**Logotipo

Descripción generada automáticamente**

**Mejorando la Detección de Lavado de Dinero en el Sector Financiero mediante Aprendizaje Activo**

Cristian Javier Sánchez Álvarez

Sebastián Naranjo Torres

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

Asesor  
David Manuel Villanueva Valdés

Universidad de Antioquia  
Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Antioquia, Colombia

2024

**Tabla de contenido**

[Resumen 4](#_heading=h.3rdcrjn)

[Abstract 5](#_heading=h.26in1rg)

[1.](#_heading=h.35nkun2) 6

[1.1.](#_heading=h.1ksv4uv) 6

[1.2.](#_heading=h.44sinio) 7

[1.3.](#_heading=h.z337ya) 8

[1.4.](#_heading=h.3j2qqm3) 9

[2.](#_heading=h.4i7ojhp) 11

[2.1.](#_heading=h.2xcytpi) 11

[2.2.](#_heading=h.3whwml4) 11

[3.](#_heading=h.2bn6wsx) 12

[3.1.](#_heading=h.qsh70q) 12

[3.2.](#_heading=h.3as4poj) 14

[3.3.](#_heading=h.1pxezwc) **¡Error! Marcador no definido.**

**Lista de figuras**

**[Figura 1](#_heading=h.2r0uhxc)** [Diagrama de la Metodología CRISP-DM" 22](#_heading=h.2r0uhxc)

[**Figura 2** Vector de Alto Nivel Agregado 22](#_heading=h.1664s55)

# Resumen

El monitoreo de transacciones financieras es una obligación crítica en la lucha contra el lavado de dinero (AML) para las instituciones financieras. En los últimos años, los sistemas de monitoreo de transacciones basados en aprendizaje automático han complementado con éxito los sistemas tradicionales basados en reglas, reduciendo el alto número de falsos positivos y el esfuerzo necesario para revisar manualmente todas las alertas. Sin embargo, las soluciones basadas en aprendizaje automático también tienen desventajas: mientras que los modelos no supervisados pueden detectar nuevos patrones anómalos, generalmente se caracterizan por un alto número de falsas alarmas; los modelos supervisados, por otro lado, suelen ofrecer una mayor tasa de detección, pero requieren una gran cantidad de datos etiquetados para alcanzar dicho rendimiento. En esta primera parte de la investigación, proponemos un enfoque que integra el aprendizaje activo para la detección de anomalías, combinando técnicas de aprendizaje no supervisado y supervisado para mejorar los procesos de monitoreo de transacciones, aumentando el rendimiento de la detección y reduciendo los costos de gestión del cumplimiento. Para ello, utilizamos un conjunto de datos sintético con transacciones que simulan perfiles de clientes que operan en mercados de capitales internacionales, generando características agregadas que representan el comportamiento de los usuarios en diferentes ventanas temporales para el entrenamiento de los modelos.

*Palabras clave*: lavado de dinero, aprendizaje supervisado, aprendizaje no supervisado, aprendizaje activo.

# Abstract

Financial transaction monitoring is a critical obligation in the fight against money laundering (AML) for financial institutions. In recent years, machine learning-based transaction monitoring systems have successfully complemented traditional rule-based systems, reducing the high number of false positives and the effort needed to manually review all alerts. However, machine learning-based solutions also have drawbacks: while unsupervised models can detect new anomalous patterns, they are generally characterized by a high number of false alarms; supervised models, on the other hand, usually offer a higher detection rate but require a large amount of labeled data to achieve such performance. In this first part of the research, we propose an approach that integrates active learning for anomaly detection, combining unsupervised and supervised learning techniques to improve transaction monitoring processes, increasing detection performance, and reducing compliance management costs. To achieve this, we use a synthetic dataset with transactions that simulate profiles of clients operating in international capital markets, generating aggregated features that represent user behavior over different time windows for model training.

*Keywords: money laundering, supervised learning, unsupervised learning, active learning.*

Repositorio en GitHub: https://github.com/HerrSebas/monografia/tree/main

# Descripción del problema

## Problema de negocio.

En el mundo actual, el lavado de dinero afecta a todas las economías del mundo y es responsable de generar flujos financieros ilegales entre 1,6 y 2,85 billones de dólares al año, equivalente al 2,1%-4% del Producto Bruto Mundial. Al principio, se implementaban un conjunto de reglas que estaban configuradas para monitorear comportamientos inusuales predeterminados. Estas generaban alertas, por ejemplo, si la cantidad era mayor que 10.000.000 (umbral estático), entonces se generaba una alerta. Los beneficios de este enfoque eran la facilidad para interpretar el resultado del sistema y la capacidad de los expertos en la materia (es decir, analistas que trabajan en el área de detección de anomalías) de utilizar esa información fácilmente. La desventaja es que las técnicas de lavado de dinero y los delitos financieros siempre están evolucionando, por lo que las reglas debían actualizarse para garantizar que fueran adecuadas para reflejar estos cambios. Además, las reglas solo podían cubrir comportamientos anómalos conocidos y no podían detectar comportamientos inusuales desconocidos, lo que daba lugar a falsos negativos.

El aprendizaje automático superó estas dificultades de los sistemas basados en reglas. Los modelos de aprendizaje automático pueden extraer y analizar patrones e ideas a partir de datos y evaluar correlaciones inusuales desconocidas para los expertos en la materia. Los modelos de aprendizaje automático supervisados pueden clasificar las transacciones como normales o anómalas. Sin embargo, requieren una gran muestra de transacciones revisadas manualmente por los expertos en la materia (etiquetadas), lo que a su vez exige una gran cantidad de tiempo para cubrir todas las transacciones.

Como consecuencia de esto, las empresas del sector financiero deben mejorar la eficacia y eficiencia en la detección de transacciones fraudulentas relacionadas con actividades de lavado de dinero, mediante una recopilación más rápida y eficiente de conjuntos de datos etiquetados. Esto se logra, en parte, aprovechando el aprendizaje activo. El aprendizaje activo es una técnica que utiliza modelos de aprendizaje automático para seleccionar, de entre todas las transacciones, aquellas que tienen mayor probabilidad de mejorar el rendimiento del sistema de aprendizaje supervisado.

Para ello, se dispone de una vasta base de datos transaccionales simuladas, que incluye millones de operaciones realizadas por clientes finales que compran y venden activos específicos en los mercados.

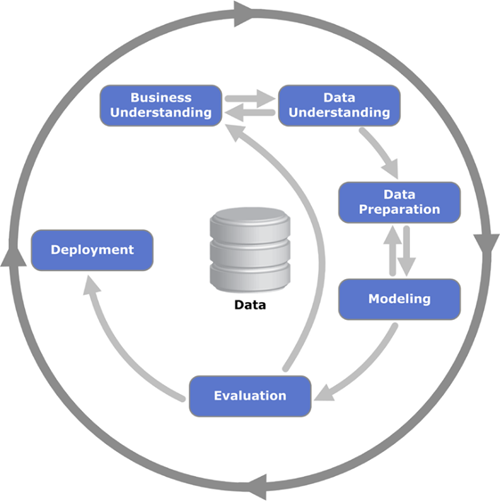
## Aproximación desde la analítica de datos.

Para abordar el problema de detección de anomalías en transacciones financieras mediante active learning, se seguirá un enfoque basado en la analítica de datos, utilizando modelos tanto de aprendizaje supervisado como no supervisado. Este enfoque se estructurará según la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), que proporciona un marco robusto para el análisis de datos en proyectos de minería de datos y aprendizaje automático.

