ANÁLISIS DE RETENCIÓN DE CLIENTES EN INSTITUCIONES BANCARIAS BASADA EN DATOS DE TARJETAS DE CRÉDITO PARA PREDECIR LA LEALTAD DEL CLIENTE

Hadys Osvaldo Agudelo
Ingeniero de Sistemas – Universidad de Antioquia
Especialista en Gerencia de Proyectos - UNIMINUTO

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

Asesor
Efraín Oviedo Carrascal
Magíster (MSc)

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2023

Cita	(Agudelo H. O., 2023)
Referencia **Estilo APA 7 (2020)**	Agudelo, H. O. (2023). Análisis de retención de clientes en instituciones bancarias basada en datos de tarjetas de crédito para predecir la lealtad del cliente. Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte V.

Centro de Investigación Ambientales y de Ingeniería (CIA).

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.
Decano: Julio Cesar Saldarriaga Molina
Jefe departamento: Diego José Luis Botia Valderrama

Dedicatoria

A mi familia, la principal damnificada con el tiempo que tardé en realizar el estudio y cada una de las actividades de la especialización, ese valioso tiempo de compartir con la gente que amas y que es tu motor de vida es muy valioso e irrecuperable.

Gracias familia por entender que necesitaba salir de mi zona de confort, ver y aprender nuevas cosas que me llevaran a sentir vivo nuevamente. Fueron momentos muy especiales los vividos, de preocupaciones y pequeñas victorias. Felicidad por el nuevo conocimiento adquirido y tristeza por mermar mi tiempo de calidad con ustedes, pero con la satisfacción de contar con todo su apoyo y amor incondicional y entender que realmente necesitaba vivir esto.

Solo me resta decir gracias gracias gracias y que Dios nos permita disfrutar la vida por mucho tiempo más, si no, saben que los amé con pasión y siempre traté de disfrutarlos al máximo.

Agradecimientos

A mi hija, por compartir no solo su conocimiento conmigo, sino, su tiempo y paciencia además de hacerme ver el infinito amor que siente hacia mí.

Tabla de contenido

Lista de tablas

Lista de figuras

Resumen

La precisa clasificación de clientes a partir del análisis de movimientos en tarjetas de crédito reviste vital importancia en el panorama financiero actual. La creciente dependencia de las transacciones con tarjetas de crédito desafía a los bancos a identificar clientes de manera efectiva, considerando su capacidad económica, hábitos de gastos y perfiles de riesgo. Así, resulta crucial desarrollar técnicas avanzadas para esta clasificación, permitiendo a los bancos tomar decisiones informadas, gestionar riesgos y mejorar la experiencia del cliente. En este contexto, se implementaron diversas técnicas de ML utilizando un conjunto de datos de Kaggle, con el objetivo de discriminar el estado de deserción y fidelidad del cliente con las entidades bancarias según el historial reportado en las tarjetas de crédito.

La metodología empleada abordó la limpieza y tratamiento de datos, el análisis y la extracción de características, así como el modelado con diversos algoritmos de aprendizaje automático, seguido del ajuste de hiperparámetros específicos para cada modelo. La evaluación del rendimiento se llevó a cabo mediante diversas métricas, que incluyeron accuracy, precisión, recall, F1 score y el área bajo la curva ROC (AUC ROC). A pesar de los desafíos computacionales asociados con la optimización de hiperparámetros, se logró un rendimiento del 96% con el modelo Random Forest mediante la técnica de validación cruzada K-Fold.

Este resultado se atribuye posiblemente a la capacidad del modelo para capturar relaciones no lineales, gestionar grandes volúmenes de datos y aplicar técnicas avanzadas de regularización y optimización. Por lo tanto, destaca su habilidad para distinguir entre clientes propensos a abandonar el banco y aquellos propensos a permanecer, proporcionando así una perspectiva sólida sobre la problemática basada en los datos. https://github.com/osvalcode/Seminario

Palabras clave: clasificación de clientes, tarjetas de crédito, Machine Learning, análisis de datos financieros, modelado predictivo

Abstract

Accurate customer classification based on credit card transaction analysis is of vital importance in today's financial landscape. The increasing reliance on credit card transactions challenges banks to identify customers effectively, considering their financial capacity, spending habits and risk profiles. Thus, it is crucial to develop advanced techniques for this classification, enabling banks to make informed decisions, manage risks and improve the customer experience. In this context, several ML techniques were implemented using a Kaggle dataset, with the objective of discriminating customer defection and loyalty status with banks according to the history reported on credit cards. The methodology employed encompassed data cleaning and processing, feature analysis and extraction, modeling with various ML algorithms, and model-specific hyperparameter tuning. Performance evaluation was performed using various metrics, including accuracy, precision, recall, F1 score and AUC ROC. Despite computational challenges related to hyperparameter optimization, 96% performance with K-fold was achieved with a Random Forest. The performance of the RF outperformed the other architectures used, possibly due to its ability to capture nonlinear relationships, the handling of large volumes of data and the implementation of advanced techniques. Therefore, this result underscores their ability to discern between clients likely to leave and those likely to stay, thus providing a solid perspective on the issue from the data. https://github.com/osvalcode/Seminario

Keywords: *customer classification, credit cards, Machine Learning, financial data analysis*

1. Descripción del problema

La clasificación precisa de los clientes en función de la información recopilada de los movimientos en las tarjetas de crédito es de alta importancia en el sistema financiero actual. Con la creciente dependencia de las tarjetas de crédito para diversas transacciones, los bancos enfrentan el desafío de identificar y categorizar de forma efectiva a los clientes en función de su capacidad económica, hábitos de gastos y perfiles de riesgo [1]. La capacidad de clasificar con precisión a los clientes le permite a las instituciones y a las empresas ofrecer servicios personalizados, opciones de crédito ajustados a su historial y campañas de marketing dirigidas. Asimismo, cumple un papel crucial en la detección y prevención de actividades fraudulentas, minimizando el riesgo crediticio y manteniendo la estabilidad general del sistema financiero. Por lo tanto, el desarrollo de técnicas sólidas y sofisticadas para la clasificación de clientes basadas en la información de la tarjeta de crédito es esencial para que los bancos tomen decisiones informadas, mitiguen los riesgos y brinden experiencias óptimas a los clientes. Entre los diversos abordajes implementados, se encuentran tanto técnicas estadísticas como técnicas basadas en Machine Learning (ML).

Este último, ha ido ganando gran popularidad con el paso del tiempo, el cual ha sido estudiado en diversidad de campos, como, por ejemplo, el sector financiero, brindando una herramienta útil para el análisis de los datos de los clientes y tomar decisiones basados en ello[2].Este, al ser una técnica basada en datos, requiere ser utilizado con datos de alta calidad, con el fin de que en su tratamiento se puedan encontrar patrones y/o relaciones fiables y de esta forma, brindar perspectivas acerca la problemática financiera a resolver. Por ejemplo, con este tipo de técnicas se puede realizar una discriminación de clientes que me permita finalmente predecir la fiabilidad del mismo con el banco, lo que permite tomar decisiones (marketing, propuestas de retención y modificación de servicios financieros) basados en los datos analizados [3].

1.1. Problema de negocio

Una empresa financiera tiene como objetivo entender y atender las necesidades individuales de sus clientes de manera personalizada y eficaz. Su justificación, recae en que las transacciones con tarjetas de crédito son omnipresentes, por lo que se enfrentan a un constante desafío de ofrecer servicios según los movimientos de una amplia diversidad de perfiles de clientes, lo que puede conllevar a una deserción temprana o una insatisfacción del cliente con los productos ofrecidos por la empresa. Por ende, se presenta como solución la clasificación de clientes basados en este tipo de datos, lo que permitirá una comprensión más profunda del cliente con el objetivo de facilitar la toma de decisiones en cuanto a estrategias de marketing, oferta de servicios personalizados, límites de crédito ajustados, tasas de interés preferenciales u ofertas promocionales específicas. De esta forma, la empresa al analizar los datos de sus propios clientes puede analizar su flujo de dinero y los hábitos de gasto, identificando oportunidades para ofrecer productos financieros adaptados a las necesidades y capacidades de cada cliente.

1.2. Aproximación desde la analítica de datos

El desarrollo de técnicas sólidas y sofisticadas para la clasificación de clientes basadas en la información de la tarjeta de crédito es esencial para que los bancos tomen decisiones informadas, mitiguen los riesgos y brinden experiencias óptimas a los clientes. Basado en esto, en el presente trabajo se propone la exploración de diversos modelos de Machine Learning para la clasificación de clientes basado en datos recopilados de tarjeta de crédito.

El estudio de clientes basado en el uso de tarjetas de crédito ha sido un tema ampliamente estudiado en la industria bancaria, debido a su gran importancia en la planeación tanto de estrategias de marketing como de utilidad para la estabilidad financiera de los bancos a nivel mundial [4]. Entre los diversos abordajes implementados, se encuentran tanto técnicas estadísticas como técnicas basadas en Machine Learning (ML). Este último, ha ido ganando gran popularidad con el paso del tiempo, el cual ha sido estudiado en diversidad de campos,

como, por ejemplo, el sector financiero, brindando una herramienta útil para el análisis de los datos de los clientes y tomar decisiones basados en ello [5].

1.3. Origen de los datos

El conjunto de datos utilizado contiene amplia información de cartera de tarjetas de crédito de consumo de diversos clientes. Dichos datos registrados reflejan diferentes perfiles en aspectos tanto de edad, sexo, estado civil, categoría de ingresos, como información sobre relación de cada cliente con el proveedor de la tarjeta de crédito, como el tipo de tarjeta, número de meses registrado y períodos de inactividad. Este, fue recuperado de la plataforma Kaggle [5]

1.4. Métricas de desempeño

- **Evaluación de desempeño:**
  - **Accuracy:**

    Métrica que representa el porcentaje de clasificaciones correctas que logra un modelo de Machine Learning entrenado, es decir, el número de predicciones correctas dividido por el número total de predicciones en todas las clases, como se muestra a continuación en la Figura 1

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Figura 1 Ecuación accuracy

  - **Precisión:**

    Fracción de instancias relevantes entre las instancias recuperadas, es decir, la calidad de las predicciones positivas realizadas por el modelo. Dicha métrica se define en la siguiente figura

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Figura 2. Ecuación precisión

- **Recall:**

  También reconocida como sensibilidad, esta métrica indica el porcentaje de muestras que un modelo de ML identifica correctamente como pertenecientes a una clase de interés (la positiva) del total de las muestras de esa clase. Esta, se muestra a continuación en la Figura 3

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)}$$

Figura 3. Ecuación Recall

- **F1 Score:**

  Métrica que combina las medidas de precisión y recall en clasificación binaria (promedio armónico), utilizada para evaluar el rendimiento predictivo de modelos de ML, especialmente útil para datos no balanceados.

$$F1\ score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

Figura 4. Ecuación F1

- **AUC-ROC:**

  La Curva de Característica Operativa del Receptor (ROC) es un gráfico que representa el rendimiento de un modelo de clasificación en cada uno de los umbrales de clasificación, teniendo en cuenta la tasa de verdaderos positivos y falsos positivos. Por otra parte, el Área bajo la Curva ROC (AUC), mide el área bidimensional completa debajo de la curva ROC completa, agregando el

rendimiento en todos los umbrales de clasificación posibles. El ejemplo del gráfico se muestra a continuación

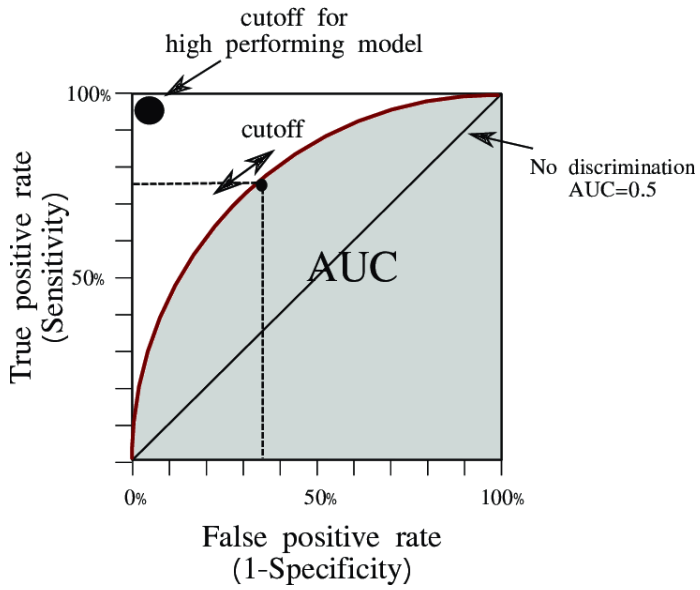


Figura 5. Ejemplo AUC-ROC

- **Métricas de negocio**

  - **Retorno de la Inversión (Return of Investment – ROI)**:

    Medida de desempeño utilizada para evaluar la eficiencia o rentabilidad de una inversión. Su objetivo, es medir directamente la cantidad de retorno de una inversión en particular, en la relación con el costo de la inversión.
    En la figura 6 se presenta la forma de calcular el ROI.

