se identifica un posible sobreajuste (Ver Tabla 5)

Tabla 5

Resultados de la Iteración #1.

TÉCNICA	**ACCURACY TRAIN**	**ACCURACY TEST**	**RECALL**
Decision Tree Classifier	1	0.8542	0.8532
Logistic Regresion	0.9199	0.9199	0.9999
Naive Bayes	0.918	0.9179	0.9924

Elaboración: Fuente Propia

A partir de los resultados obtenidos, se identifica las siguientes acciones de mejora con el fin de encontrar una técnica con mejor desempeño:

1. Teniendo en cuenta la alta dimensionalidad del dataset original, se propone emplear técnicas que presentan un mejor desempeño con alta dimensionalidad, como el support vector machine.
2. Probar el escalamiento de las variables, sea mediante el método MinMax o estandarizado.
3. Realizar balanceo de la variable TARGET, toda vez nuestra variable objetivo presenta desbalance.
4. Realizar optimización de hiperparametros, buscando maximizar los resultados de la métrica ROC_AUC.

5. Realizar Cross Validation de los 2 modelos que estén presentando mejores resultados, con el fin de validar finalmente la consistencia del modelo al predecir el incumplimiento o default del cliente.

5.2 Validación

Como se mencionó en la sección 5.1, es importante realizar una comparación entre las métricas obtenidas durante el entrenamiento, con el conjunto de train, vs las obtenidas en el test, pues esto dará indicios de la capacidad discriminante del modelo de clasificar correctamente nuevos clientes para los cuales no tiene la respuesta de que cayó o no en incumplimiento.

Para esto entonces se empleó la técnica de Split 70/30, llamando a un muestreo estratificado (Ver Figura 14), lo cual responde a la necesidad de que el conjunto de train como de test cuenten con registros que pertenezcan a la clase 1; clase que solo representa un 8% del dataset total como se mostró en la sección 3.1.

Figura 14

Split 70/30.