El sistema combinará modelos no supervisados y supervisados organizados en un marco de trabajo donde un experto o analista también forma parte fundamental del proceso. El modelo no supervisado permite que el sistema detecte anomalías desconocidas y patrones nuevos que no se han observado previamente, mientras que el modelo supervisado puede utilizar etiquetas previamente clasificadas por expertos en la materia para mejorar la tasa de detección. Este enfoque híbrido permite que el sistema sea más robusto, ya que puede identificar tanto patrones emergentes como realizar una clasificación precisa de las transacciones con base en datos etiquetados.

**Figura 1.**

*El diagrama de la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining).*



En la fase de preparación de datos, las transacciones se agregan para generar características temporales y financieras que reflejan el comportamiento del usuario a lo largo del tiempo, utilizando ventanas temporales específicas para capturar patrones a corto, mediano y largo plazo. A continuación, se utilizarán los **scores** de anomalías generados por los modelos supervisado y no supervisado para seleccionar muestras, aplicando un conjunto de estrategias establecidas para guiar la revisión de las transacciones.

La evaluación del rendimiento de los modelos se realiza utilizando métricas de desempeño, comparando los resultados obtenidos con los objetivos establecidos y ajustando los modelos según sea necesario. La evaluación final se basará en la capacidad del modelo para aprender y mejorar con el tiempo, además de considerar la eficiencia lograda en términos de tiempo y esfuerzo para los analistas y/o expertos. Este enfoque no solo busca optimizar la detección de patrones anómalos, sino también reducir los costos operativos asociados a la gestión de estos análisis​

## Origen de los datos.

En el ámbito del lavado de activos, una de las principales limitaciones es la dificultad de acceder a un conjunto de datos real de las instituciones financieras, ya que suelen tener restricciones estrictas sobre la compartición de datos debido a preocupaciones de privacidad y regulaciones. Además, es aún más complicado conseguir un conjunto de datos etiquetado. Por lo tanto, utilizaremos un conjunto de datos sintéticos generado mediante un generador de datos personalizado, que simula perfiles de transacciones de clientes que realizan operaciones en mercados de capital internacionales.

Según menciona la fuente, los datos combinan más de 10.000 parámetros extrapolados de datos reales del mercado. Esto indica que las características del conjunto de datos (por ejemplo, variables como montos de transacciones, tiempos de ejecución y tipos de activos) se derivaron o modelaron a partir de patrones reales observados en el mercado financiero, pero el conjunto en sí no es una recopilación directa de datos reales de transacciones. Es decir, las transacciones sintéticas reflejan las tendencias y comportamientos del mundo real sin divulgar información confidencial o específica de instituciones financieras.

El conjunto de datos consta de 29 millones de transacciones ejecutadas por 400 clientes finales que compran y venden valores específicos en un mercado de capitales, distribuidas a lo largo de un periodo de 60 días, dividido en 12 semanas.

## Métricas de desempeño.

A continuación, se describen las métricas que se emplearán para estimar el desempeño de los modelos utilizados. Las métricas por emplear serán Accuracy, Precision, Recall, FPR, F-Score y MCC. Dado que el objetivo de este trabajo es identificar un modelo que aprenda de manera efectiva y que brinde eficiencia en el etiquetado de transacciones, se prestará especial atención al resultado proporcionado por la métrica F-Score y a las métricas de eficiencia definidas a continuación.

**Exactitud (Accuracy):** Es una métrica que mide la proporción de predicciones correctas en relación con el total de predicciones realizadas. Se calcula como la cantidad de predicciones correctas (verdaderos positivos + verdaderos negativos) dividida entre el número total de predicciones.

**Precisión (Precision):** Es una métrica que mide la proporción de predicciones positivas correctas respecto al total de predicciones positivas realizadas por el modelo. En otras palabras, indica cuántas de las instancias que el modelo clasificó como positivas realmente son positivas.

**Sensibilidad o TPR (Recall):** Es una métrica que mide la capacidad del modelo para identificar todas las instancias positivas en un conjunto de datos. Es decir, cuántos de los verdaderos positivos fueron correctamente detectados por el modelo.

**False Positive Rate (FPR):** Es una métrica que mide la proporción de instancias negativas que el modelo ha clasificado incorrectamente como positivas. Es decir, cuántos de los casos negativos fueron erróneamente identificados como positivos por el modelo.

**F–Score:** Es una métrica combinada que busca balancear la precisión (Precision) y el recall (también conocido como TPR o tasa de verdaderos positivos). En este caso es especialmente útil ya que existe un desbalance entre las clases.

**Matthews Correlation Coeficient (MCC):** es una métrica de evaluación utilizada principalmente en problemas de clasificación binaria, especialmente cuando las clases están desbalanceadas. Es una forma más equilibrada de evaluar el rendimiento de un modelo que las métricas tradicionales como la precisión (precision) o el recall, ya que tiene en cuenta todos los posibles resultados de la clasificación: verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).

# Objetivos

## Objetivo general.

Desarrollar un sistema robusto de detección de lavado de dinero en transacciones financieras, basado en aprendizaje activo y técnicas híbridas de aprendizaje supervisado y no supervisado, que permita mejorar la precisión en la identificación de patrones anómalos y reducir costos operativos, facilitando la eficiencia y eficacia en el monitoreo y análisis de transacciones sospechosas.

## Objetivos específicos.

* Implementar un sistema híbrido de detección de anomalías que combine modelos de aprendizaje supervisado y no supervisado, con el fin de identificar tanto patrones de transacción conocidos como nuevos comportamientos inusuales.
* Optimizar el proceso de selección de muestras para revisión mediante aprendizaje activo, aplicando estrategias para maximizar la calidad y representatividad de las muestras revisadas por expertos.
* Evaluar la efectividad del modelo en un entorno diario que implique que el modelo se entrene y ajuste constantemente, en función de las transacciones procesadas y etiquetadas cada día, simulando las condiciones reales de operación de una institución financiera mediante una base de datos sintética de transacciones en el mercado de capitales.
* Reducir la tasa de falsos positivos y el costo de revisión manual manteniendo un alto desempeño en términos de precisión, tasa de verdaderos positivos (TPR), y reducción de alertas falsas.

# Datos

## Datos originales.

El dataset original se distribuye en ficheros en formato .csv, organizados en columnas que contienen diferentes atributos relevantes para cada transacción. A continuación, se describe la estructura y el contenido de las columnas:

**Tabla 1.** *Estructura del dataset original.*



El conjunto de datos inicial consta de 29.704.090 transacciones ejecutadas por 400 clientes finales que compran y venden valores específicos en un mercado específico. Las transacciones se distribuyen uniformemente entre 12 semanas y la mayoría de ellas se ejecutan durante las horas de apertura del mercado, mientras que solo un pequeño porcentaje se ejecuta durante las primeras horas de la mañana y al final del día. Los sábados y domingos no están incluidos porque durante el fin de semana los mercados están cerrados.

Los campos clave de los datos seleccionados para este trabajo incluyen el monto de la transacción, la clase de producto (existen 17 tipos diferentes que representan los principales productos negociados en el mercado de capitales, como acciones y renta fija), el tipo de producto (por ejemplo, acciones en efectivo, futuros sobre acciones, bonos) y el campo de tiempo (fecha y hora de la transacción).

**Tabla 2.** *Composición del dataset original.*



Los conjuntos de datos financieros suelen estar extremadamente desequilibrados y, por lo general, contienen entre un 0,1 % y un 1 % de transacciones anómalas. Por lo tanto, para replicar escenarios del mundo real, se fijó el número de anomalías en menos del 1 % del total de los datos. También se generaron cinco clases de anomalías basadas en patrones sospechosos sugeridos por el GAFI que es un organismo intergubernamental que promueve la implementación efectiva de medidas legales, regulatorias y operativas para combatir el lavado de dinero. Se supone que las transacciones anómalas seguirán patrones similares a los de las transacciones normales, con el objetivo de mantener la ilusión de que estas anomalías están ocultas dentro del conjunto de datos.