    En el caso particular del proyecto, esta métrica brindaría un apoyo respecto a los beneficios derivados obtenidos con la retención de los clientes, ya que, ante mayor rendimiento del modelo, mayor fiabilidad para ofrecer posibles productos personalizados ya sea prolongando su permanencia o encontrando ofertas ajustadas a las necesidades de los clientes. Por lo tanto, dado que se podrían obtener ganancias en ambas opciones si se toman las decisiones correctas, los costos del modelo no representarían un valor significativo que altere el margen de utilidad.

$$ROI = \frac{(Outcome - Costs)}{Costs}$$

Figura 6. Ecuación - Return of Investment

Por ende, para que el modelo proporcione un valor significativo, viable y confiable para la organización y se puedan tomar decisiones inteligentes basado en ello, se podría optar por un nivel de F1 del 85-90% como rendimiento aceptable en el modelo de predicción de clientes desertores. Ya que esto ayudaría a los directivos de la empresa a identificar con un margen de error aceptable y retener a los clientes que, de otra forma, por ejemplo, se habrían marchado, generando de esta forma ingresos adicionales. Determinar los costos exactos de adquirir nuevos clientes y retener clientes existentes puede ser difícil, ya que varía según la estrategia específica, la industria y la situación de cada banco. Se anexa una perspectiva de cómo podría realizarse un cálculo:

Adquisición de Clientes Nuevos:

Los costos de adquirir nuevos clientes suelen ser más altos porque implica esfuerzos de marketing, publicidad y promociones para atraer a personas que aún no tienen relación con el banco.

Incluye gastos en campañas publicitarias, eventos promocionales, incentivos para nuevos clientes, y posiblemente comisiones para agentes externos o intermediarios.

Retención de Clientes Existentes:

El costo de retención de clientes implica una combinación de gastos directos e indirectos asociados con estrategias y programas destinados a mantener la lealtad del cliente. Algunos de los factores que podrían influir en los costos de retención incluyen:

Programas de Lealtad:

Los bancos a menudo implementan programas de lealtad, como recompensas, bonificaciones o descuentos, para incentivar a los clientes a permanecer.

Servicio al Cliente:

Proporcionar un servicio al cliente excepcional, que podría incluir personal capacitado y sistemas eficientes, puede contribuir a la retención, pero también implica costos asociados.

Ofertas y Beneficios Personalizados:

Crear ofertas y beneficios personalizados para clientes existentes puede ser una estrategia efectiva, pero implica la asignación de recursos y posiblemente la oferta de tasas preferenciales.

Tecnología y Sistemas:

La implementación y el mantenimiento de tecnologías que mejoren la experiencia del cliente y faciliten las transacciones también pueden ser un componente del costo de retención.

Campañas de Marketing:

Las campañas de marketing dirigidas a la retención de clientes pueden implicar gastos en publicidad y promociones específicas.

Medición del Retorno de Inversión (ROI):

Evaluar la efectividad de las estrategias de retención a menudo implica el monitoreo y análisis del ROI, lo que a su vez puede tener costos asociados.

Es importante destacar que, aunque la retención de clientes puede tener costos, estos se comparan generalmente con el valor a largo plazo que un cliente puede aportar al banco. La retención de clientes exitosa no solo implica mantener a los clientes, sino también maximizar su participación y satisfacción a lo largo del tiempo.

Una fórmula de tasa de retención podría ser la siguiente:

Tasa de Retención = ((Clientes al final del período - Nuevos Clientes Adquiridos durante el período) / Clientes al inicio del período)) x 100

Una tasa de retención por encima del 80% es un buen indicador de retención.

2. Objetivos

2.1.Objetivo general

Desarrollar un modelo de Machine Learning utilizando datos de transacciones de tarjetas de crédito para clasificar clientes entre los propensos a abandonar la afiliación bancaria y los de alta probabilidad de retención, contribuyendo a estrategias preventivas de retención y eficiencia operativa en el sector financiero

2.2.Objetivos específicos

- Analizar datos recuperados de transacciones de tarjetas de crédito con el fin de identificar y extraer patrones que aporten información relevante para la construcción y el entrenamiento efectivo del modelo de Machine Learning
- Implementar un modelo de Machine Learning que permita la clasificación binaria entre clientes con tendencia a permanencia y deserción mediante datos de tarjetas de crédito
- Validar y ajustar el modelo con diferentes estrategias de optimización, con el propósito de asegurar una adecuada generalización en el proceso de aprendizaje y garantizar una precisión robusta en la predicción de clientes