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3,random_state =42, stratify=y)
```

Fuente: Elaboración Propia.

Finalmente, con el fin de revisar los resultados que está teniendo el modelo desde el mismo momento de entrenamiento, se empleó la técnica de cross validation en dos momentos:

1. Durante el tuning de hiperparametros, mediante GridSearchCV disponible en scikit-learn. El CV estándar elegido fue de 5.

2. Cross Validation de los dos últimos modelos con el mejor desempeño en la métrica ROC_AUC, mediante StratifiedKfold disponible en model_selection de sklearn. El número de splits elegidos para esta validación fue de 10.

5.3 Iteraciones y evolución

Con el objetivo de mejorar el desempeño obtenido en la iteración 1 descrita en 5.1, se realizaron un total de 8 interacciones:

1. **Modelo con 82 variables:** Se vuelve hasta el paso de consolidación del dataset final, evitando eliminar variables que no sean bajo un argumento matemático-estadístico. Para esta corrida se presta especial atención a aquellas variables categóricas que presenta 7 o más levels dentro de la misma, por lo cual requirió hacer transformaciones en 3 de ellas como se mostró en la Figura 6. Para el paso de modelación, se conservó las mismas tres (3) técnicas: Regresión Logística, Árbol de Decisión y Naive Bayes. ROC_AUC promedio: 0.58. Ver Figura 13.

2. **Modelo con 62 columnas:** Este dataset nace a partir del punto 1 de inflexión identificado en la correlación de acuerdo con la Figura 11. Para este dataset se prueba con dos nuevas técnicas: Support Vector Machine (SVM) y Random Forest. El desempeño promedio de acuerdo con el ROC_AUC es de 0.62, menos para el árbol que presenta un valor de 0.53.

Uno de los retos más importantes durante esta iteración fue en la técnica de SVM, ya que, gracias a la alta dimensionalidad del dataset, sumado a la gran cantidad de registros, la técnica no logró converger hacia un resultado aceptable en ROC_AUC sin que esto implicara gran cantidad de recursos computacionales que excedía la capacidad ofrecida por la herramienta de modelamiento en la cual se profundiza en la sección 5.4.

3. **Modelo con 60 columnas, datos escalados:** Se prueban nuevamente las 5 técnicas que se emplearon en la iteración anterior, pero realizando un escalamiento de las variables numéricas mediante la utilidad scale, disponible en preprocessing de la librería sklearn. No se identifica una mejoría sustancial al realizar este preprocesamiento en la data. Ver Tabla 6.

Tabla 6

Comparación de resultados con y sin Scaler.

TÉCNICA	ROC_AUC SIN SCALER	ROC_AUC CON SCALER
Decision Tree Classifier	0.5382	0.5105
Logistic Regresion	0.6121	0.6393
Naive Bayes	0.6352	0.6137
SVM	-	0.5508
Random Forest	0.6376	0.5508

Fuente: Elaboración Propia

4. **Modelo con 15 columnas desbalanceado**: A partir de la presente iteración y la siguientes, se realiza con un dataset de 15 variables predictoras. Las 15 variables fueron seleccionadas mediante SelectKBest y f_classif, disponible en feature_selection de Sklearn (Ver Figura 15). Este método permite mediante una prueba Anova, seleccionar las variables predictoras más relevantes a partir de la puntuación más alta k (Scikit-learn, 2022), que para el presente reto se eligió k=15. Al emplear SelectKBest se logró eliminar aquellas variables menos importantes y reducir por lo tanto el tiempo de entrenamiento de cada técnica.

Figura 15

Resultado de Select KBest.

Fuente: Elaboración Propia.

Para revisar con mayor profundidad la descripción de las variables, consultar el notebook: 01-Carga_datos_1ra_Iter.ipynb.