* **Transacciones pequeñas, pero altamente frecuentes generadas en un corto período de tiempo:** Un patrón que contiene múltiples transacciones con montos por debajo de umbrales definidos.
* **Transacciones con montos redondeados normalizados comprados o vendidos dentro de una cuenta:** Es inusual que las transacciones en los mercados de capitales tengan montos redondeados).
* **Compra o venta de valores en un momento inusual:** Es inusual que los clientes comercien con valores específicos fuera de un marco temporal determinado (por ejemplo, fuera del horario de apertura y cierre de una bolsa de valores).
* **Retiro grande de activos:** Un aumento repentino en el monto de una transacción retirada de una cuenta o transferida fuera de ella, que se desvía de la actividad transaccional previa y carece de una justificación comercial o de un evento relacionado con acciones corporativas.
* **Una cantidad inusualmente grande de activos transferidos dentro y fuera de una cuenta en un corto período de tiempo:** Este comportamiento es inusual si se desvía del comportamiento transaccional previo del cliente.

Finalmente, el conjunto de datos tiene un tamaño aproximado de 3.3 GB (29,000,000 filas x 12 columnas).

## Datasets.

* + 1. **Construcción de datasets de Entrenamiento y Validación.**

Para crear los datasets de entrenamiento y validación a partir de los datos originales, se genera un conjunto de características agregadas derivadas de los datos transaccionales iniciales. Estas características se pueden construir utilizando ventanas de agregación específicas que capturan el comportamiento a corto, mediano y largo plazo del usuario. Se utilizará una ventana de agregación de 1 hora para captar comportamientos en intervalos de tiempo muy cortos.

**Características temporales:**

* A partir de la columna **FechaRegistro**, se extraen nuevas variables como el día de la semana (**Weekday**) y la hora (**Hour**), que permiten identificar patrones temporales de la transacción.
* Las transacciones también se categorizan en tres períodos del día: Mañana (6:00 - 11:59), Tarde (12:00 - 17:59) y Noche (18:00 - 5:59), creando indicadores binarios (**Morning**, **Evening**, **Night**) que facilitan el análisis de la actividad en estos segmentos horarios.

**Condiciones especiales en el monto:**

* Se identifican transacciones con montos redondeados, utilizando una condición que detecta si el monto normalizado (**MontoNormalizado**) tiene varios ceros en las posiciones decimales.
* También se detectan transacciones con montos pequeños por debajo de cierto umbral que cumplen con ciertas condiciones. Estas transacciones pueden ser indicadores de comportamientos inusuales, particularmente cuando corresponden a acciones como retiros o movimientos de efectivo.

**Condiciones basadas en el tipo de transacción:**

* Se distingue entre transacciones de compra (**Buy**) y venta (**Sell**), generando dos condiciones binarias (**Sell\_Condition** y **Buy\_Condition**) para cada tipo.
* Se calcula una variable **InputOutput\_Delta**, que refleja la diferencia entre las operaciones de entrada y salida. Esto permite obtener un indicador general del flujo financiero (positivo para compras y negativo para ventas) para cada cliente.

**Agregación de transacciones:**

* Para lograr una visión integral del comportamiento transaccional de cada usuario, las transacciones se agrupan de manera sistemática, teniendo en cuenta diversas dimensiones clave. En particular, se realiza una agrupación por cliente, día de la semana, hora, períodos del día y el indicador de anomalía. Esta segmentación permite analizar patrones de actividad en distintos momentos del día, capturando tanto los comportamientos regulares como las posibles irregularidades o anomalías.
* A partir de cada grupo, se generan estadísticas agregadas que ofrecen un perfil detallado de la actividad del cliente, considerando tanto el volumen y frecuencia de las transacciones como el tipo y clase de productos financieros involucrados. Además, esta estructura de agrupación facilita el cálculo de medidas específicas relacionadas con transacciones inusuales, como montos pequeños o redondeados, y permite una evaluación detallada de las diferencias entre operaciones de compra y venta. Este proceso de agregación proporciona una base sólida para identificar patrones de comportamiento atípicos y contribuye a la detección eficaz de actividades sospechosas.
  + 1. **Resultado.**

El proceso de agregación genera un conjunto de datos con características agregadas que capturan la 'firma' de comportamiento de cada usuario. Estos vectores están diseñados específicamente para el módulo de detección de anomalías, ya que incluyen información clave que refleja patrones transaccionales relevantes, lo que facilita la identificación de actividades inusuales o sospechosas.

A continuación, se presenta un ejemplo que muestra las características principales derivadas del comportamiento agregado de un usuario.

**Tabla 3.** *Estructura del dataset agregado.*

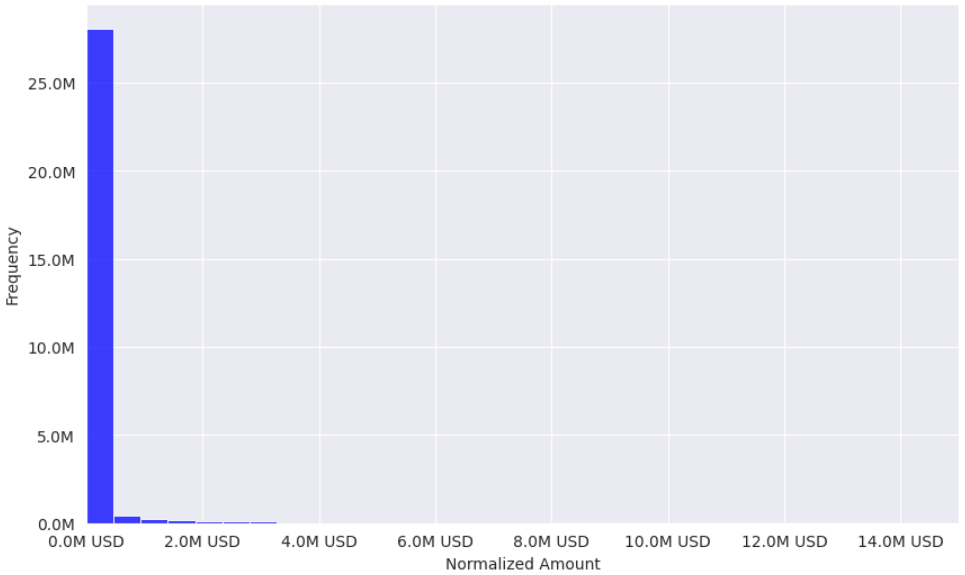


## Analítica descriptiva.

# 

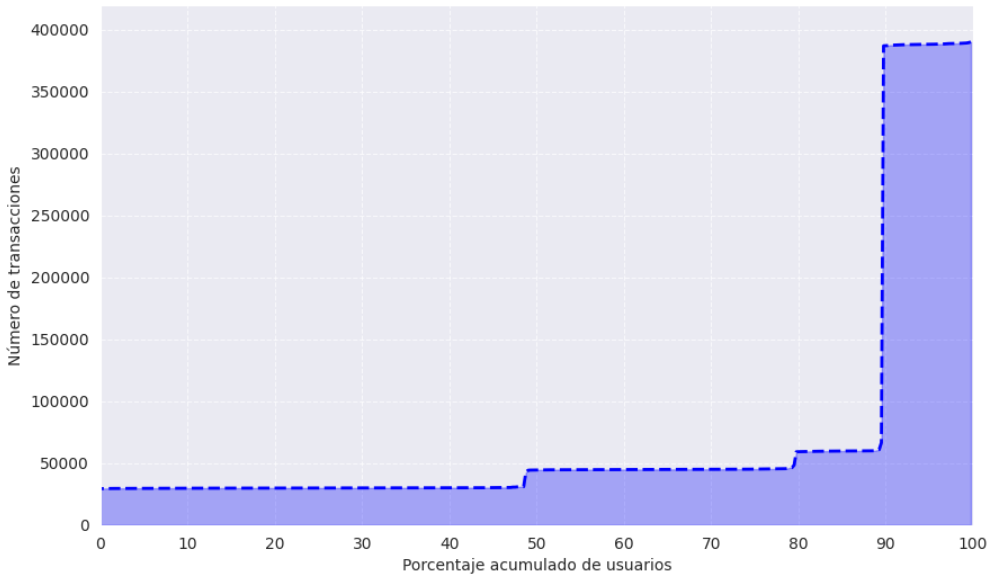
En el dataset inicial, los datos no poseen valores duplicados ni nulos, y la única variable numérica está normalizada para facilitar comparaciones. Sin embargo, es posible extraer algunos datos interesantes.