3. Datos

3.1. Datos originales

El dataset OpenSource utilizado contiene amplia información de cartera de clientes de tarjeta de crédito, con el objetivo de predecir la pérdida de usuarios por parte una empresa. Este, incluye detalles tal como edad, sexo, estado civil y categoría de ingresos, así como información sobre la relación de cada cliente con el proveedor de la tarjeta de crédito, número de meses que han transcurrido desde la transacción (préstamo) y los períodos de inactividad, además de datos del comportamiento de gastos de los diferentes usuarios. Por lo tanto, este se compone tanto de datos discretos, continuos y categóricos, con un total de 23 características y 10127 observaciones. A continuación, se ilustra en la Figura 7 un resumen descriptivo de las características trabajadas.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   CLIENTNUM                 10127 non-null  int64
 1   Attrition_Flag            10127 non-null  object
 2   Customer_Age              10127 non-null  int64
 3   Gender                    10127 non-null  object
 4   Dependent_count           10127 non-null  int64
 5   Education_Level           10127 non-null  object
 6   Marital_Status            10127 non-null  object
 7   Income_Category           10127 non-null  object
 8   Card_Category             10127 non-null  object
 9   Months_on_book            10127 non-null  int64
 10  Total_Relationship_Count  10127 non-null  int64
 11  Months_Inactive_12_mon    10127 non-null  int64
 12  Contacts_Count_12_mon     10127 non-null  int64
 13  Credit_Limit              10127 non-null  float64
 14  Total_Revolving_Bal       10127 non-null  int64
 15  Avg_Open_To_Buy           10127 non-null  float64
 16  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 17  Total_Trans_Amt           10127 non-null  int64
 18  Total_Trans_Ct            10127 non-null  int64
 19  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 20  Avg_Utilization_Ratio     10127 non-null  float64
 21  NB12mon1                  10127 non-null  float64
 22  NB12mon2                  10127 non-null  float64
dtypes: float64(7), int64(10), object(6)
memory usage: 1.8+ MB
```

Figura 7. Descripción de características del dataset Kaggle

Asimismo, al dataset cuenta con dos tipos de etiqueta, los cuales se enuncian en la Tabla 1

Tabla 1. Clases del proyecto

Etiqueta	**Descripción**	**Cantidad/Porcentaje**
1	Clientes con tendencia a permanencia en la entidad bancaria	8700 / 85.9%
0	Clientes desertores de la entidad bancaria	1427 /14%

Como se puede observar, las clases presentes en el conjunto de datos no son balanceadas, puesto que hay más datos demográficos y de transacciones de tarjetas de crédito de clientes sin deserción que desertores, lo que puede presentar un desafío en el momento de entrenamiento del algoritmo de Machine Learning. Por otra parte, dichos datos estructurados están alojados en un archivo separado por comas el cual tiene un peso de 1.5 KB.

3.2. Datasets

En el aprendizaje automático, una tarea común es el estudio y la construcción de algoritmos que puedan aprender de los datos y hacer predicciones sobre ellos. Estos algoritmos funcionan haciendo predicciones o tomando decisiones basadas en los datos, mediante la construcción de un modelo matemático a partir de los datos de entrada. Estos datos de entrada utilizados para construir el modelo suelen dividirse en varios conjuntos de datos. En concreto, se suelen utilizar tres conjuntos de datos en distintas fases de la creación del modelo: conjuntos de entrenamiento, de validación y de prueba. Esto, se realiza una vez preprocesados los datos, con el fin de aleatorizar la elección de las muestras utilizadas para las diferentes fases del modelado, lo que permite por una parte obtener los parámetros de cada modelo candidato, ajustar los hiperparámetros de cada uno de los modelos y realizar pruebas en datos no vistos, respectivamente.

Para ello, se utiliza un método llamado Split, el cual realiza una estratificación basada en una variable específica de forma aleatoria según un porcentaje específico. Usualmente se utiliza

un 80:20 (entrenamiento:prueba) o un 65:15:20 si se incluye el conjunto de validación, dependiendo la proporción de la cantidad de datos que se tengan para la alimentación del algoritmo.

Por otra parte, para evaluar la generalización del modelo, se utiliza la validación cruzada, la cual consiste en dividir los datos en K subconjuntos de aproximadamente el mismo tamaño, para posteriormente realizar una iteración de los K-1 como datos de entrenamiento y el restante como datos de prueba. Este proceso, se realiza K veces, de manera que cada fold se utiliza una vez como conjunto de prueba y K-1 veces como conjunto de entrenamiento.
Esta técnica ayuda en el análisis predictivo para la reducción de la varianza de la estimación del rendimiento. Asimismo, ayuda a evitar el sobreajuste, ya que expone al modelo a diferentes subconjuntos de datos, brindando una opción para comparar diferentes modelos y parámetros de entrenamiento.

3.3. Analítica descriptiva

En la figura 8, se muestra el conteo de las variables categóricas del conjunto de datos utilizado.

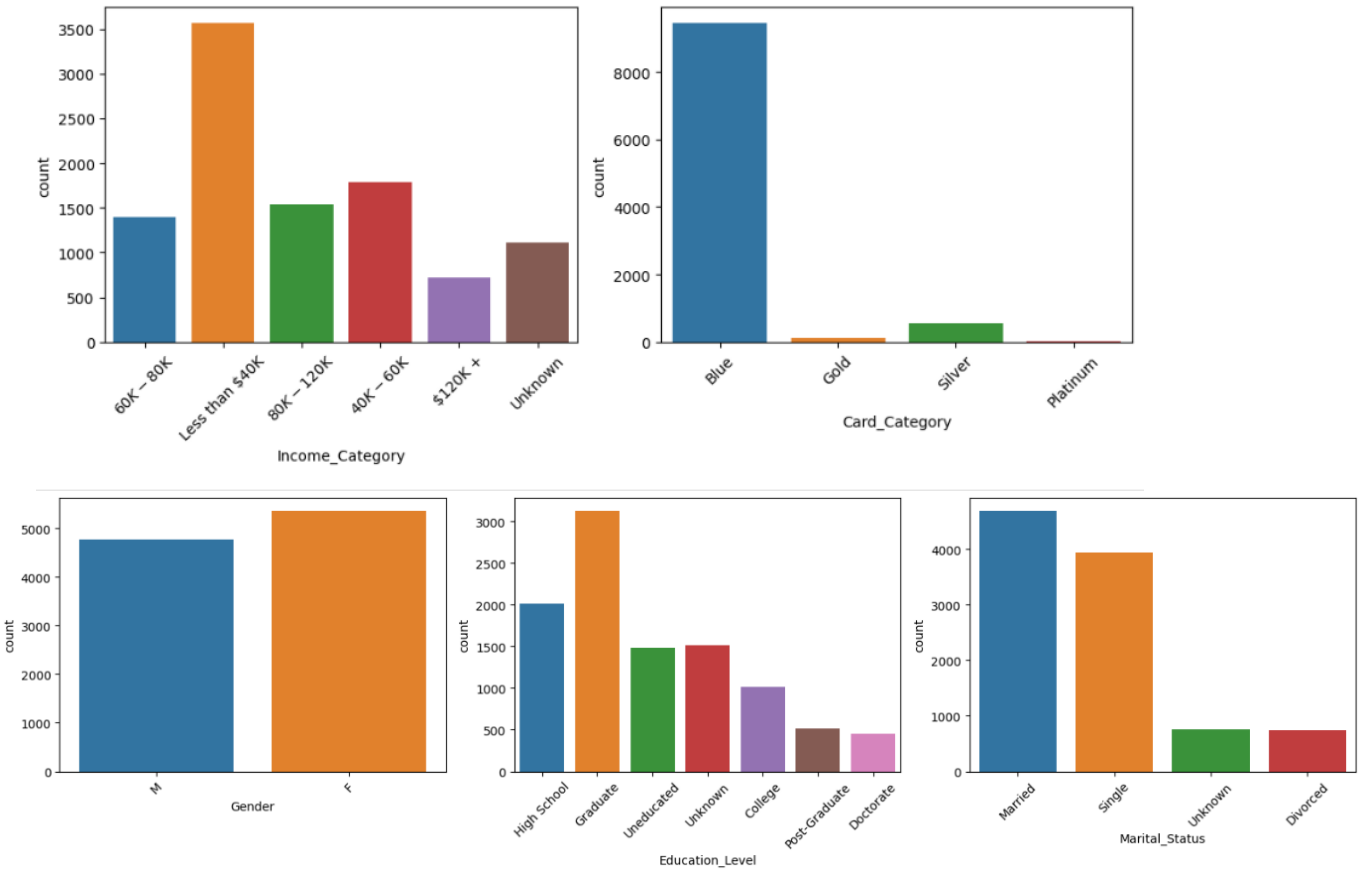


Figura 8 Gráficos de variables categóricas del dataset utilizado en el proyecto

En dichas gráficas, se puede observar un balanceo en los datos recopilados de sujetos femeninos y masculinos, mientras que en las demás variables se encuentra una pevalencia de estado civil casado, categoría de ingreso menos de 40K USD y categoría de tarjeta Blue. Además, se presenta también a continuación el histograma de las clases binarias:

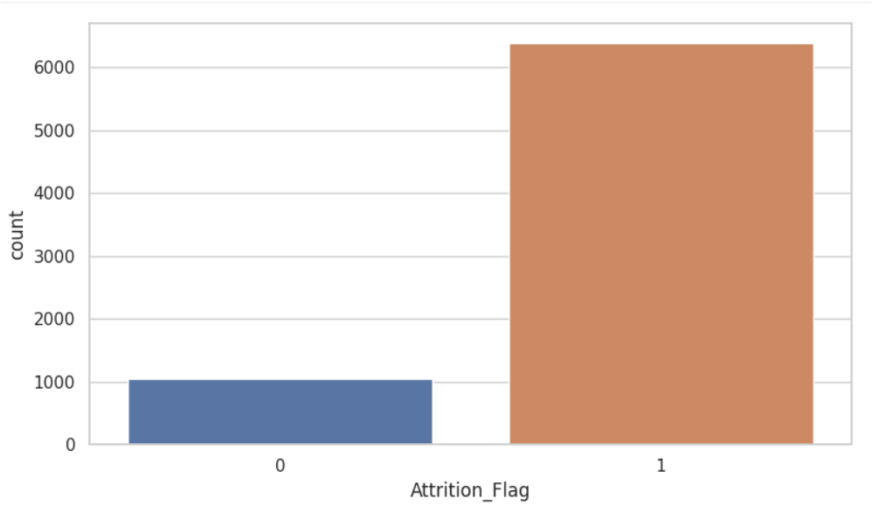


Figura 9. Histograma de clases binarias

En la Figura 9, se puede observar que hay mayor cantidad de la clase 1 que la 0, lo que significa que se debe recurrir a técnicas de balanceo o utilizar métricas, métodos y modelos específicos que puedan manejar este tipo de clases no balanceadas.

Con respecto a las variables continuas, se tienen las siguientes medidas de estadística básica reportadas en la *Tabla 2*

Tabla 2. Reporte de estadística básica de las variables numéricas

	Attrition flag	**Customer age**	**Dependent count**	**Months on book**
Count	10127	10127	10127	10127
Mean	0.84	46.33	2.35	35.93
Std	0.37	8.02	1.30	7.93
Min	0.00	26.00	0.00	13.00
25%	1.00	41.00	1.00	31.00
50%	1.00	46.00	2.00	36.00
75%	1.00	52.00	3.00	40.00
max	1.00	73.00	5.00	56.00

	Total_Relationship_Count	**Months_inactive_12_mon**	**Contacts_Count_12_mon**
Count	10127	10127	10127
Mean	0.84	46.33	2.46

Std	0.37	8.02	1.11
Min	0.00	26.00	0.00
25%	1.00	41.00	2.00
50%	1.00	46.00	2.00
75%	1.00	52.00	3.00
max	1.00	73.00	6.00

	Credit_Limit	**Total_Revolving_Bal**	**Total_Amt_Chng_Q4_Q1**	**Total_Trans_Amt**
Count	10127	10127	10127	10127
Mean	8631.95	1162.81	0.76	4404.09
Std	9088.78	814.99	0.22	3397.13
Min	1438.30	0.00	0.00	510.00
25%	2555.00	359.00	0.63	2155.50
50%	4549.00	1276.00	0.74	3899.00
75%	11067.50	1784.00	0.86	4741.00
max	34516.00	2517.00	3.40	18484.00

De la Tabla 2 se puede inferir lo siguiente:

-La edad promedio de los clientes es 46 años

-Casi todos los clientes tienen entre 1 y 3 personas que dependen económicamente de ellos

-Existen clientes sin dependientes y otros con un máximo de 5

-El promedio de relación con el banco es de 3 años

-Los clientes tienen al menos una tarjeta de otro proveedor

-En promedio, los clientes tienen un cupo de 8631.95, gran parte de los clientes poseen un cupo de 11067.50 y con hasta un máximo de $34516

-Los clientes tienen un saldo promedio de 1162.81, teniendo la mayor parte de ellos un saldo de 1784

-En promedio los clientes están inactivos solamente por dos meses y por un máximo de seis meses

-Los clientes poseen un promedio transaccional de 4404.09, teniendo la gran mayoría de la población total del banco un valor de 4741.00 y un valor máximo de $18484.00

-La gran mayoría de los clientes tienen una cantidad de transacciones de 81.