Los resultados obtenidos para las 5 técnicas aún para una configuración de hiperparametros por default, la Regresión Logística comienza a debutar con un desempeño en la métrica ROC_AUC de 0.7143, seguido por Naive Bayes en 0.6862 y el Random Forest en 0.6266, que después de la optimización de hiperparametros logró subir a 0.6986

5. **Modelo con 15 columnas, igualando el desbalance:** Uno de los desafíos del presente reto corresponde al profundo desbalance presente dentro la variable objetivo. Por esta razón se realiza las últimas dos iteraciones realizando un balanceo de la data:

- Balanceo por encima, donde se crearon datos sintéticos de la clase 1 (cae en incumplimiento), con el fin de igualar la cantidad de la clase 0. Este procedimiento representó pasar de un dataset de train de 206.655 registros a 379.932, siendo 173.277 los nuevos registros. Para este balanceo se empleó SMOTE, disponible en over_sampling de la librería imlearn.
- Balanceo por debajo, donde la clase 0 del conjunto de train se redujo hasta igualar la cantidad de registros presentes de la clase 1, correspondiente 16.688. Este procedimiento representó una

reducción de la data de entrenamiento en un 88,69%. Para este balanceo se empleó RandomUnderSampler(), disponible en under_sampling de la librería imlearn.

Para el balanceo por encima y por debajo se prueban las cinco (5) técnicas, para las cuales se realiza la optimización de hiperparametros con el fin de encontrar la combinación óptima para cada técnica que permita maximizar la métrica ROC_AUC. Los resultados obtenidos se disponen en la Tabla 7.

Tabla 7

Resultados de las iteraciones finales.

TÉCNICA	**SIN BALANCEO**	**BALANCEO POR ENCIMA**	**BALANCEO POR DEBAJO**
Decision Tree Classifier	N/A	0.6268	0.6851
Logistic Regresion	0.7143	0.6648	0.7142
Naive Bayes	0.6862	0.5966	0.6855
SVM	0.4105	0.4539	0.4322
Random Forest	0.6986	0.6646	0.7009

Fuente: Elaboración Propia.

De estas últimas iteraciones se rescata lo siguiente:

- El desempeño del árbol de decisión mejoró significativamente al realizar el balanceo de la data, sea por debajo o por encima.
- La técnica de SVM no logra converger, toda vez que el el hiper parámetro de max_iter se configuró en 2.000. No se logra una mayor cantidad de iteraciones sin que la herramienta solicite una mayor cantidad de recursos computacionales.
- El balanceo por debajo muestra un mejor desempeño que el balanceo por encima, lo cual se le atribuye a la gran cantidad de registros sintéticos creados.
- Los dos modelos que pasarán a la última etapa de Cross Validation y de Generalización de la técnica son los recuadros naranjas en la Tabla 7, correspondientes a la Regresión Logística y el Random Forest. La configuración ganadora para cada técnica se mostrará en mayor detalle en la sección 6.1.

5.4 Herramientas

Para el presente proyecto se empleó la herramienta Python, el cual es un lenguaje de programación que dentro de su abanico de posibilidades, hay un gran espectro de desarrollos ya adelantados dentro del mundo del Machine Learning. De los entornos donde se puede ejecutar Python, se usó Jupyter Notebook, el cual permite crear código, ejecutarlo y visualizar permanentemente dicho resultado. Estos Notebooks fueron creados en Colaboratoy o “Colab”, el cual es un producto de Google Research. Cabe resaltar que se utilizó el colab sin coste adicional, por lo cual se estuvo acotado a unos recursos computacionales limitados y no garantizados (Google, 2022).

Las librerías más utilizadas durante el desarrollo del proyecto fueron: Numpy, Pandas, os, matplotlib.pyplot, Seaborn, plotly, scipy.stats, sklearn.model_selection con train_test_split, sklearn con model_selection, sklearn.metrics, imblearn, entre otros.