**Figura 2.** *Histograma de frecuencias del monto de las transacciones.*



El 96% de las transacciones tienen un monto inferior a 1 millón de USD, y el 57% de ellas tienen un monto inferior a 10,000 USD.

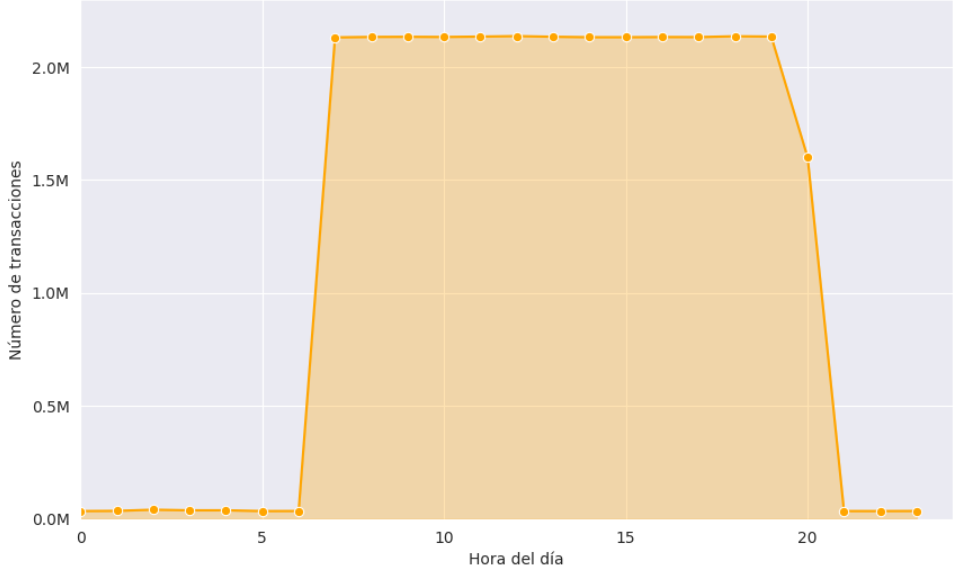
**Figura 3.** *Distribución acumulada del número de transacciones por usuario.*



Aproximadamente el 90% de los usuarios realizaron menos de 50,000 transacciones, mientras que el 10% superior ejecutó alrededor de 400,000 transacciones, lo que indica que este 10% representó casi el 50% del total de las transacciones.

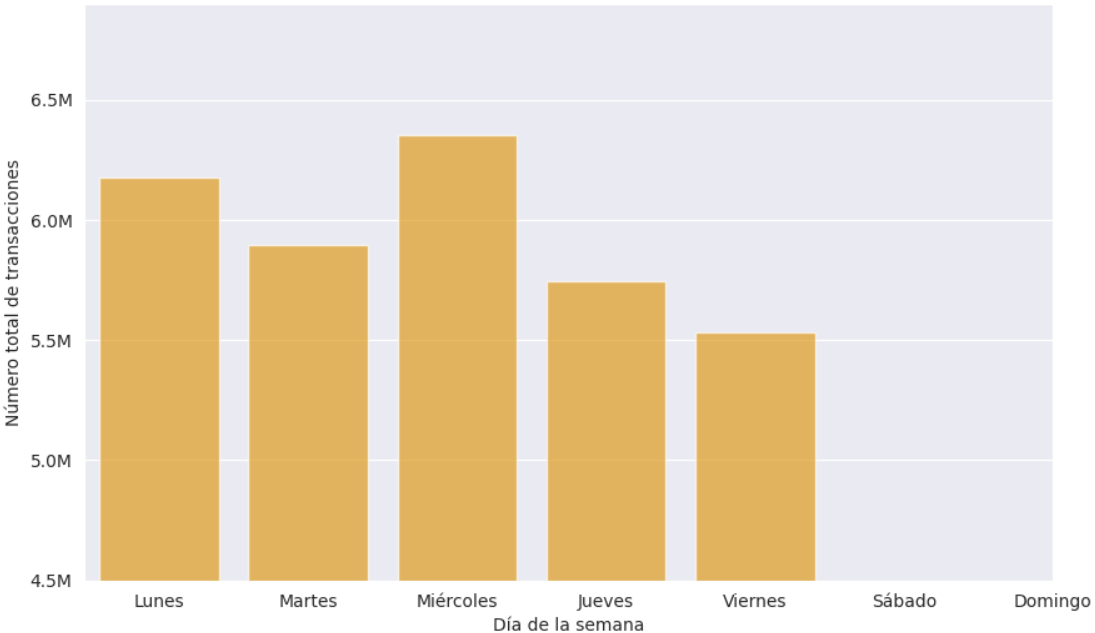
Al construir el dataset agregado que se utilizará para entrenar los modelos, podemos obtener características representativas de los datos más fácilmente. Este enfoque no solo mejora la capacidad de los modelos para capturar patrones relevantes, sino que también optimiza el proceso de entrenamiento al reducir la cantidad y complejidad de los datos, enfocando el modelo en las características más significativas para la predicción.

**Figura 4.** *Distribución de transacciones por hora del día.*

****

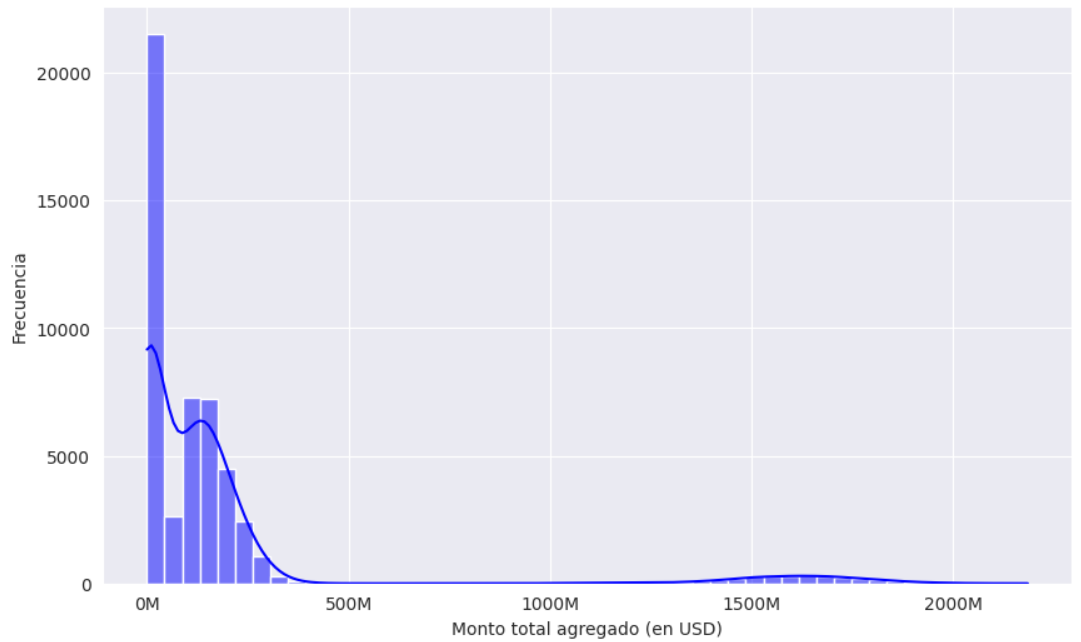
Las transacciones se distribuyen uniformemente tanto en número como en monto total negociado durante el horario de apertura del mercado, mientras que solo un pequeño porcentaje se ejecuta durante las primeras horas de la mañana y al final del día.