Por otra parte, se puede observar en la Figura 10 la matriz de correlación obtenida a partir de las características propias del dataset, donde se puede observar que hay variables altamente correlacionadas, lo que brinda información relevante para el manejo de selección de características e ingeniería de datos. Necesario para la limpieza y la preparación de los mismos.

	Attrition_Flag	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit	Total_Revolving_Bal
Attrition_Flag	1.000000	-0.018203	-0.018991	-0.013687	0.150005	-0.152449	-0.204491	0.023873	0.263053
Customer_Age	-0.018203	1.000000	-0.122254	0.788912	-0.010931	0.054361	-0.018452	0.002476	0.014780
Dependent_count	-0.018991	-0.122254	1.000000	-0.103062	-0.039076	-0.010768	-0.040505	0.068065	-0.002688
Months_on_book	-0.013687	0.788912	-0.103062	1.000000	-0.009203	0.074164	-0.010774	0.007507	0.008623
Total_Relationship_Count	0.150005	-0.010931	-0.039076	-0.009203	1.000000	-0.003675	0.055203	-0.071386	0.013726
Months_Inactive_12_mon	-0.152449	0.054361	-0.010768	0.074164	-0.003675	1.000000	0.029493	-0.020394	-0.042210
Contacts_Count_12_mon	-0.204491	-0.018452	-0.040505	-0.010774	0.055203	0.029493	1.000000	0.020817	-0.053913
Credit_Limit	0.023873	0.002476	0.068065	0.007507	-0.071386	-0.020394	0.020817	1.000000	0.042493
Total_Revolving_Bal	0.263053	0.014780	-0.002688	0.008623	0.013726	-0.042210	-0.053913	0.042493	1.000000
Avg_Open_To_Buy	0.000285	0.001151	0.068291	0.006732	-0.072601	-0.016605	0.025646	0.995981	-0.047167
Total_Amt_Chng_Q4_Q1	0.131063	-0.062042	-0.035439	-0.048959	0.050119	-0.032247	-0.024445	0.012813	0.058174
Total_Trans_Amt	0.168598	-0.046446	0.025046	-0.038591	-0.347229	-0.036982	-0.112774	0.171730	0.064370
Total_Trans_Ct	0.371403	-0.067097	0.049912	-0.049819	-0.241891	-0.042787	-0.152213	0.075927	0.056060
Total_Ct_Chng_Q4_Q1	0.290054	-0.012143	0.011087	-0.014072	0.040831	-0.038989	-0.094997	-0.002020	0.089861
Avg_Utilization_Ratio	0.178410	0.007114	-0.037135	-0.007541	0.067663	-0.007503	-0.055471	-0.482965	0.624022
Gender_F	-0.037272	0.017312	-0.004563	0.006728	-0.003157	0.011163	-0.039987	-0.420806	-0.029658
Gender_M	0.037272	-0.017312	0.004563	-0.006728	0.003157	-0.011163	0.039987	0.420806	0.029658
Education_Level_College	0.007840	-0.014788	0.003369	-0.010281	-0.013582	0.004038	-0.008996	0.001929	-0.011058
Education_Level_Doctorate	-0.029386	0.025199	-0.003368	0.024114	-0.009077	0.002432	-0.001016	-0.005195	-0.018208
Education_Level_Graduate	0.009046	-0.000203	0.000671	0.003531	0.005397	0.005885	0.002660	-0.004844	-0.000356
Education_Level_High School	0.011730	0.001199	-0.013127	0.002637	-0.001707	-0.005575	-0.003927	-0.001432	0.019276
Education_Level_Post-Graduate	-0.011127	-0.022081	0.009459	-0.016703	0.012050	-0.006240	-0.006878	0.005879	0.007068
Education_Level_Uneducated	0.001444	0.005057	0.002190	0.001099	0.008202	0.010127	0.012596	0.012213	-0.004446
Education_Level_Unknown	-0.009005	0.005377	0.004922	-0.003610	-0.003969	-0.012378	0.000843	-0.006478	-0.001219
Marital_Status_Divorced	-0.000850	-0.042614	0.006697	-0.027678	0.009276	0.001796	-0.008389	0.022578	-0.002368

Figura 10. Matriz de correlación de las características de los datos

A partir de los resultados obtenidos en la matriz de correlación, se decide analizar la distribución de los datos de las dos variables más correlacionadas, con fines de visualización y análisis de la información. Este, se muestra a continuación.

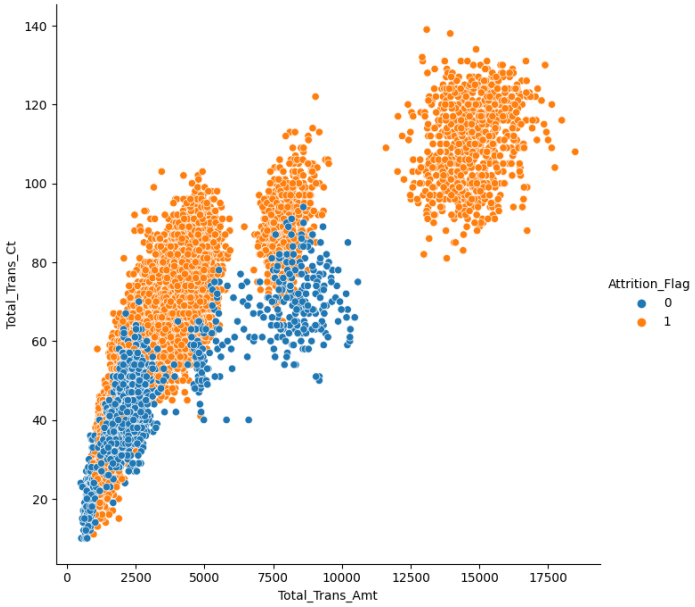


Figura 11. Gráfico de dispersión de dos características altamente correlacionadas

En este, se confirma la relación directamente proporcional de Total Trans Ct y Total Trans Amt, las cuales indican la cantidad de transacciones y el número de transacciones respectivamente referente a la clase respectiva. Este comportamiento es el esperado, puesto que a medida que haya un mayor uso de la tarjeta de crédito, mayor será la necesidad del cliente de pagar el saldo pendiente. Además, se puede observar que para los clientes que se retiran del banco, la cantidad a pagar es menor en relación con los que permanecen. Esto puede ser debido a que cuando el cliente se encuentra a paz y salvo con la tarjeta de crédito, puede llegar a retirarse más fácilmente del banco.

Por otra parte, para ver la diferencia que puede haber con la media, los cuartiles y la existencia de valores atípicos en cada una de las variables bajo estudio, según la clase perteneciente, se procede a visualizar dichas características mediante cajas de bigotes en la Figura 12

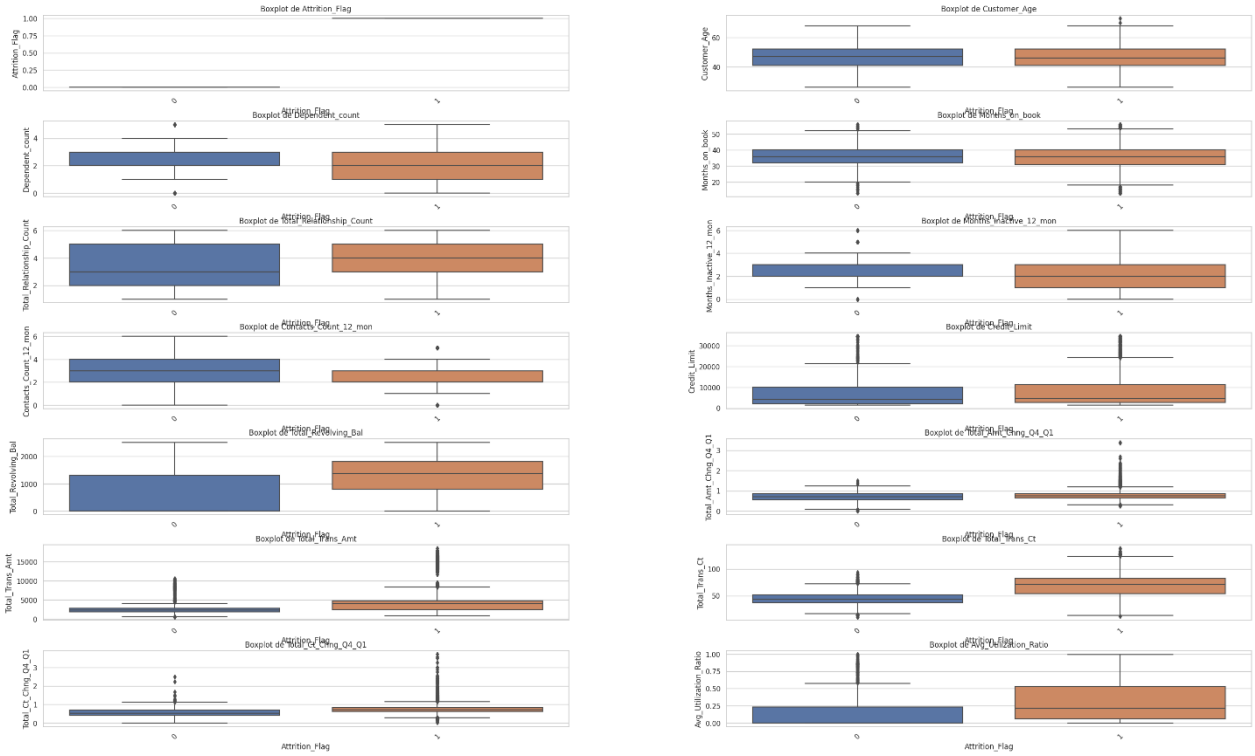


Figura 12. Boxplot de variables numéricas

Por lo tanto, basado en la gráfica de la caja de bigotes, se puede observar que en los datos sin procesar hay diferentes valores atípicos en varias de las características del dataset, y que en algunas en particulares, tales como *Total Revolving Bal*, *Contacts*, *Count 12 mon* ,*Total trans* y *avg utilization ratio*, se puede evidenciar diferencias significativas según la clase perteneciente (desertor o no desertor), lo que puede reflejar alta importancia en el momento del modelo tomar decisiones para la extracción de patrones diferenciadores en la tarea de clasificación.

3.4. Pipeline principal

En la Figura 13 se presenta el esquema general del flujo de trabajo seguido en el proyecto, en el cual se detallan los pasos, los algoritmos utilizados para responder a los objetivos planteados y las métricas empleadas para su evaluación.

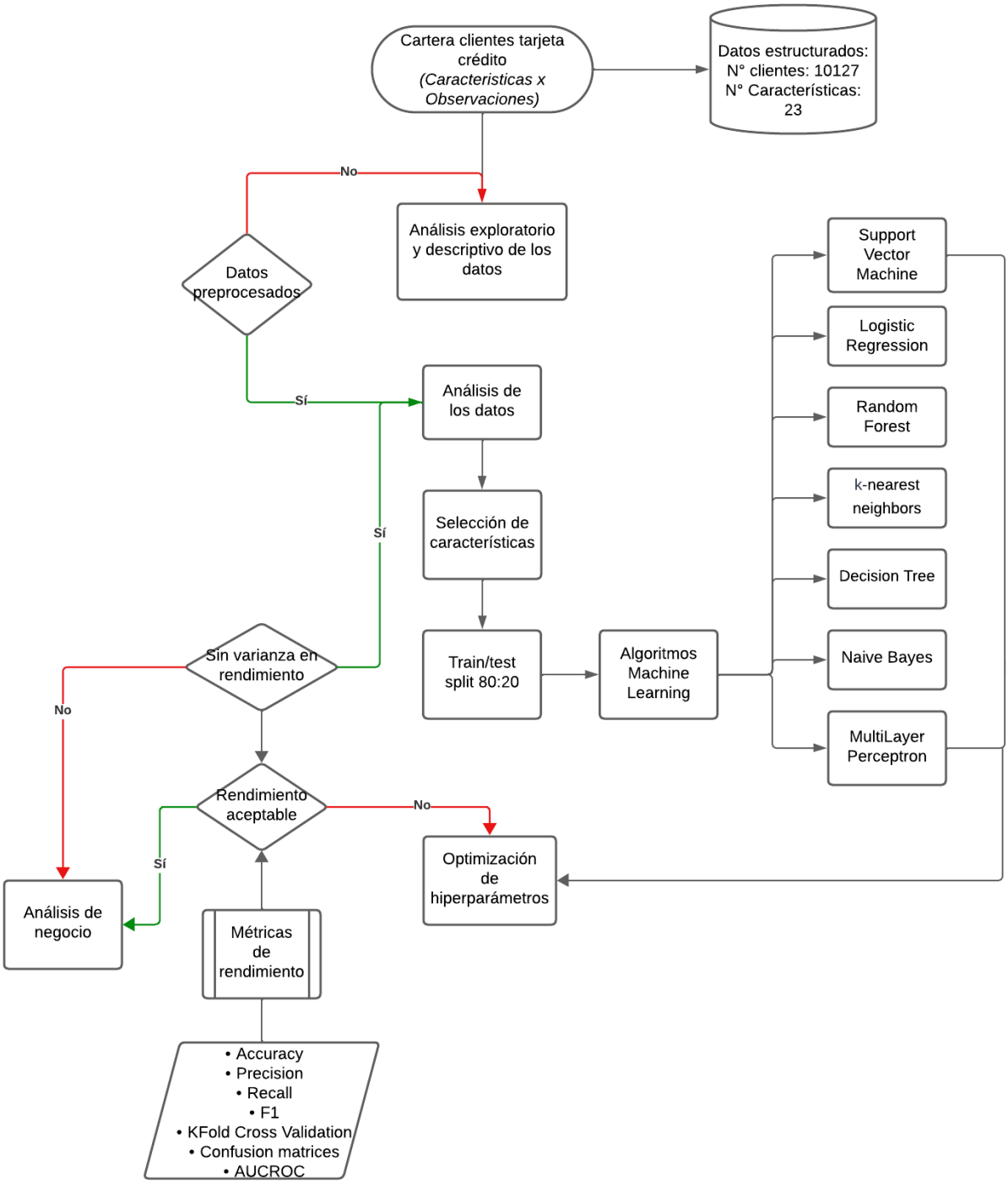


Figura 13. Flujo de trabajo seguido en el proyecto

Con el fin de cumplir con los objetivos del proyecto, se planteó la metodología anterior, la cual se parte de un conjunto de datos de tarjeta de crédito con 10127 registros y 23 características de interés. Dichos datos son preprocesados mediante diversas técnicas según el comportamiento y la variación de estos. Una vez se encuentren en óptimas condiciones se procede a hacer un análisis con las variables filtradas, lo que ayudará a decidir qué tipo de características son las más importantes con el fin de que el algoritmo pueda ser entrenado basado en información de interés, y así se puedan extraer patrones que permitan la identificación de las clases.

Posterior a ello, se realiza un *train/test Split*, el cual es necesario para la separación aleatoria del conjunto de datos que permita utilizar un porcentaje específico para la etapa de entrenamiento y la etapa de validación/prueba. Una vez los datos estén correctamente seccionados, se procede al entrenamiento con diversos algoritmos de ML, entre ellos SVM, LR, RF, KNN, árboles de decisión, Naive Bayes y MLP. Para ello, con el objetivo de realizar un entrenamiento adecuado, se deben ajustar los hiperparámetros propios del modelo estadístico específico de cada uno, el cual se puede evaluar con las diferentes métricas de rendimiento propias del problema de clasificación. Estas, nos permiten conocer qué tan bien el modelo generaliza y predice los clientes que pertenecen a un grupo específico, lo que conlleva a tomar decisiones para el mejor ajuste a los datos. Finalmente, cuando ya se obtenga un resultado aceptable de rendimiento, se procede a realizar un análisis de negocio, que brinde soluciones o estrategias que se deban tomar para le retención de clientes.

3.5. Preprocesamiento

En relación con el preprocesamiento de datos, entre otras fueron usadas las siguientes alternativas:

1. Limpieza de datos:
    - Manejo de datos faltantes

    En relación al manejo de datos faltantes, se evidenció este caso algunas variables, se muestra a continuación cuales fueron con su respectivo número:

Registros con Marital status Unknow: 749
Registros con Education_Level Unknow: 1519
Registros con Income_Category Unknow: 1112

En el caso de Marital status, este fue imputado con el valor más frecuente, el cual en nuestro caso fue “Married”.
Para los otros 2 casos, la imputación fue realizada con vecinos más cercanos, con un valor de 5 vecinos en el parámetro de ambos.

- Detección y manejo de datos atípicos