6. Resultados

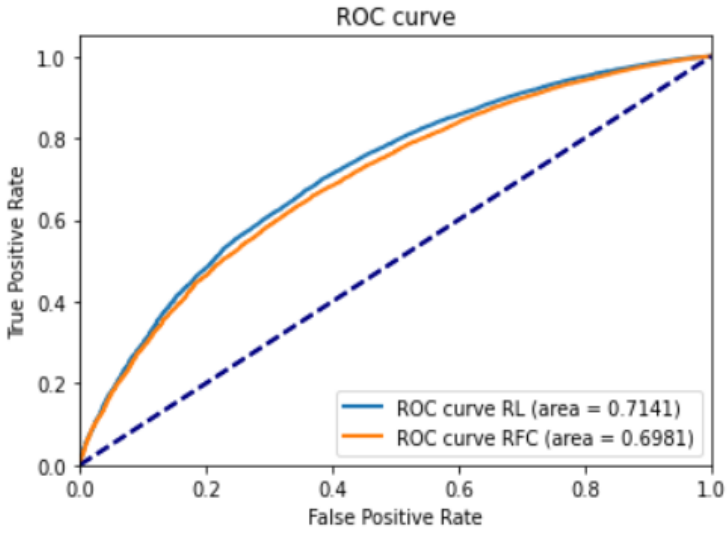
6.1 Métricas

Como se explica en la sección 4.5, las métricas de evaluación de para elegir los modelos seleccionados fueron el área ROC y el Recall. De esta manera, se seleccionan como modelos finales un modelo de regresión logística y otro modelo de random forest classifier. Los resultados de dicha métrica fueron de 0.7141 y 0.6981, respectivamente.

En la Figura 16 se muestra el área ROC obtenido en nuestros modelos con la data de evaluación.

Figura 16

Área ROC de los modelos para la data de evaluación.

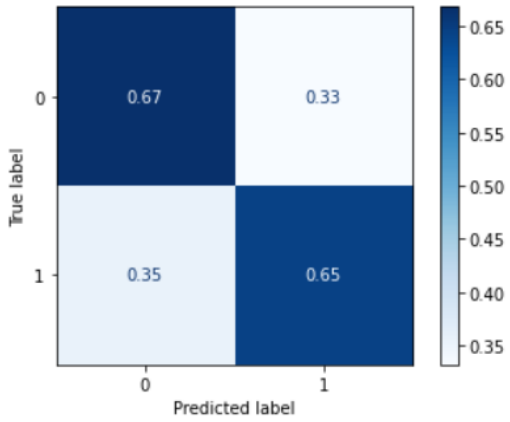


Elaboración: Fuente Propia

A continuación, en las ilustraciones 17 y 18, se puede observar la matriz de confusión para ambos modelos.

Figura 17

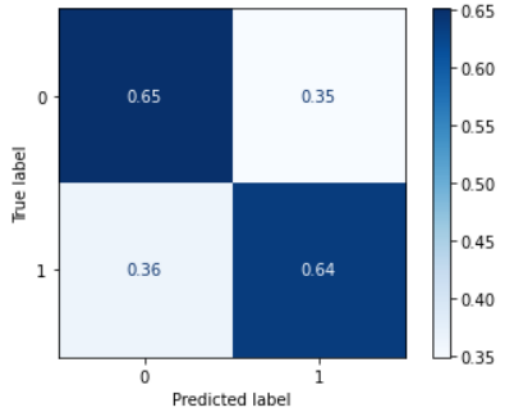
Matriz de confusión del modelo de regresión logística



Elaboración: Fuente Propia

Figura 18

Matriz de confusión del modelo de random forest classifier



Elaboración: Fuente Propia

En las figuras 17 y 18 es posible observar que para los ambos modelos elegidos, se predice correctamente como cliente que cae en default en un 65% y 64% respectivamente, teniendo en cuenta que son estos los clientes que, al predecirlos mal, sería crítico para el negocio. También se observa que los clientes que no caen en default, se predicen correctamente en un 67% y 65%, respectivamente.

A los modelos se les ajustaron los hiper parámetros, a continuación, se presentan dichos hiper parámetros y sus respectivos valores.

El modelo de regresión lineal cuenta con solo 2 hiper parámetros, “penalty” y “C”, al realizar dicha búsqueda se encuentra que el mejor modelo se obtiene con un “penalty”=l2 y un “C”=1.

El modelo de random forest classifierse realizó el ajuste a 5 hiper parámetros, “n_estimators”, “max_depth”, “min_samples_leaf”, “max_features”, “criterion”, y “ccp_alpha”, al realizar dicha búsqueda se encuentra que el mejor modelo se obtiene con un “n_estimators”=250, “max_depth”=5, “min_samples_leaf”=0.05, “max_features”=’auto’, “criterion”=gini, y “ccp_alpha”=0.

6.2 Evaluación cualitativa

Para la evaluación cualitativa de los modelos resultantes, a continuación, se observarán los valores de las métricas de la precisión (accuracy) tanto para los datos de entrenamiento como para los datos de prueba.

Tabla 8

Precisión de los modelos para los datos de entrenamiento y de prueba.

TÉCNICA	**Accuracy Train**	**Accuracy Test**
Regresión Logística	0.6559	0.6655
Random Forest	0.6506	0.6627

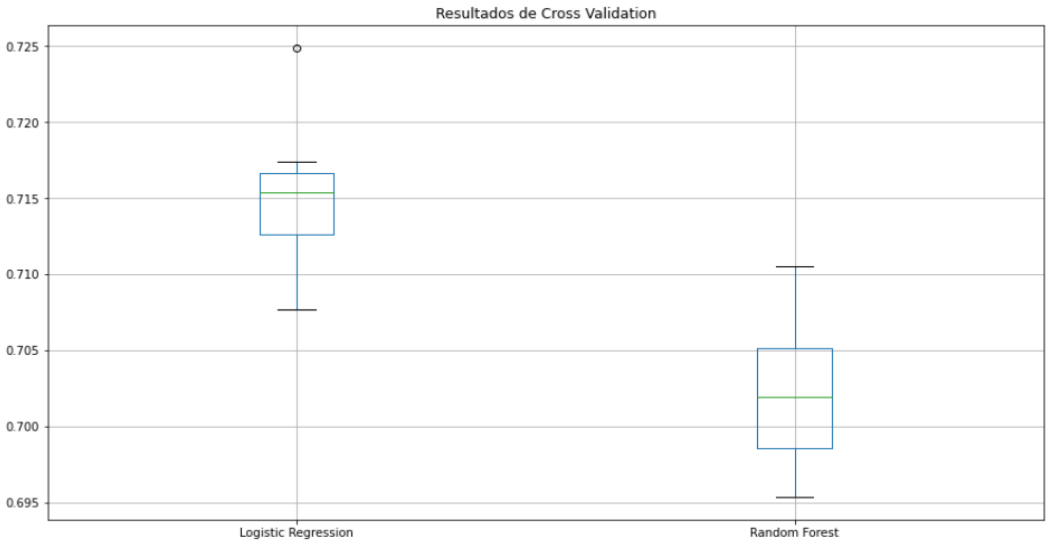
Fuente: Elaboración Propia.

Según los resultados obtenidos se puede observar que la diferencia entre la precisión de la data de entrenamiento y de prueba es menor al 2%, por tanto, para los modelos elegidos se puede decir que estos no presentan overfitting ni underfitting. Sin embargo, en algunos de los modelos de las diferentes iteraciones si se encuentran algunos casos de overfitting.

También se realiza una validación cruzada en estos modelos para observar la desviación y la media de las métricas haciendo esto con 10 pliegues. En la Figura 19 se puede observar un gráfico de caja, donde están los dos modelos ganadores. Dichos resultados permiten decir que no hay mucha variación del score cuando se cambia la data de validación.

Figura 19

Gráfico de caja de los modelos seleccionados.



Elaboración: Fuente Propia