**Figura 5.** *Distribución de transacciones por día de la semana.*



Las transacciones también se distribuyen uniformemente a lo largo de los cinco días de la semana. Los sábados y domingos no están incluidos, ya que durante el fin de semana los mercados están cerrados.

**Figura 6.** *Distribución del monto total en el dataset agregado.*



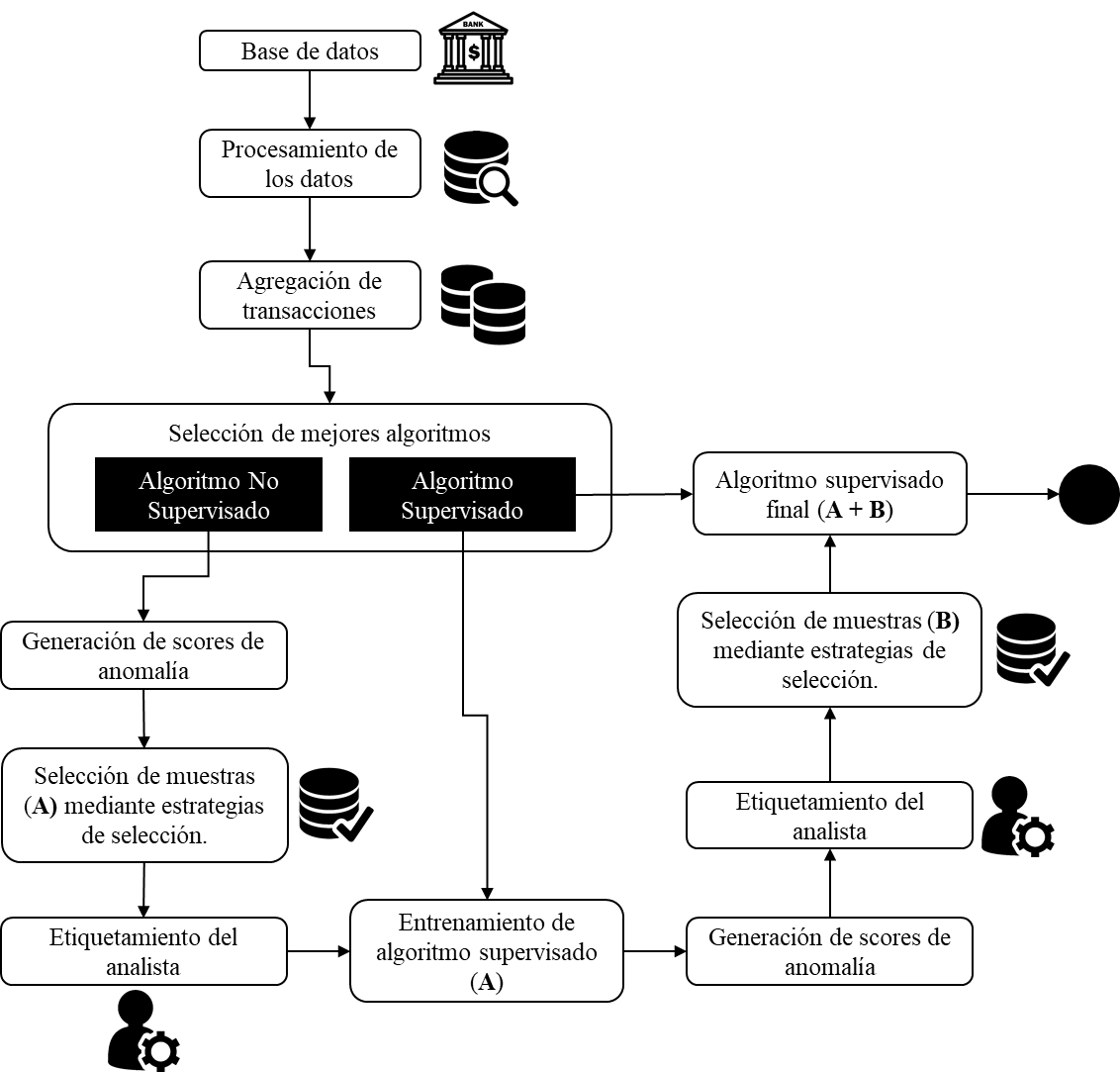
Al agrupar los datos en ventanas de tiempo, los patrones ocultos en el comportamiento de cada cliente se vuelven más evidentes y ciertos puntos se destacan como valores atípicos en la distribución del monto. Sin embargo, dado que estamos abordando un problema de detección de anomalías, es fundamental conservar estos valores atípicos, ya que son precisamente los puntos que los algoritmos deben identificar como posibles anomalías.

**Tabla 4.** *Numero total de anomalías en los datos.*



1. **Proceso de analítica**
   1. **Pipeline principal.**

**Figura 8.** *Pipeline principal.*



* 1. **Preprocesamiento.**

Dado que en cada iteración diaria se espera que el tamaño de los datos de entrenamiento y prueba cambien de tamaño significativamente, es crucial adoptar técnicas de preprocesamiento que minimicen el impacto de estas fluctuaciones en el rendimiento del modelo. Esta variabilidad surge porque el volumen de transacciones financieras cambia de un día a otro, lo cual hace que el conjunto de entrenamiento sea cada vez más grande a medida que transcurren los días. Estas diferencias en el tamaño de los conjuntos de datos pueden introducir sesgos en las características agregadas, afectando negativamente la precisión de los modelos de detección de anomalías.

Para abordar este problema, hemos utilizado la **media** en lugar de otras medidas, como la suma o el conteo, en las variables de nuestro dataset agregado. La media permite que las características sean comparables entre los conjuntos de entrenamiento y prueba, independientemente de la cantidad total de transacciones presentes en cada conjunto en una iteración específica. Esto ayuda a que el modelo no interprete incorrectamente variaciones en el tamaño de los conjuntos de datos como patrones de comportamiento.

Por ejemplo:

* En lugar de usar el total de transacciones pequeñas, se calcula el promedio de estas en relación con el conjunto total de transacciones, obteniendo la variable **Transactions\_Count\_Small\_Amount**.
* De manera similar, la proporción de transacciones de montos redondeados o de tipo “compra” se representa como un promedio, utilizando variables como **Transactions\_Count\_Round\_Amount** y **Transactions\_Count\_Buy**.

Este enfoque basado en la media ayuda a crear un dataset agregado robusto y balanceado, asegurando que los modelos se centren en las características de comportamiento transaccional y no en las variaciones del tamaño de los datos entre el entrenamiento y prueba, lo cual es fundamental para el éxito de los resultados del modelo.

* 1. **Modelos.**
     1. **Comparativa de algoritmos supervisados.**

Al enfrentarnos a un problema de clasificación, es crucial seleccionar el algoritmo de aprendizaje supervisado adecuado. Una forma efectiva de iniciar esta evaluación es mediante una tabla que resuma la complejidad temporal de varios algoritmos. Esto nos proporciona una visión general de las opciones disponibles y nos ayuda a identificar cuáles podrían ser más óptimas en términos de eficiencia computacional, tiempo de entrenamiento y capacidad de predicción.