En la detección de valores atípicos, se separó en primera instancia el dataframe en variables categóricas y numéricas, sobre las numéricas se usó LocalOutlierFactor con 5 vecinos obteniendo los índices sobre los cuales se encontraban los valores atípicos, los cuales se refieren a observaciones que son inusualmente diferentes del resto de los datos en un conjunto. Estos valores atípicos pueden afectar significativamente los resultados de un análisis estadístico o de datos, distorsionando las medidas de tendencia central y afectando la validez de ciertos modelos.

Un ejemplo de valor atípico en una de las características se puede observar en la siguiente imagen:

	Credit_Limit
count	10127.00
mean	8631.95
std	9088.78
min	1438.30
25%	2555.00
50%	4549.00
75%	11067.50
max	34516.00

Figura 14.Ejemplo de dato atípico en variable Credit_Limit

Se destaca que, a pesar de que la media de la variable es de $8631, se han identificado límites de crédito de $34516, los cuales se consideran como valores atípicos, debido a que se dispersa en gran medida dicho valor respecto al promedio de los datos.

- Eliminación de duplicados

Después de la realización del análisis exploratorio, no se encontró en los datos registros duplicados en las observaciones del dataset.

2. Transformación de datos:
   - Normalización y escalamiento

- Para ejecutar este ítem en primer lugar fue dividido el dataframe en 2, uno para las variables categóricas y otro para las variables numéricas. Para escalar las variables numéricas fue usado MinMaxScaler de la librería Sklearn de Python.

Se presenta a continuación una muestra de 5 valores de la variable Credit_Limit antes y después de la normalización:

Tabla 3. Ejemplo variable numérica Credit_Limit antes y después de normalización

Credit Limit Variable	
Antes Normalización	**Después Normalización**
12691	0.4346
8256	0.0748
3418	0.0166
3313	0.0098
4716	1.000

Esta variable, representa el límite de crédito otorgado al cliente. Donde se puede observar que la variable tenía un comportamiento en todo el rango de los enteros positivos. Posterior a la normalización, se puede observar que el comportamiento de dicha variable ya está en el rango [0,1]

- Codificación de variables categóricas

Fue codificada la variable Attrition_Flag la cual originalmente contenía los valores Existing Customer y Attrited Customer. Esta variable fue cambiada a boolean con representación 1 y 0 respectivamente mediante One Hot Encoding

- Reducción de dimensionalidad

En el caso de reducción de dimensionalidad, en primera instancia se analizaron todas las características incluidas en el dataset, en donde se observó que la variable Naive_Bayes Classifier no brindaba información de interés para los análisis posteriores, por lo que se decidió eliminar de la muestra a estudiarse. En segunda instancia, al comprobar gráficamente la correlación entre todas las variables, se evidenció que Avg_Open_To_Buy y Credit_limit, respecto a la variable target, tienen una correlación muy baja (0.000284) (Ver figura 10), por lo que se tomó la decisión de eliminarla del dataset.

2. Manejo de Datos Desbalanceados

- Se realizó un balanceo sobre la variable target con el objetivo de obtener mejores resultados en las predicciones y que no estuvieran con un peso mayor hacia una predicción debido a cantidad de datos mayoritarios en los clientes fieles al banco. Este, se puede evidenciar a continuación, antes y después del balanceo de clases correspondientes:
  Data desbalanceada

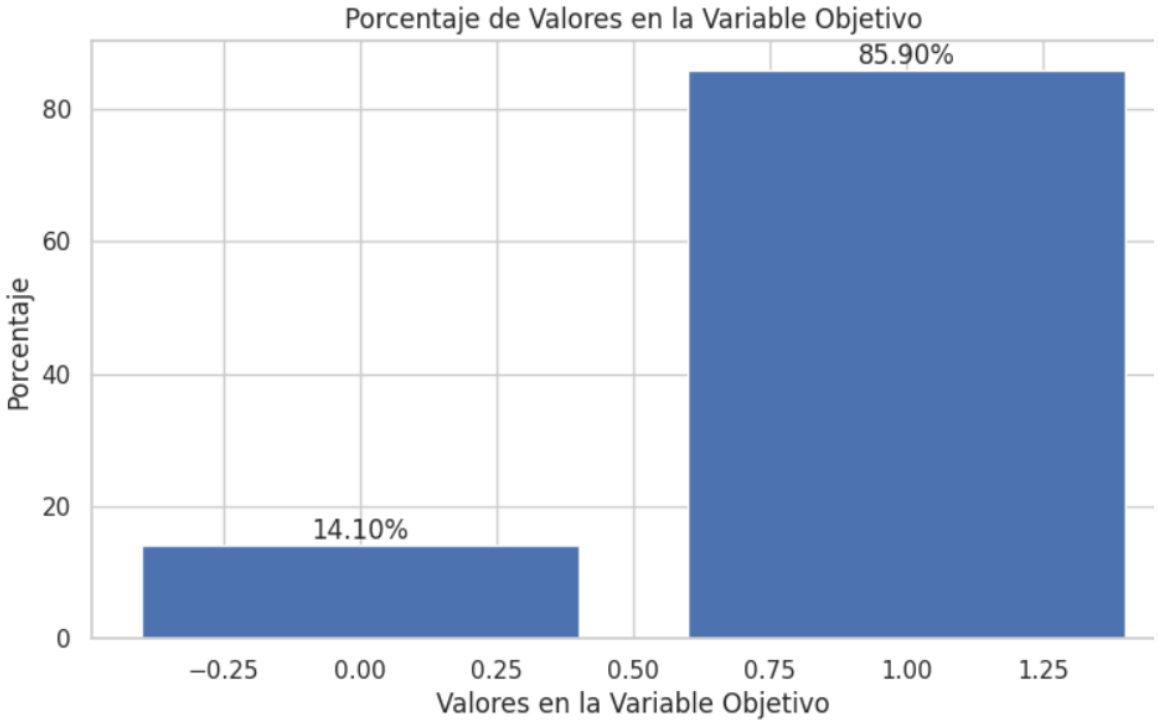


Figura 15. Porcentaje de clases al inicio del proyecto sin balancear

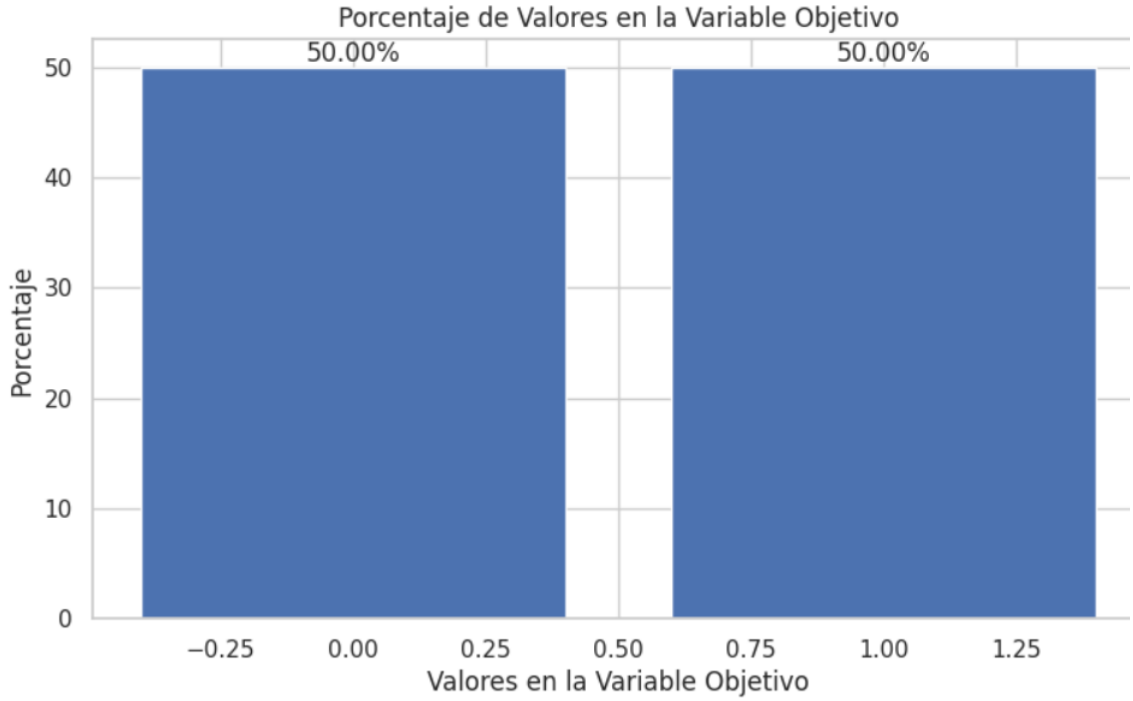


Figura 16. Porcentaje de clases después del balanceo

3.6.Modelos

En relación con los modelos considerados, fueron usados varios para poder realizar comparaciones entre ellos y ver cual tenía el mejor desempeño. Entre los valorados se tiene los siguientes:

1. **Support Vector Machine (SVM)**

SVM es un algoritmo de aprendizaje supervisado utilizado tanto para clasificación como para regresión. Su objetivo principal es encontrar un hiperplano en un espacio multidimensional que mejor separe las diferentes clases presentes en los datos [7].

Este algoritmo se basa en los siguientes principios básicos:

- **Hiperplano:** En un problema de clasificación binaria, el SVM busca el hiperplano que maximiza la distancia entre las dos clases. Este hiperplano es la frontera de decisión
- **Vectores de Soporte:** Son los puntos de datos más cercanos al hiperplano y son cruciales para determinar la posición y orientación del mismo. Estos puntos influyen en la definición del margen, que es la distancia entre el hiperplano y los puntos más cercanos de cada clase
- **Margen:** El SVM busca maximizar el margen, es decir, la distancia entre el hiperplano y los puntos más cercanos de cada clase. Un margen más amplio proporciona una mayor robustez y generalización del modelo
- **Kernel Trick:** SVM puede manejar datos no lineales transformándolos en un espacio de mayor dimensión a través de funciones kernel. Esto permite abordar problemas que no son linealmente separables en el espacio original

Al tener dichas propiedades, ofrece las siguientes ventajas:

- Efectivo en espacios de alta dimensión
- Buen rendimiento en conjuntos de datos con separación clara entre clases
- Versátil debido al uso de kernels para manejar datos no lineales

Sin embargo, puede ser sensible a la elección de los hiperparámetros que se escojan. Entre ellos, se encuentran:

- Kernel: función que cuantifica la similitud entre dos puntos de un espacio de características, donde el objetivo es realizar cálculos en el espacio de características de mayor dimensión sin necesidad de explícitamente calcular las coordenadas del espacio. Entre los posibles valores se encuentra lineal, polinómico, radial y sigmoidal.
- C: parámetro de regularización que busca encontrar el hiperplano que maximiza el margen entre las clases bajo estudio de clasificación. Esta, controla el equilibro entre dos objetivos, como la maximización del margen y la minimización de la clasificación errónea. Su rango de valores se encuentra entre (0, inf)
- Gamma: este parámetro es especialmente útil cuando se utiliza un kernel no lineal o kernel radial (RBF), el cual permite agregar una cantidad de dispersión específica a la función del kernel, conllevando a un límite de decisión más o menos suave. Entre menor sea este valor, tendrá una dispersión mucho más amplia (dispersión más suave). Sus valores oscilan entre (0, inf)
- Epsilon: es un parámetro que controla cuánta violación del margen es permitido, es decir, la cantidad máxima de holgura permitida para cada punto. A medida que este se aumenta, se permite una mayor violación del margen (0, inf), donde usualmente se prueba con valores en escala logarítmica (0.1, 1, 10, 100, etc)

Este algoritmo, se ha utilizado en amplia gama de aplicaciones, tanto de clasificación, como de regresión, en problemas de todo tipo, como, por ejemplo, financiero, biológicos y de medicina [8].

2. Logistic Regression model (ModelLR)

La Regresión Logística es un algoritmo de aprendizaje supervisado utilizado para problemas de clasificación, especialmente en situaciones donde la variable dependiente es binaria. Aunque lleva el nombre de "regresión", se utiliza comúnmente para la clasificación.

Su metodología, comprende los siguientes principios básicos:

- Función Logística (Sigmoide): La regresión logística utiliza la función logística o sigmoide para transformar la salida del modelo a un rango entre 0 y 1. Esto permite interpretar la salida como la probabilidad de pertenecer a una clase.
- Probabilidades y Umbral: La salida de la regresión logística se interpreta como la probabilidad de que un punto de datos pertenezca a la clase positiva. Al aplicar un umbral (típicamente 0.5), se clasifica a los puntos como pertenecientes a la clase positiva si la probabilidad es mayor que el umbral, y a la clase negativa de lo contrario.
- Función de Costo Logarítmico (Log-Loss): La regresión logística utiliza la función de coste logarítmico para medir la diferencia entre las predicciones y las etiquetas reales. El objetivo es minimizar esta función para mejorar la precisión del modelo.

Como ventajas, podemos encontrar las siguientes:

- Simple e interpretable.
- Eficiente en problemas de clasificación binaria.
- Proporciona probabilidades que pueden ser útiles en algunos contextos.