El reto de kaggle permite ver que el modelo ganador de la competencia obtuvo un área ROC de 0.8057 con un número de variables de entradas igual a 499, donde al comparar con los resultados mostrados en la sección 6.1, se puede observar que el área ROC tiene una diferencia de alrededor del 9%, que si bien pareciera un poco alta dicha diferencia, al observar la cantidad de variables de entradas del modelo, se puede deducir que se usan diferentes técnicas y notoriamente se puede hablar de que hay diferentes costos computacionales, por ende, los resultados de los dos modelos seleccionados son aceptables. También, teniendo en cuenta que los modelos creados no usan una cantidad considerable de variables, se podría hablar de omisión de información que podría ser importante a la hora de predecir.

Llevar estos modelos a la oportunidad del negocio, puede ser tentador, sin embargo, tener una precisión de aproximadamente 65% es equivalente a hablar de un riesgo del 35% para la entidad bancaria de que un cliente que potencialmente vaya a caer en default sea predicho como un cliente que pagará su crédito, lo que representa un riesgo grande.

7. Conclusiones

El reto de kaggle permite ver que el modelo ganador de la competencia obtuvo un área ROC de 0.8057 con un número de variables de entradas igual a 499, mientras que los modelos desarrollados en este proyecto consta de solo 15 variables de entrada, donde al comparar con los resultados mostrados en la sección 6.1, se puede observar que el área ROC tiene una diferencia de alrededor del 9%, que si bien pareciera un poco alta dicha diferencia, al observar la cantidad de variables de entradas del modelo, se puede deducir que se usan diferentes técnicas y notoriamente se puede hablar de que hay diferentes costos computacionales, por ende, los resultados de los dos modelos seleccionados son aceptables.

A pesar de que el reto tenía una métrica de evaluación bajo el área ROC, se realiza análisis de las condiciones del negocio, aportándole al modelo, no solo un buen área ROC, sino aumentando la métrica Recall, que, para el negocio, es la más penalizada en caso de predecir mal.

A pesar de no tener cómo cubrir el costo computacional para lograr un modelo con muchas variables, como se quería con las máquinas de soporte vectorial, el tratamiento y la reducción de la data, nos permitió encontrar modelos con unos buenos resultados para la complejidad del problema, que comparado con la solución ganadora del reto, permitiría al cliente hacer un plan piloto sin necesidad de invertir en costo computacional y con este modelo mirar si para el plan de la empresa si tiene relevancia hacer un gasto económico en recursos computacionales para alcanzar el área roc del modelo ganador del reto de kaggle.

Si bien el costo computacional no permitió generar un mejor modelo, es posible aumentar el rendimiento del modelo con un mejor equipo de hardware, ya que para este proyecto se usó un equipo limitado. Además, teniendo un mejor equipo de cómputo, es posible implementar técnicas de ML más complejas, con estas dos soluciones, en un futuro se podría mejorar el modelo, teniendo la posibilidad de

no omitir información de una base de datos tan grande como la empleada y así volver más robusto el modelo.

8. Referencias

Abella, B., & González, A. (2021). *Mejora de las predicciones en muestras desbalanceadas.* Obtenido de https://repositorio.uam.es/bitstream/handle/10486/697900/abella_miravet_blanca_tfg.pdf?sequence=1

Basel Committee on Banking Supervision . (1999). Principles for the Management of Credit Risk. *Basel Committee on Banking Supervision*, 4. Obtenido de https://www.bis.org/publ/bcbs54.pdf

BBVA. (2021). *Análisis financiero: ¿Cómo se mide la calidad crediticia de un banco?* Obtenido de https://www.bbva.com/es/como-se-mide-la-calidad-crediticia-de-un-banco/