**Tabla 5.** *Complejidades temporales de algoritmos supervisados.*

|  |  |  |
| --- | --- | --- |
| **Algoritmo** | **Complejidad Temporal** | **Descripción** |
| Naive Bayes |  | *n:* Número de muestras, *m:* Número de características. |
| Decision Tree |  | *n:* Número de muestras. |
| Random Forest |  | *n:* Número de muestras, *m:* Número de características. |
| CatBoost |  | *n:* Número de muestras, *k:* Número de iteraciones. |
| SVM |  | *n:* Número de muestras, *d:* Número de características. |
| LightGBM |  | *n:* Número de muestras. |

En la tabla de complejidad temporal, podemos observar que Naive Bayes es generalmente el algoritmo más rápido, con una complejidad de lo que lo hace adecuado para conjuntos de datos grandes con múltiples características. Por otro lado, Support Vector Machine (SVM) presenta la mayor complejidad, , lo que puede resultar en tiempos de entrenamiento significativamente más largos, especialmente con grandes cantidades de datos.

Random Forest y LightGBM ofrecen un equilibrio interesante, con complejidades de y respectivamente. Estos algoritmos son conocidos por su buena capacidad de predicción y, al mismo tiempo, manejan razonablemente bien la eficiencia computacional.

Sin embargo, al comparar diferentes algoritmos, es importante considerar no solo la complejidad temporal, sino también métricas de rendimiento como precisión, recall y F1-score. Esto nos permitirá tomar decisiones informadas sobre cuál modelo es el más adecuado para nuestras necesidades específicas. Para llevar a cabo lo anterior, entrenamos y ajustamos todos los modelos en el conjunto de datos de entrenamiento, compuesto por el 70% (--- transacciones) de los datos, y evaluamos el rendimiento en el 30% (--- transacciones) restante.

**Tabla 5.** *Comparativa del rendimiento de algoritmos supervisados.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Métrica** | **Random Forest** | **CatBoost** | **Decision Tree** | **Naive Bayes** | **LightGBM** |
| Accuracy | **\*0.998** | 0.9938 | 0.9971 | 0.968 | 0.9969 |
| Precision | **\*0.9814** | 0.8804 | 0.9425 | 0.5758 | 0.9499 |
| F1-Score | **\*0.9702** | 0.9278 | 0.9581 | 0.595 | 0.9573 |
| True Positive Rate | **\*0.998** | 0.9938 | 0.9971 | 0.968 | 0.9969 |
| False Positive Rate | **\*0.0004** | 0.0012 | 0.0006 | 0.0064 | 0.0006 |
| False Negative Rate | **\*0.002** | 0.0062 | 0.0029 | 0.032 | 0.0031 |
| True Negative Rate | **\*0.9996** | 0.9988 | 0.9994 | 0.9936 | 0.9994 |
| MCC | **\*0.9669** | 0.9091 | 0.9529 | 0.5675 | 0.9501 |
| Training Time (s) | 5.3929 | 165.5975 | 0.3265 | **\*0.0281** | 3.2283 |

El modelo Random Forest se destaca como el mejor clasificador en términos generales según los resultados obtenidos. Presenta la mayor precisión (0.998), superando a los otros modelos en cuanto a exactitud en las predicciones. Además, muestra un excelente equilibrio entre las métricas de rendimiento, como el F1-Score (0.9702), una de las más altas, y una tasa de falsos positivos mínima (0.0004), lo que indica que casi no comete errores al identificar las clases negativas.

Si bien Random Forest no es el modelo más rápido (su tiempo de entrenamiento es de 5.39 segundos), sigue estando entre los más eficientes en cuanto a tiempo de procesamiento, solo por detrás de Naive Bayes y LightGBM. Este rendimiento lo convierte en una opción competitiva, ya que ofrece una combinación ideal entre velocidad y precisión.

Por otro lado, modelos como CatBoost y Decision Tree también ofrecen buenos resultados en algunas métricas, pero no logran alcanzar la consistencia y equilibrio que muestra Random Forest. En contraste, Naive Bayes tiene un rendimiento significativamente más bajo en métricas clave como la precisión (0.5758) y el F1-Score (0.595), a pesar de ser el más rápido.

* + 1. **Comparativa de algoritmos no supervisados.**

Presentamos igualmente la tabla de complejidad temporal de algoritmos no supervisados para ayudar a guiar en el proceso de selección del mejor algoritmo no supervisado.

**Tabla 6.** *Complejidades temporales de algoritmos supervisados.*

|  |  |  |
| --- | --- | --- |
| **Algoritmo** | **Complejidad** | **Descripción** |
| Isolation Forest |  | *n:* número de muestras, *log n:* profundidad de los árboles. |
| Autoencoder |  | *n:* número de muestras, *m:* dimensiones de entrada, *e:* número de épocas de entrenamiento. |
| One-Class SVM |  | *n:* número de muestras, lo que implica alta complejidad en grandes conjuntos de datos. |
| K-Nearest Neighbors |  | *n:* número de muestras, *d:* dimensiones de los datos; se calcula la distancia a todos los puntos. |
| Local Outlier Factor |  | *n:* número de muestras, *d:* dimensiones, *log n:* búsqueda de vecinos cercanos. |

Según la tabla, el algoritmo más rápido es Isolation Forest, lo que lo hace eficiente para grandes conjuntos de datos debido a su estructura de árboles y profundidad logarítmica. Este algoritmo es ideal cuando se requiere escalabilidad. Por otro lado, One-Class SVM es el más lento, lo que lo hace menos práctico para conjuntos de datos grandes, ya que su tiempo de ejecución aumenta significativamente con el número de muestras.

Autoencoder tiene una complejidad de donde m representa las dimensiones de entrada y e el número de épocas de entrenamiento, lo que lo vuelve más dependiente de la estructura de los datos y el número de épocas. Aunque no es tan costoso como One-Class SVM, puede volverse más lento con un alto número de dimensiones o épocas.

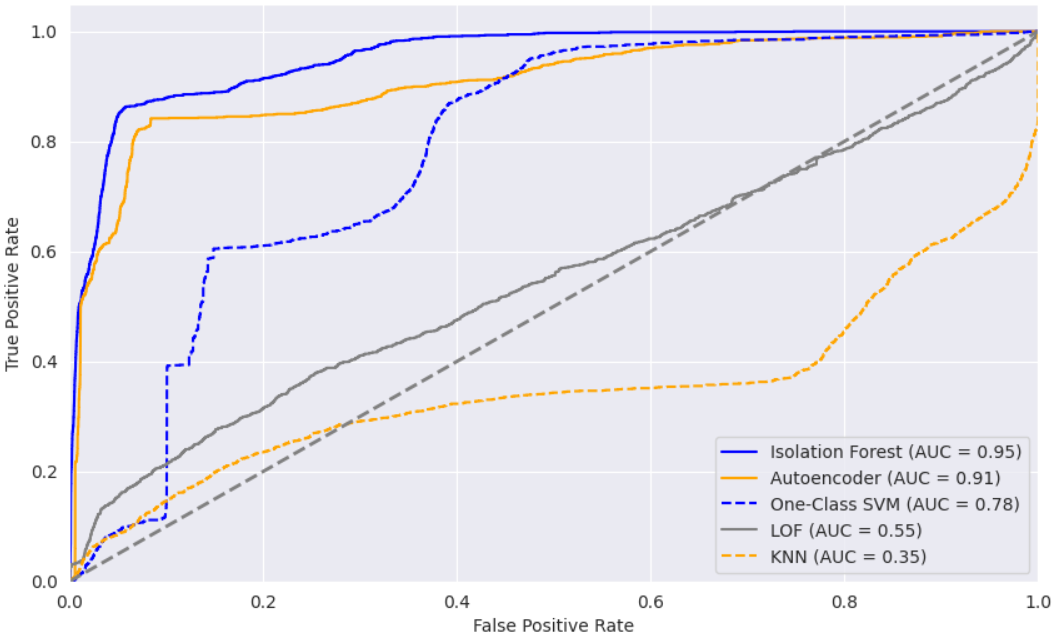
**Tabla 6.** *Comparativa del rendimiento de algoritmos no supervisados.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Métrica** | **Isolation Forest** | **Autoencoder** | **One-Class SVM** | **K-Means** | **LOF** |
| Accuracy | **\*0.9661** | 0.9566 | 0.8521 | 0.9243 | 0.9476 |
| Precision | **\*0.7243** | 0.6814 | 0.5474 | 0.5111 | 0.5417 |
| F1-Score | **\*0.7488** | 0.7207 | 0.5518 | 0.5124 | 0.5394 |
| True Positive Rate | 0.5821 | **\*0.607** | 0.5424 | 0.0841 | 0.1003 |
| False Positive Rate | **\*0.0216** | 0.0322 | 0.138 | 0.0489 | 0.0254 |
| False Negative Rate | 0.4179 | **\*0.393** | 0.4576 | 0.9159 | 0.8997 |
| True Negative Rate | **\*0.9784** | 0.9678 | 0.862 | 0.9511 | 0.9746 |
| MCC | **\*0.5014** | 0.4566 | 0.1958 | 0.0279 | 0.0791 |
| Training Time (s) | 1.4414 | 31.0427 | 31.0427 | **\*0.1172** | 1.4186 |