Sin embargo, puede no funcionar bien en conjuntos de datos con relaciones no lineales complejas y ser sensible a valores atípicos

Al ser un método versátil, se ha utilizado en diversos tipos de problema, como el diagnóstico médico, análisis de riesgos crediticios y clasificación de correos electrónicos como spam o no spam [9].

Entre sus hiperparámetros, se encuentran:

- C: parámetro inverso de la fuerza de regularización, que ayuda a evitar el sobreajuste al penalizar coeficientes grandes. Este tiene un rango de valores de (0, inf), donde a

menor valor implica una regularización más fuerte, conllevando a una simplicidad mayor del modelo.

- Penalty: parámetro que especifica la norma utilizada en la regularización, como L1 (puede conllevar a coeficientes dispersos), L2 (penaliza la magnitud cuadrática de los coeficientes), Elastic Net (combinación de penalizaciones L1, L2)
- Tol: tolerancia para la detección de convergencia durante el entrenamiento. Este oscila entre valores positivos pequeños como 1e-4 o 1e-5.
- Solver: parámetro utilizado para la optimización durante el entrenamiento. Alguno de sus valores son liblinear, newton-cg, lbfgs y sag,saga

3. k-nearest neighbors (KNN)

El algoritmo k-Nearest Neighbors (KNN) es un método de aprendizaje supervisado utilizado para problemas de clasificación y regresión. Su enfoque principal es asignar a un punto de datos la etiqueta que más frecuentemente aparece entre sus k vecinos más cercanos

Para su funcionamiento, se tienen en cuenta los siguientes principios básicos:

- Vecinos más Cercanos: El "k" en KNN representa el número de vecinos más cercanos que se toman en cuenta para realizar una predicción. La elección de "k" es un hiperparámetro crucial y puede afectar el rendimiento del modelo.
- Medida de Distancia: KNN utiliza una medida de distancia, como la distancia euclidiana, para calcular la proximidad entre puntos de datos. Los vecinos más cercanos son aquellos con la menor distancia al punto que se está clasificando.
- Votación Mayoritaria: Para la clasificación, KNN asigna la clase más común entre los k vecinos más cercanos al punto a clasificar. En problemas de regresión, se promedian los valores de los k vecinos más cercanos.

Entre sus ventajas se encuentran:

- Fácil de entender e implementar.

- No requiere entrenamiento explícito, ya que almacena todos los datos de entrenamiento.
- Funciona bien en conjuntos de datos pequeños y simples.

Como desventajas:

- Sensible a datos atípicos.
- Puede ser computacionalmente costoso, especialmente en conjuntos de datos grandes.
- Sensible a la elección de la métrica de distancia y del valor de "k"

Por lo tanto, se utiliza en aplicaciones tales como [10]:

- Recomendación de productos.
- Clasificación de documentos.
- Diagnóstico médico.

4. Complement Naive Bayes

El Complement Naive Bayes (CNB) es una variante del algoritmo de Naive Bayes, que se utiliza comúnmente para problemas de clasificación en aprendizaje supervisado.

Este, se basa en el teorema de Bayes para realizar predicciones. Este teorema utiliza la probabilidad condicional para calcular la probabilidad de que ocurra un evento dado que otro evento ya ha ocurrido. A continuación, se enuncian sus principios básicos para su funcionamiento:

- Modelo Probabilístico: CNB es un modelo probabilístico que calcula la probabilidad de pertenencia a una clase dada la presencia de ciertas características. Asume independencia condicional entre las características dadas las clases, similar al Naive Bayes tradicional.
- Enfoque de Complemento: A diferencia del Naive Bayes convencional, que asigna pesos a las características presentes, CNB utiliza un enfoque de complemento. Esto

significa que se ponderan más las características que no están presentes en la clase, lo que puede ser beneficioso en situaciones donde hay desequilibrios en la frecuencia de las clases.

Dada su naturaleza probabilística, presenta las siguientes ventajas:

- Eficiente y rápido para entrenar y predecir.
- Funciona bien en conjuntos de datos con características categóricas.
- Robusto ante datos ruidosos y es menos propenso al sobreajuste.

Sin embargo, presenta las siguientes desventajas:

- La suposición de independencia condicional puede no ser realista en todos los conjuntos de datos.
- No es ideal para datos con dependencias significativas entre las características.

Por otra parte, se ha utilizado en aplicaciones tales como [11]:

- Clasificación de texto.
- Detección de spam.
- Análisis de sentimientos

Entre sus parámetros, se encuentran:

- Alpha: parámetro de suavizado, utilizado para evitar problemas cuando se encuentran clases o características que no están presentes en el conjunto de entrenamiento. Entre más alto su valor, más suavizado y menor sensibilidad a valores atípicos. Su rango oscila entre (0, inf)
- Fit prior: parámetro que controla si se deben aprender o ajustar las probabilidades a priori de las clases a partir de los datos de entrenamiento. Rango booleano (True, False), donde True indica que el modelo ajustará automáticamente las probabilidades a priori basándose en la distribución de las clases en el conjunto de training.
- Class prior: parámetro que permite proporcionar manualmente las probabilidades a priori de las clases. Si se proporciona, anula la estimación ajustada por fit_prior. Se

debe proporcionar una lista con un rango de valores flotantes donde cada uno representa la probabilidad a priori de la clase correspondiente.

5. Decision Tree

El Árbol de Decisión es un algoritmo de aprendizaje supervisado que se utiliza tanto para problemas de clasificación como de regresión. Su objetivo principal es dividir el conjunto de datos en subconjuntos homogéneos basándose en las características para tomar decisiones.

Para su funcionamiento, sigue los principios básicos enunciados a continuación:

- Nodos y Ramas: Un árbol de decisión consta de nodos y ramas. Cada nodo representa una característica o atributo, y cada rama representa una decisión basada en esa característica.
- Nodo de Decisión: Los nodos de decisión se utilizan para realizar divisiones en el conjunto de datos. Cada nodo compara una característica con un umbral y, según el resultado, dirige el flujo del árbol hacia otro nodo o hacia una hoja.
- Hoja (Nodo Terminal): Las hojas del árbol representan las etiquetas de clasificación o los valores de regresión. Cuando el árbol alcanza una hoja, se realiza la predicción final.
- Criterios de División: El árbol de decisión utiliza criterios como Gini impurity o entropía para determinar la mejor manera de dividir el conjunto de datos en cada nodo de decisión. Estos criterios miden la homogeneidad de las clases en los subconjuntos resultantes.

Ventajas:

- Fácil de entender e interpretar.
- No requiere normalización de datos.
- Maneja tanto datos numéricos como categóricos.

Desafíos:

- Puede ser propenso al sobreajuste, especialmente en árboles profundos.

- Sensible a pequeñas variaciones en los datos.

Aplicaciones:

- Detección de fraudes.
- Diagnóstico médico.
- Clasificación de clientes para marketing [12]

Entre sus parámetros, se puede encontrar:

- Max_depth: especifica la profundidad máxima del árbol, controlando la complejidad del modelo, previniendo el sobreajuste. Oscila entre valores enteros positivos (0,inf) o None
- Min_samples_split: número mínimo de muestras requeridas para dividir un nodo interno. Oscila entre valores enteros positivos
- Max_features: controla el número máximo de características que se deben considerar para la división de un nodo. Puede ser un número entero, porcentaje o cadena, como auto, sqrt, log2, ayudando a controlar la diversidad y complejidad del árbol.
- Criterion: parámetro que mide la calidad de una partición. En Sklearn se encuentran dos funciones, Gini o entropy.

6. Random Forest

Random Forest es un algoritmo de aprendizaje supervisado que utiliza múltiples árboles de decisión para realizar predicciones. Es una técnica de conjunto (ensemble) que combina las predicciones de varios modelos base para mejorar la precisión y la robustez del modelo.