Borja, R., & Monleon, A. (2020). Estandarización de Métricas de Rendimiento para Clasificadores Machine y Deep Learning. *VI Congreso Internacional de Ciencia, Tecnología e Innovación para la Sociedad, CITIS*, (págs. 172-184). Obtenido de https://www.researchgate.net/publication/339943922_Estandarizacion_de_Metricas_de_Rendimiento_para_Clasificadores_Machine_y_Deep_Learning.

Castillo Rodríguez, M., & Pérez Hernández, F. (2008). Gestión del riesgo crediticio:un análisis comparativo entre Basilea II y el Sistema de Administración del Riesgo Crediticio Colombiano, SARC. *Pontificia Universidad Javeriana*, 229-250. Obtenido de https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiX38if4cT3AhWCdd8KHaHNAmYQFnoECB4QAQ&url=https%3A%2F%2Frevistas.javeriana.edu.co%2Findex.php%2Fcuacont%2Farticle%2Fview%2F3249%2F2471&usg=AOvVaw3YeaSi-i4DVm0SO1PWuVgv

Credito y Caución. (2022). *La morosidad bancaria, al 4,3%.* Obtenido de https://www.creditoycaucion.es/es/cycnews/entorno/detalle/Creditosdudosos

Cuenca, J. P. (2019). Propuesta de modelo de machine learning para la evaluación de riesgo de crédito utilizando algoritmos de predicción para la Cooperativa de Ahorro y Crédito La Merced Ltda. Cuenca. Obtenido de https://www.researchgate.net/publication/337480778_Propuesta_de_modelo_de_machine_learning_para_la_evaluacion_de_riesgo_de_credito_utilizando_algoritmos_de_prediccion_para_la_Cooperativa_de_Ahorro_y_Credito_La

Dastile, X., Celik, T., & Potsane, M. (2020). *Statistical and machine learning models in credit scoring: A systematic literature survey.* Obtenido de https://www.sciencedirect.com/science/article/abs/pii/S1568494620302039

Data Science Process Alliance. (s.f). *What is CRISP DM?* Obtenido de https://www.datascience-pm.com/crisp-dm-2/

decide. (2019). *Machine Learning en banca: gestión de riesgos crediticios.* Obtenido de https://decidesoluciones.es/machine-learning-banca-gestion-de-riesgos/#:~:text=Los%20modelos%20de%20Machine%20Learning,de%20un%2090%25%20de%20acierto.

El Economista. (2022). *La gran banca reduce la morosidad al 3,3%, mínimo histórico desde 2008.* Obtenido de https://www.eleconomista.es/empresas-finanzas/noticias/11746026/05/22/La-gran-banca-reduce-la-morosidad-al-33-minimo-historico-desde-2008.html

Google. (2022). *Colaboratoy.* Obtenido de https://research.google.com/colaboratory/intl/es/faq.html#:~:text=Colaboratory%2C%20o%20"Colab"%20para,análisis%20de%20datos%20y%20educación.

IBM. (2020). *Machine Learning*. Obtenido de https://www.ibm.com/cloud/learn/machine-learning

INCP. (2020). *El sector bancario, la IFRS 9 y el crédito procíclico.* Obtenido de https://incp.org.co/el-sector-bancario-la-ifrs-9-y-el-credito-prociclico/#:~:text=Las%20provisiones%20bancarias%20se%20realizan,un%20préstamos%20o%20crédito%20fallido.

Kaggle. (2018). *Home Credit Default Risk.* Obtenido de https://www.kaggle.com/competitions/home-credit-default-risk/data

Ossa, W., & Jaramillo, V. (2021). *Machine Learning para la estimación del riesgo de crédito en una cartera de consumo.* Obtenido de https://repository.eafit.edu.co/bitstream/handle/10784/29589/Wbeimar_OssaGiraldo_Veronica_JaramilloMarin_2021.pdf?sequence=2&isAllowed=y

Scikit-learn. (2022). *sklearn.feature_selection.f_classif.* Obtenido de https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html#sklearn.feature_selection.f_classif

Scikit-learn. (2022). *sklearn.feature_selection.SelectKBest.* Obtenido de https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

Villagrá, C. (2015). Sistema Predictivo Progresivo de Clasificación Probabilística como Guía de Aprendizaje. *Universitat d'Alacant*. Obtenido de https://rua.ua.es/dspace/bitstream/10045/54256/1/tesis_carlos_j_villagra_arnedo.pdf