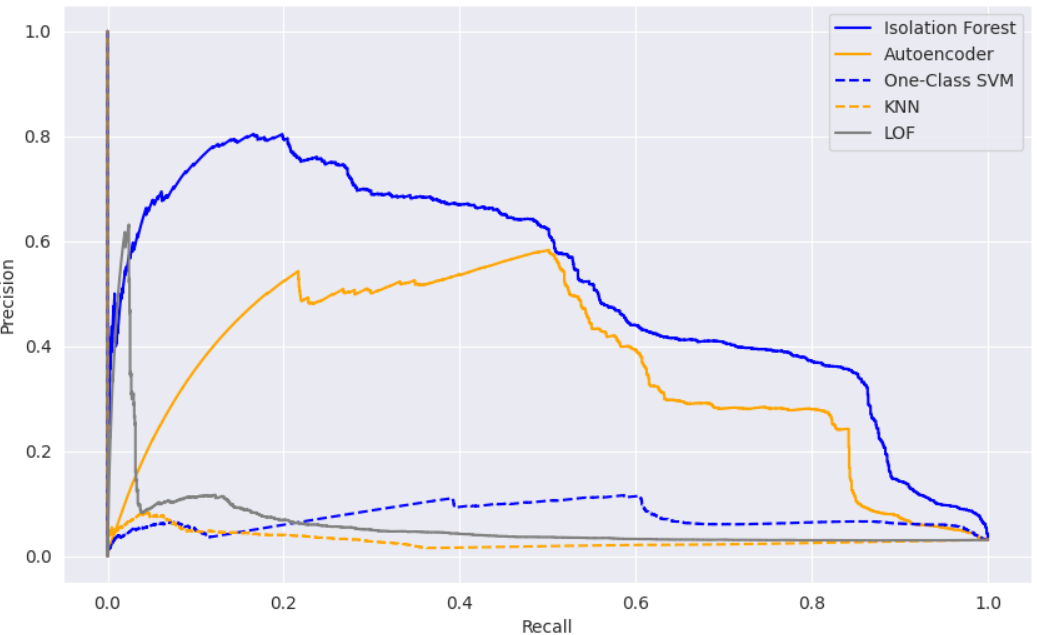
Como se observa en los resultados, el Isolation Forest es el algoritmo con el mejor desempeño general. Presenta una alta accuracy de 0.9661, la mejor precision (0.7243) y F1-Score (0.7488), lo que indica su capacidad para equilibrar correctamente la identificación de anomalías y transacciones normales. Además, tiene una MCC (coeficiente de correlación de Matthews) de 0.5014, que es la más alta entre todos los algoritmos, lo que refleja una excelente correlación entre las predicciones y los valores reales.

Aunque el Autoencoder también ofrece buenos resultados en cuanto a accuracy (0.9566) y True Positive Rate (0.6070), su tiempo de entrenamiento es significativamente mayor, lo que lo hace menos eficiente en comparación con Isolation Forest.

**Figura 11.** *Curvas ROC de algoritmos no supervisados.*



**Figura 10.** *Curvas precision-recall de algoritmos no supervisados.*



* 1. **Métricas.**

Para calcular las métricas para evaluar los algoritmos utilizados se utiliza la librería metrics de sklearn.

1. **Metodología**
   1. **Baseline.**

El aprendizaje activo es un método de aprendizaje automático que detecta y reconoce de manera eficiente los atributos de los datos utilizando sus elementos diferenciadores. Su objetivo es maximizar el rendimiento del modelo y, al mismo tiempo, reducir los costos de etiquetado, a pesar de su susceptibilidad a cambios impredecibles.

El aprendizaje activo selecciona muestras con características distintivas utilizando un marco bien estructurado que combina varias técnicas de selección de muestras. La selección de un grupo de ejemplos no etiquetados del conjunto de datos es el primer paso en el proceso del aprendizaje activo. Un experto humano selecciona y anota los puntos de datos más informativos de este grupo, reentrenando el algoritmo. Este proceso iterativo continúa hasta que se alcanza el nivel de precisión requerido.

Por otro lado, la precisión de las predicciones no solo depende del modelo de aprendizaje automático, sino también del conjunto de entrenamiento utilizado para el ajuste. El conjunto de entrenamiento debe representar la variedad completa de transacciones para evitar extrapolaciones durante la evaluación de las transacciones. Un algoritmo de selección práctico debe elegir las transacciones para el conjunto de entrenamiento únicamente con base en los datos no etiquetados, ya que en la práctica queremos calcular las etiquetas solo después de la selección. Los enfoques relacionados con la construcción (o selección) de un conjunto de entrenamiento óptimo se conocen como enfoques de aprendizaje activo.

Se implementan varios algoritmos de Machine Learning para respaldar el proceso de etiquetado interactivo en las estrategias de selección de muestras:

* **Primera estrategia:**

1. Sea el puntaje de anomalía de cada vector de alto nivel en el conjunto de datos .
2. Ordenamos en orden descendente según :
3. Seleccionar las muestras más anómalas:
4. Seleccionar las muestras menos anómalas:
5. Conjunto final de muestras para revisión es:

*​*

Primero, se seleccionan las muestras en los extremos del espectro de anomalía, capturando tanto las transacciones más como las menos anómalas. Este enfoque ayuda a representar casos altamente sospechosos, así como aquellos que parecen muy normales, lo cual es útil para afinar el modelo y minimizar la extrapolación. Para ello, se utiliza el score de anomalía entregado por el algoritmo Isolation Forest, seleccionando las muestras con los puntajes de anomalía más altos (top) y las con los más bajos (bottom) a partir del conjunto de datos .

* **Segunda estrategia:**

Dado que la estrategia anterior puede no garantizar que se cubran todos los tipos de anomalías (es decir, las anomalías principales por puntaje de anomalía pueden pertenecer todas al mismo tipo de anomalía) se intenta diversificar el tipo de patrones inusuales que se seleccionan mediante la agrupación vectores de alto nivel similares en función de la puntuación de anomalía.

Aplicamos clustering mediante el algoritmo de HDBSCAN para agrupar muestras similares. Posteriormente, priorizamos los clústeres menos densos, que tienden a contener patrones atípicos y así extraer muestras de cada grupo**.** Esta estrategia es útil para evitar sesgos hacia ciertos tipos de anomalías y asegurar que el modelo pueda detectar diversas configuraciones anómalas.