Dicho algoritmo se base en los siguientes principios básicos:

- Ensemble de Árboles de Decisión: Random Forest construye múltiples árboles de decisión durante el proceso de entrenamiento. Cada árbol se entrena con una muestra

aleatoria del conjunto de datos y utilizando un subconjunto aleatorio de características.

- Bagging (Bootstrap Aggregating): El proceso de muestreo aleatorio con reemplazo se conoce como bagging. Cada árbol se entrena con una muestra diferente del conjunto de datos, y las predicciones finales se obtienen promediando (en el caso de regresión) o realizando una votación (en el caso de clasificación) entre los árboles.
- Aleatorización de Características: Para cada árbol, solo se considera un subconjunto aleatorio de características en cada nodo de decisión. Esto introduce más diversidad entre los árboles, mejorando la generalización del modelo.

Presenta las siguientes ventajas y desventajas:

Ventajas:

- Mejora la precisión y la generalización en comparación con un solo árbol de decisión.
- Menos propenso al sobreajuste debido a la diversidad de los árboles y el bagging.
- Maneja bien conjuntos de datos grandes y con muchas características.

Desventajas:

- Menos interpretable que un árbol de decisión único.
- Puede ser computacionalmente costoso en comparación con modelos más simples.

Gracias a su capacidad matemática de clasificación, se ha utilizado en amplios escenarios, tales como [13]:

- Clasificación y regresión.
- Detección de fraudes.
- Diagnóstico médico.
- Análisis de imágenes y reconocimiento de patrones

Este algoritmo tiene ciertos parámetros que afectan su comportamiento y complejidad del modelo, tales como:

- N_estimators: número de árboles en el bosque. Cuanto mayor sea el número, mejor será la generalización del modelo. Oscila entre valores enteros positivos como 100, 200, 500, etc
- Max_depth: especifica la profundidad máxima de cada árbol en el bosque. Entre sus valores, están los enteros positivos o None
- Min_samples_split: mínimo de muestras requeridas para dividir un nodo interno. Rango de valores enteros positivos
- Min_samples_leaf: mínimo de muestras requeridas para formar una hoja. Controla la cantidad mínima de datos necesarios para formar una hoja del árbol. Posibles valores (0, inf)
- Max_features: controla el número máximo de características que se deben considerar para la división de un nodo, puede ser un número entero, un porcentaje o una cadena (auto, sqrt, log2)
- Bootstrap: indicador booleano que especifica si se deben usar muestras con reemplazo (bootstrap) al construir árboles.

7. Multi-Layer Perceptron

El Perceptrón Multicapa (MLP) es un tipo de red neuronal artificial que consta de múltiples capas de nodos, incluyendo una capa de entrada, una o más capas ocultas y una capa de salida. Estas redes se utilizan comúnmente en problemas de aprendizaje supervisado para clasificación y regresión.

Principios Básicos:

- Capa de Entrada: Representa las características del conjunto de datos y consta de nodos que transmiten la información a las capas ocultas.
- Capas Ocultas: Estas capas realizan transformaciones no lineales en los datos. Cada nodo en una capa oculta combina la información de la capa anterior y pasa la salida a la siguiente capa.

- Capa de Salida: Produce la predicción final del modelo. La cantidad de nodos en esta capa depende del tipo de problema, siendo un nodo para problemas de clasificación binaria, y más de uno para problemas de clasificación multiclase o regresión.
- Pesos y Sesgos: Cada conexión entre nodos tiene un peso que ajusta la contribución de esa conexión. Además, cada nodo tiene un sesgo que ajusta la salida de la suma ponderada.
- Funciones de Activación: Las funciones de activación, como la sigmoide, la tangente hiperbólica (tanh) o la función rectificadora lineal (ReLU), introducen no linealidades en el modelo, permitiendo aprender patrones más complejos.
- Entrenamiento: Se utiliza el algoritmo de retropropagación (backpropagation) para ajustar los pesos y sesgos de la red. Este algoritmo minimiza una función de pérdida que mide la discrepancia entre las predicciones del modelo y las etiquetas reales del conjunto de entrenamiento.

Entre sus ventajas y desventajas se encuentran:

Ventajas:

- Capacidad para aprender patrones complejos y no lineales.
- Puede manejar grandes cantidades de datos y características.

Desventajas:

- Requiere un mayor tiempo de entrenamiento y más datos que modelos más simples.
- Puede ser susceptible al sobreajuste.

Aplicaciones [14]:

- Clasificación de imágenes.
- Procesamiento de lenguaje natural.
- Pronóstico financiero.

Para su entrenamiento, se deben tener en cuenta los siguientes parámetros:

- Hidden_layer_sizes: Tupla de números enteros positivos, el cual especifica la arquitectura de la red neuronal, es decir, el número de neuronas de cada capa oculta.

- Activation: parámetro que especifica la función de activación utilizada en las capas ocultas. Pueden tener los siguientes valores ReLu, Logistic, TanH, entre otros.
- Solver: especifica el algoritmo de optimización utilizado para ajustar los pesos de la red. Posibles valores: Adam, sgd, lbfgs, entre otros
- Learning_rate_init: especifica la tasa de aprendizaje inicial, la cual controla el tamaño de los pasos que se dan durante la optimización. Puede tener valores enteros positivos, donde un valor más bajo puede conducir a una convergencia más precisa, pero puede requerir más tiempo de entrenamiento.
- Alpha: término de regularización que penaliza los pesos grandes, ayudando a prevenir el sobreajuste. Puede tener valores positivos
- Max_iter: especifica el número máximo de iteraciones (épocas) sobre el conjunto de entrenamiento. Controla la cantidad de veces que se ajustarán los pesos de la red. Valores enteros positivos
- N_outputs: parámetro que especifica el número de neuronas en la capa de salida, que generalmente corresponde al número de clases en el problema de clasificación. Valor entero positivo.

Como se puede observar, fueron usados diferentes modelos de scikit-learn, incluidos SVM, regresión logística, K vecinos más cercanos, Complement Naive Bayes, árbol de decisión, random forest y un perceptrón multicapa. Cada modelo se entrena con los datos de entrenamiento representados por X_train (características) e y_train (etiquetas).

3.7.Métricas

Este punto es esencial para evaluar la calidad de un modelo y su impacto en los objetivos.
Para calcular métricas comunes de evaluación de modelos, como F1 Score, Accuracy y Precision, en Python, generalmente se utilizan bibliotecas de aprendizaje automático como scikit-learn:

Métricas de Negocio:

En relación a las métricas de negocio, existen varias que pudiesen evaluarse, para el caso específico nuestro, se evalúa tácitamente el incremento en la retención de clientes.

El cálculo de métricas de negocio requiere una comprensión profunda de los objetivos y procesos comerciales. Estas métricas se utilizan para evaluar el impacto real de un modelo en la organización y tomar decisiones basadas en datos. Por lo tanto, es importante alinear las métricas de ML con los objetivos de negocio para medir el éxito de un proyecto de aprendizaje automático.

4. Metodología

4.1. Baseline

La ejecución de todo el proyecto se realizó desde Google Colab, dadas las características de máquina que brinda, superando las locales, lo que permite un mejor rendimiento con un menor tiempo de la ejecución de cada uno de los entrenamientos y manejo de datos en general.

Inicialmente, se aplicó una imputación en las variables estatus marital, nivel educativo y categoría de ingreso, utilizando vecinos cercanos mediante una función personalizada llamada imputar_vecinos. Esta función, reemplazó los valores desconocidos con el valor más común dentro de la ventana de vecinos.
Sin embargo, se utilizó posteriormente SimpleImputer con estrategia del más frecuente para el reemplazo de los valores faltantes.

A continuación, se presentan los Accuracy obtenidos para cada modelo en la primera iteración:

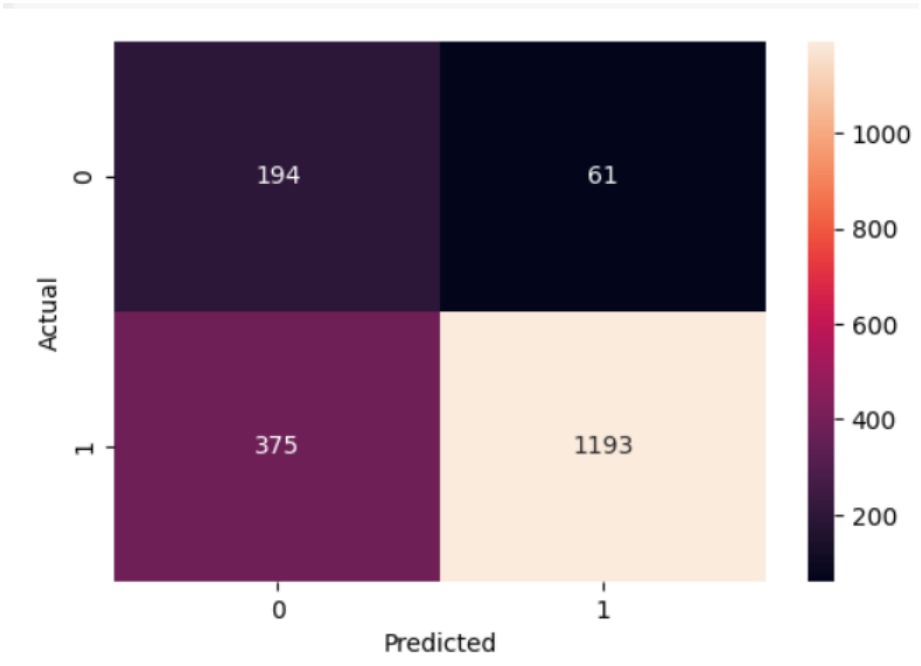
```
===== Accuracy de los modelos ====================

modelSVC      : 0.9072956664838179
modelLR       : 0.9051014810751509
modelknn      : 0.8710916072408118
modelComplNB  : 0.7608337904552934
modelTreeClas : 0.9100383982446517
modelRanForest: 0.9237520570488206
ModelNN       : 0.8601206801974767
```

Figura 17. Accuracy de los evaluados

Es de anotar que estos valores fueron obtenidos SIN hacer un balanceo de la variable target.

Se puede observar en la Figura 18 el rendimiento de uno de los modelos sin tener en cuenta el balanceo:



	precision	recall	f1-score	support
0	0.34	0.76	0.47	255
1	0.95	0.76	0.85	1568
accuracy			0.76	1823
macro avg	0.65	0.76	0.66	1823
weighted avg	0.87	0.76	0.79	1823

Figura 18. Matriz de confusión del algoritmo Random Forest

Como se puede observar, los valores no son ideales, ya que por ejemplo la predicción está más del lado del no abandono de clientes, lo cual se denota en la precisión y f1-score notoriamente. A pesar de tener un F1-score promedio de 0.79, se puede notar que el modelo posee un mayor rendimiento en la clasificación de los clientes que permanecen fieles al banco, en comparación con los que abandonan que es de 0.47.

Adicionalmente, no fue realizada curva de ROC. Esta se empezó a utilizar a partir de la iteración 2 con el fin de analizar a mayor profundidad los resultados de las diferentes técnicas de validación y evaluación de la generalización empleadas.

En relación con los principales problemas técnicos encontrados entre ellos se encuentran los siguientes:

- Desconocimiento del proceso
- Primera actividad de aplicación de lo aprendido
- Desconexiones constantes de Colab
- Entendimiento del tipo de errores presentado en cada paso
- Orden de ejecución y validación
- Entendimiento de otro tipo de herramientas como jupiter como solución a las desconexiones presentados por Colab
- No existencia de una metodología clara para abordar el problema
- Definición inadecuada del tema a abordar
- Falencia de conocimiento en algunos ítems debido a la no profundización adecuada de algunas materias.

4.2. Validación

Las validaciones fueron realizadas en la última iteración, correspondiente a la iteración 4, debido a que en este punto ya se tenían los mejores hiperparámetros definidos por cada algoritmo.

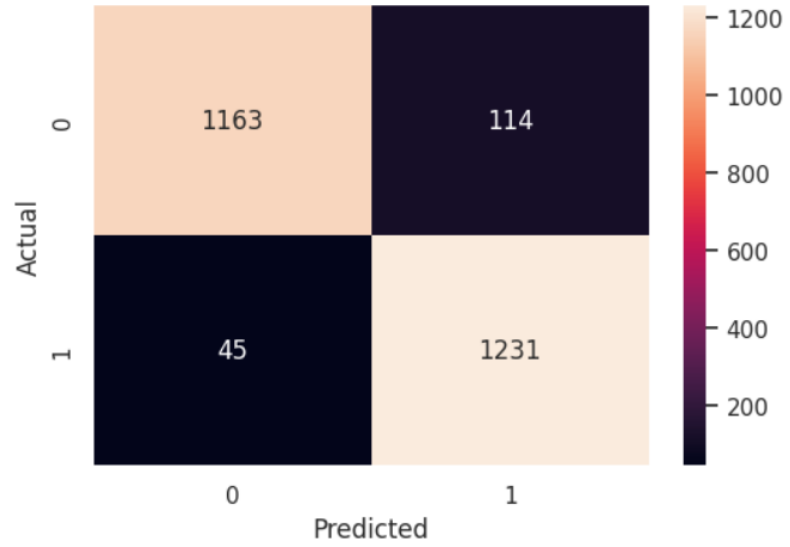
Con el fin de probar la capacidad de generalización y clasificación del modelo entrenado, se utiliza el proceso de Split, la cual es una técnica de división de datos, donde se particionó el conjunto de datos principal en dos subconjuntos distintos: uno destinado al entrenamiento del modelo y otro para su validación. En este, se seleccionó aleatoriamente el 80% de los datos para el conjunto de entrenamiento y el 20% restante para el conjunto de validación. Este procedimiento garantiza una representación significativa de datos para entrenar el modelo y, al mismo tiempo, reserva una porción sustancial para evaluar su rendimiento en datos no vistos.

De esta forma, los resultados obtenidos en la iteración 4 son presentados en la Tabla 4

Tabla 4. Resultados de Accurary con Split

Accuracy de los modelos	
Modelo	**Accuracy**
Support Vector Machine	0.93
Logistic Regression	0.92
K- Nearest neighbors	0.89
Complement Naive Bayes	0.77
Decision Tree Clasifier	0.90
Random Forest	0.94
MultiLayer Perceptron	0.94

La matriz de confusión nos muestra lo siguiente



	precision	recall	f1-score	support
0	0.96	0.91	0.94	1277
1	0.92	0.96	0.94	1276
accuracy			0.94	2553
macro avg	0.94	0.94	0.94	2553
weighted avg	0.94	0.94	0.94	2553

Figura 19 Matriz de confusión iteración 4

Se evidencia en esta gráfica el balanceo de la variable target y el buen resultado del f1-score

Como se puede observar el modelo que dio mejor fue el ModelNN seguido muy de cerca por Random Forest.

4.3. Iteraciones y evolución

Iteración 1

En esta primera etapa, se involucró la comprensión del problema y la preparación de datos, acá surgieron diversos desafíos y problemas. Por lo tanto, se presentan algunos de los problemas obtenidos en esta fase inicial:

- Falta de claridad sobre los objetivos del modelo y el problema que se está abordando.
- No explorar en profundidad los datos, no identificar patrones, desequilibrios o problemas de calidad de datos.
- Datos incompletos, incorrectos o desactualizados.
- No selección de las variables adecuadas para el modelo.

En la segunda iteración del análisis, se logró una comprensión más profunda y detallada del problema en cuestión, lo cual llevó a la implementación de un enfoque más estructurado y refinado para cada fase del proceso. Durante esta revisión exhaustiva del conjunto de datos, se identificaron y abordaron diversas áreas que permitieron optimizar la calidad y relevancia de la información utilizada en el análisis.

Uno de los aspectos cruciales de este proceso fue la identificación y eliminación de gráficas redundantes que, aunque se presentaron inicialmente, no aportaban de manera significativa al entendimiento general de la información. Este paso permitió simplificar la visualización de datos, enfocándose en representaciones visuales más claras y relevantes para el análisis.

Adicionalmente, se llevó a cabo una depuración cuidadosa de variables que no contribuían de manera sustancial al modelo de predicción. Entre estas variables se encontraban Clientnum,
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1:Naive Bayes classifier y Avg_Open_To_Buy. La exclusión de las dos primeras se basó en la falta de información relevante que ofrecían, mientras que la eliminación de Avg_Open_To_Buy se fundamentó en su escasa correlación con la variable objetivo.
Este proceso de refinamiento y depuración no solo contribuyó a la eficiencia computacional del análisis, sino que también mejoró la calidad de las características consideradas, permitiendo una representación más precisa de la relación entre las variables y la pérdida de clientes, objetivo principal del modelo predictivo.

En el transcurso de la evolución del proyecto, en la tercera iteración se concentró de manera específica en la tarea esencial de buscar y ajustar los hiperparámetros óptimos del modelo. Este paso reviste una importancia fundamental, ya que influye directamente en la capacidad del modelo para realizar predicciones con la mayor precisión posible.
La búsqueda y ajuste de hiperparámetros se traducen en la configuración de las características fundamentales del modelo, tales como la tasa de aprendizaje, la profundidad del árbol en un modelo de árboles de decisión o la elección del kernel en una Máquina de Vectores de Soporte (SVM). Cada uno de estos parámetros tiene un impacto único en cómo el modelo interpreta y aprende de los datos.
Optimizar estos hiperparámetros es crucial para lograr un equilibrio adecuado entre la capacidad de adaptación del modelo y su capacidad para generalizar patrones de manera efectiva en nuevos datos. Al encontrar los valores más adecuados, se mejora significativamente el rendimiento del modelo, lo que se traduce en predicciones más confiables y precisas. Este enfoque meticuloso en la búsqueda y ajuste de hiperparámetros contribuye directamente a la robustez y eficacia del modelo en la tarea específica de predecir la pérdida de clientes.

En la cuarta fase de desarrollo, se capitalizó la ventaja de contar con un conjunto de datos balanceado, lo que implica que las clases objetivo estaban representadas de manera equitativa. Además, se incorporó la técnica de validación cruzada, una estrategia robusta para evaluar el rendimiento del modelo, particularmente útil cuando se dispone de datos limitados o desbalanceados.

Durante esta iteración, se observó una proximidad considerable en los valores arrojados por dos modelos específicos, Random Forest y MLPClassifier. Aunque MLPClassifier continuó siendo reconocido como el modelo principal, la diferencia entre ambos fue mínima en esta instancia.

La implementación de la validación cruzada, utilizando la metodología KFold con 10 pliegues, permitió obtener resultados más robustos y generalizables. Este proceso se aplicó a los tres modelos con mejores desempeños, y para Random Forest se registró un valor impresionante de 0.96. Este resultado sugiere que el modelo Random Forest es altamente capaz de discriminar efectivamente entre las clases, destacando su fiabilidad y precisión en la tarea de clasificación.

En conjunto, la combinación de datos balanceados y la aplicación de técnicas avanzadas de validación cruzada fortalecieron la confianza en la capacidad predictiva de los modelos, proporcionando una evaluación más sólida de su rendimiento y su capacidad para generalizar patrones en nuevos conjuntos de datos.

5.4 Herramientas

Las herramientas utilizadas para la ejecución del presente proyecto fueron:

- Lenguaje de programación Python, versión 3.10.12
- Entorno de ejecución: Google Colab
- Librerías científicas tales como: Numpy, Scipy, Scikit-Learn, Pandas, Matplotlib

5. Resultados y discusión

El dataset sujeto de estudio, contiene información de clientes de tarjeta de crédito, con el objetivo de predecir la pérdida de usuarios por parte de un banco. Basado en las características proporcionadas se presentan las siguientes observaciones:

La columna Attrition_Flag es la variable objetivo, la cual indica si un cliente ha abandonado o no. Esta es la característica a predecir en los modelos.

Existen algunas características demográficas tales como Customer_Age, Gender, Dependent_count, Education_Level, Marital Status e Income_Category que nos muestran una idea del comportamiento de los clientes, de sus costumbres, modos de gasto, y de como de acuerdo a su nivel de vida realizan cierto tipo de gasto, esta información puede llegar a ser muy útil para generar campañas de acuerdo a estos niveles de vida.

En relación al historial de relación, características como Months_on_book y Total_Relationship_Count ofrecen información sobre la duración de la relación del cliente y la cantidad total de relaciones con el proveedor de la tarjeta de crédito. Estos se concluyen son indicadores importantes de la estabilidad de la relación.

En el punto de vista del comportamiento financiero, las variables Months_incative_12_mon, Contacts_Count_12_mon, Credit_limit, Total_Revolving_Bar y Avg_Utilization_Ratio proporcionan información sobre el comportamiento financiero del cliente.

Por último, se evidencia que los patrones de gasto (características como Total_Trans_Amt, Total_Trans_Ct) ofrecen información sobre el comportamiento de gastos de los clientes, mientras que Total_Amt_Chng_Q4_Q1 y Total_Ct_Chng_Q4_Q1 indican cambios en el gasto a lo largo del tiempo.

De la matriz de correlación (figura 10) se puede observar que las variables más correlacionadas con la variable target(Attrition_Flag) corresponden a Total_Trans_Ct y Total_Ct_Chng_Q4_Q1 las cuales corresponden a patrones de gasto de los clientes (ver figura 11), en la cual se puede ver claramente que después de cierta cantidad de transacciones, no hay abandono de clientes.

Otro hallazgos importantes se dan con las variables categóricas (ver figura 8):

-En cuanto al genero masculino y femenino, se observa un valor muy similar

-La mayor parte de las personas tienen un nivel de educación de graduado

-Aunque la mayor parte de las personas es casada, la cantidad de solteros tiene un valor muy cercano

-La gran mayoría de los clientes presentan un nivel de ingresos de menos de $40000

-Casi todos los clientes manejan la tarjeta Blue.

Es importante resaltar los resultados obtenidos antes y después del balanceo de la variable objetivo, los cuales obviamente presentan ecuanimidad una vez esta data se encuentra en balance.

5.1. Métricas

Resultados de métricas más relevantes en las iteraciones

Tabla 5. Mejor rendimiento de modelos entrenados

Iteración	Modelo	Accuracy	precision	recall	F1score	Observaciones
1	Random Forest	0.9465	0.90/0.97	0.91/0.96	0.88/0.98	Mejor modelo, variable objetivo desbalanceada
2	ModelNN	0.9244	0.97/0.89	0.88/0.97	0.92/0.93	Mejor modelo, variable objetivo desbalanceada
3	ModelNN	0.9067	0.97/0.86	0.84/0.97	0.90/0.91	Mejor modelo, mejores

						hiperparámetros, variable objetivo balanceada
4	ModelNN	0.9377	0.96/0.92	0.91/0.96	0.94/0.94	Mejor modelo, variable objetivo balanceada
Validación cruzada	Random Forest	0.96				

5.2. Evaluación cualitativa

En relación al Overfitting no se llegó a evidenciar ya que la realizar la validación cruzada y observar el comportamiento al ir iterando en los grupos de entrenamiento y prueba se observó un comportamiento muy similar.

En relación al underfitting tampoco se evidenció ya que el rendimiento no se observó bajo ni en entrenamiento y prueba.

A continuación, se muestran los valores obtenidos con un Split de 10 usando KFold en la validación cruzada para el algoritmo Random Forest:

Tabla 6. Valores obtenidos con KFold cross validation

iteración	1	2	3	4	5	6	7	8	9	10
Accurancy	0.870	0.869	0.935	0.971	0.986	0.996	0.993	0.998	0.994	0.992

Como se puede observar en la validación cruzada, estos valores tuvieron un rendimiento aceptable en cada iteración no importando la variabilidad entre data de entrenamiento y prueba.

5.3. Consideraciones de producción

Inicialmente fue usado Google Colab en la fase de desarrollo del proyecto, al entrar a producción y manejar data considerable se sugiere el uso de la siguiente arquitectura de nube Creación de cluster EMR autoescalable en aws la cual está diseñada para procesar y analizar grandes cantidades de información, adicionando a esta configuración un storage para almacenar la información el cual puede ser AMAZON Simple Storage Service (S3).

Hoy día, también existe la posibilidad de usar Amazon EMR Serverless, en el cual se puede desentender el usuario de la cantidad de nodos necesarios para la ejecución del código.

6. Conclusiones

El modelo Random Forest permitió identificar los factores más relevantes en la predicción de la deserción de clientes. Variables como la duración de la relación(cliente-banco), el comportamiento financiero y las características demográficas demostraron ser influyentes en la retención de clientes, como se puede observar en la tabla 4.

El modelo logró una buena precisión y recall en la clasificación de clientes desertores. Esto indica que, en general, el modelo tiene la capacidad de predecir tanto clientes que abandonan como aquellos que se quedan, minimizando los falsos positivos y falsos negativos.

Los resultados obtenidos se traducen en una aplicación práctica significativa para el banco. Proporcionan una herramienta eficaz que permite anticipar la deserción de clientes y emprender medidas preventivas, como el lanzamiento de campañas y la oferta de mejores condiciones para retener a los clientes.

Los resultados están alineados de manera directa con los objetivos establecidos en la formulación del problema. Se logró el desarrollo de un modelo predictivo eficiente que clasifica a los clientes entre los propensos a abandonar y los de alta probabilidad de retención, cumpliendo así con el propósito inicial de anticipar y abordar la deserción.

A pesar de los resultados prometedores, se reconoce la imperfección inherente a cualquier modelo. Se subraya la importancia de una supervisión y mejora continuas del modelo, especialmente a medida que se acumulan más datos y se enfrenta a nuevos desafíos. Se destaca la necesidad de evaluar críticamente la interpretación de los resultados y su aplicabilidad práctica en el contexto del negocio.

7. Recomendaciones

Se podría considerar la exploración de técnicas más avanzadas como el aprendizaje profundo, puesto que este podría capturar patrones más complejos en los datos que brinden información de interés para la identificación de ambos tipos de cliente. En esta, se podrían utilizar redes como AutoEncoders, Recurrent Neural Networks, Vanilla Networks, Attentional Networks y otras arquitecturas más especializadas para este fin, como los implementados por Btoush et. Al [6]. Asimismo, se podrían utilizar otras herramientas ampliamente utilizadas en la literatura como lo es el Transfer Learning, con el fin de utilizar pesos aprendidos y aplicarlos para el problema predictivo en cuestión. Además, sería beneficioso realizar análisis más detallados de los errores del modelo para entender mejor las situaciones en las que podría no funcionar tan bien y ajustar en consecuencia.

Referencias

[1] E. Ileberi, Y. Sun, and Z. Wang, “A machine learning based credit card fraud detection using the GA algorithm for feature selection”, doi: 10.1186/s40537-022-00573-8.

[2] N. Nazareth and Y. V. Ramana Reddy, “Financial applications of machine learning: A literature review,” *Expert Syst Appl*, vol. 219, p. 119640, Jun. 2023, doi: 10.1016/J.ESWA.2023.119640.

[3] R. Bin Sulaiman, V. Schetinin, and P. Sant, “Human-Centric Intelligent Systems (2022) 2:55-68 Review of Machine Learning Approach on Credit Card Fraud Detection,” vol. 1, p. 3, doi: 10.1007/s44230-022-00004-0.

[4] J. K. Afriyie *et al.*, “A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions,” *Decision Analytics Journal*, vol. 6, p. 100163, 2023, doi: 10.1016/j.dajour.2023.100163.

[5] S. Raj, S. Roy, S. Jana, S. Roy, T. Goto, and S. Sen, “Customer Segmentation Using Credit Card Data Analysis,” *Proceedings - 2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications, SERA 2023*, pp. 383–388, 2023, doi: 10.1109/SERA57763.2023.10197704.

[6] E. A. L. M. Btoush, X. Zhou, R. Gururajan, K. C. Chan, R. Genrich, and P. Sankaran, “A systematic review of literature on credit card cyber fraud detection using machine and deep learning,” *PeerJ Comput Sci*, vol. 9, p. e1278, Apr. 2023, doi: 10.7717/PEERJ-CS.1278/TABLE-7.