1. Filtrar las muestras con puntajes de anomalía altos. Seleccionamos el subconjunto que contiene solo las muestras con un puntaje de anomalía superior a un percentil definido :
2. Aplicar clustering. Aplicamos un algoritmo de clustering (específicamente HDBSCAN) a ​ para dividirlo en clústeres de acuerdo con su similitud:
3. Ordenar clústeres por densidad. Denotamos la densidad de cada clúster como . Ordenamos los clústeres de menor a mayor densidad, ya que los clústeres menos densos son más propensos a contener anomalías:
4. Seleccionar muestras de cada clúster. A partir de los clústeres menos densos, seleccionamos una cantidad fija de muestras de cada clúster hasta alcanzar el número total de muestras deseado :

Donde​ y *m* es el número de clústeres necesarios para cubrir el total .

* **Tercera estrategia:**

La tercera estrategia está orientada a reducir la incertidumbre en el modelo Random Forest, que se entrena inicialmente sobre el conjunto combinado (muestras de top, bottom y diversas de clústeres). La incertidumbre se mide en función de la proximidad del puntaje de probabilidad del modelo al valor central de 0.5, conocido como el valor de máxima entropía. Las muestras cuya puntuación se aproxima a 0.5 indican una alta incertidumbre, ya que el modelo no puede clasificarlas con claridad como normales o anómalas. Por lo tanto, selecciona las

muestras más cercanas a este valor para revisión, optimizando la capacidad del modelo para aprender de casos inciertos y mejorar su discriminación en futuras clasificaciones.

1. Definimos como el conjunto combinado de muestras seleccionadas desde las muestras anteriores:
2. Entrenamos un modelo de Random Forest sobre para obtener el puntaje de probabilidad para cada muestra del conjunto de datos inicial
3. Definimos el puntaje de entropía para cada muestra en *Ut* como la distancia a 0.5:
4. Ordenamos y seleccionamos las muestras con mayor incertidumbre:

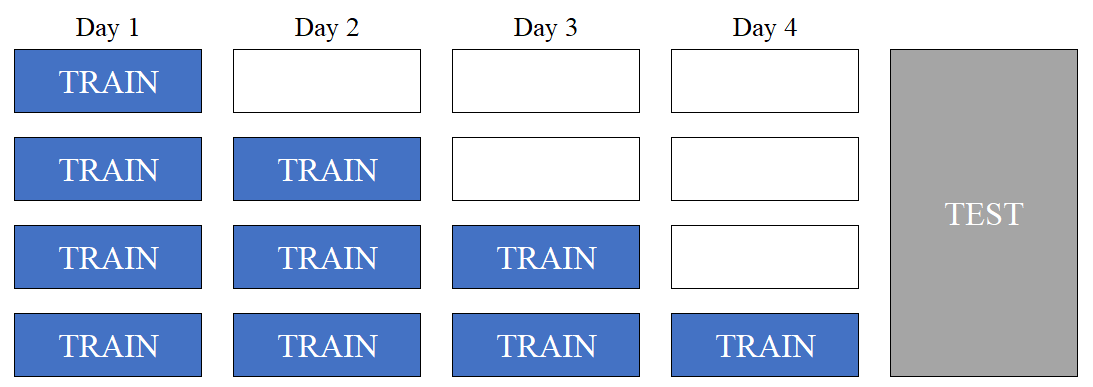
* **Cuarta estrategia:**

Finalmente, buscamos identificar muestras en las que existe mayor discrepancia entre el modelo de Isolation Forest y el modelo Random Forest, lo que indica conflicto en la predicción. Para cada muestra, se calcula la diferencia absoluta entre el puntaje de anomalía de Isolation Forest y el puntaje de probabilidad del Random Forest. Las muestras con la discrepancia más alta se consideran prioritarias para revisión, ya que reflejan casos en los que los modelos no están de acuerdo. Este enfoque permite que el sistema aprenda de estos casos ambiguos, logrando una mejora continua y mejor ajuste del modelo a patrones complejos de anomalía.

1. Calculamos el puntaje de conflicto para cada muestra *∈* basado en la discrepancia entre Isolation Forest y el Random Forest inicial entrenado sobre :
2. Ordenamos y seleccionamos las muestras con mayor conflicto entre los modelos:
   1. **Validación.**

Dado el componente temporal de los datos, en el experimento de evaluación del modelo, utilizamos un entorno realista que involucra un enfoque de prueba progresiva, como se ilustra en la Figura X. Este enfoque permite realizar una evaluación exhaustiva del sistema en una rutina de trabajo diaria, similar a un escenario del mundo real, en el cual un grupo de expertos debe investigar un conjunto de casos anómalos cada día. Los datos de prueba se dividen por día, y en cada jornada, un número K de muestras anómalas es seleccionado por el módulo no supervisado para su revisión por parte de los analistas. Según el tipo de anomalía predicho por el sistema de detección, se procederá al entrenamiento del módulo supervisado.

**Figura 11.** *Representación grafica de la evaluación del modelo.*

****

Una vez que los analistas revisan las muestras detectadas como anómalas y etiquetan las que realmente son anomalías, estas muestras etiquetadas se agregan a las muestras anteriores que ya fueron recolectadas en días anteriores. Esto crea una base de datos acumulativa de ejemplos de anomalías y sus etiquetas correspondientes. Al integrar las muestras anómalas de cada día con las de días anteriores, el sistema puede aprender de las anomalías pasadas y adaptarse a nuevas formas de anomalía que puedan surgir con el tiempo. Esto permite que el modelo evolucione y no se quede obsoleto, ya que se entrena continuamente con los casos más recientes y relevantes.

* 1. **Iteraciones y evolución.**
  2. **Herramientas.**

1. **Resultados**
   1. **Métricas**
   2. **Evaluación cualitativa.**
   3. **Consideraciones de producción.**

# Referencias

United Nations Of ce on Drugs and Crime. (2011). Estimating Illicit Financial Flows Resulting From Drug Traf cking and Other Transnational Organized Crimes. [Online]. Available: https://www.unodc.org/documents/data-and-analysis/Studies/Illicit\_nancial\_ows\_2011\_web.pdf

FATF. (2018). Guidance for a Risk-Based Approach: Securities Sector. [Online]. Available: https://www.fatf-ga.org/media/fatf/documents/recommendations/pdfs/RBA-Securities-Sector.pdf

T. L. V. Barnett, Outliers in Statistical Data, 3rd ed. Hoboken, NJ, USA: Wiley, 1994.

Z. Chen, L. D. V. Khoa, E. N. Teoh, A. Nazir, E. K. Karuppiah, and K. S. Lam, Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: A review, Knowl. Inf. Syst., vol. 57, no. 2, pp. 245285, 2018, doi: 10.1007/s10115-017 1144-z.

M.Carminati, R. Caron, F. Maggi, I. Epifani, and S. Zanero, BankSealer: A decision support system for online banking fraud analysis and investigation, Comput. Secur., vol. 53, pp. 175186, Sep. 2015, doi: 10.1016/j.cose.2015.04.002.

N. A. L. Khac and M.-T. Kechadi, Application of data mining for anti-money laundering detection: A case study, in Proc. 10th IEEE Int. Conf. Data Mining Workshops (ICDMW), W. Fan, W. Hsu, G. I. Webb, B. Liu, C. Zhang, D. Gunopulos, and X. Wu, Eds. Sydney, NSW, Australia: IEEE Computer Society, Dec. 2010, pp. 577584, doi: 10.1109/ICDMW.2010.66.

R. A. L. Torres and M. Ladeira, A proposal for online analysis and identification of fraudulent financial transactions, in Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), M. A. Wani, F. Luo, X. A. Li, D. Dou, and F. Bonchi, Eds., Miami, FL, USA, Dec. 2020, pp. 240245, doi: 10.1109/ICMLA51294.2020.00047.