Enfoque híbrido para la detección de lavado de activos en el sector financiero

Cristian Javier Sánchez Álvarez
Sebastián Naranjo Torres

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

Asesor
Javier Fernando Botia Valderrama

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2024

Cita	(Sánchez Álvarez & Naranjo Torres, 2024)
Referencia **Estilo APA 7 (2020)**	Sánchez Álvarez, C. J., & Naranjo Torres, S. (2024). *Enfoque híbrido para la detección de lavado de activos en el sector financiero* [Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte VII.

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: Danny Alexandro Munera Ramírez

Tabla de contenido

Resumen……...8

Abstract……9

1. Descripción del problema……10
    1.1. Problema de negocio……10
    1.2. Aproximación desde la analítica de datos……11
    1.3. Origen de los datos……12
    1.4. Métricas de desempeño……13
2. Objetivos……..16
    2.1. Objetivo general……16
    2.2. Objetivos específicos……16
3. Datos……17
    3.1. Datos originales……17
    3.2. Base de datos……19
    3.3. Construcción de la base de datos de Entrenamiento y Validación……19
    3.4. Analítica descriptiva……22
4. Proceso de analítica……..26
    4.1. Pipeline principal……..26
    4.2. Preprocesamiento……..26
    4.3. Modelos……27
        4.3.1. Comparativa de algoritmos supervisados……...27
        4.3.2. Comparativa de algoritmos no supervisados……29
    4.4. Métricas……32
5. Metodología……33
    5.1. Baseline……33
    5.2. Validación……36
    5.3. Iteraciones y evolución……37
    5.4. Herramientas……38
6. Resultados……39

6.1. Métricas ……39
6.2. Evaluación cualitativa……...43
6.3. Conclusiones y consideraciones de producción……43
Referencias……45

Lista de tablas

Tabla 1. Estructura de la base de datos original……17
Tabla 2. Composición de la base de datos original……18
Tabla 3. Estructura de la base de datos agregada resultante……..21
Tabla 4. Número total de anomalías en los datos……..….25
Tabla 5. Costo computacional temporal de algoritmos supervisados……28
Tabla 6. Comparativa del rendimiento de algoritmos supervisados en los datos……..29
Tabla 7. Costo computacional temporal de algoritmos no supervisados……..30
Tabla 8. Comparativa del rendimiento de algoritmos no supervisados en los datos…….31
Tabla 9. Muestra de resultados diarios en la reducción de transacciones a revisar……43

Lista de figuras

Figura 1. El diagrama de la metodología CRISP – DM……...11
Figura 2. Histograma de frecuencias del monto normalizado de las transacciones…….22
Figura 3. Distribución acumulada del número de transacciones por usuario……23
Figura 4. Crecimiento del tamaño de la base de datos agregada……..23
Figura 5. Distribución de transacciones por hora del día y día de la semana……...24
Figura 6. Distribución de algunas variables en la base de datos agregada……24
Figura 7. Pipeline principal……...26
Figura 8. Curvas ROC de algoritmos no supervisados…….31
Figura 9. Curvas precision-recall de algoritmos no supervisados……32
Figura 10. Representación gráfica de la evaluación del modelo……...37
Figura 11. Evolución grafica del F1 Score a través de los días…….39
Figura 12. Evolución del TPR y FNR a través de los días……40
Figura 13. Evolución del TNR y FPR a través de los días……41
Figura 14. Grafica t-SNE de la muestra total seleccionada por el modelo……...41
Figura 15. Grafica t-SNE de la distribución de las anomalías seleccionadas por el modelo……42
Figura 16. Muestra de los resultados de las ultimas iteraciones……...42

Siglas, acrónimos y abreviaturas

AML	Anti-Money Laundering
CRISP-DM	Cross-Industry Standard Process for Data Mining
UNODC	United Nations Office on Drugs and Crime
TPR	True Positive Rate
FPR	False Positive Rate
FNR	False Negative Rate
TNR	True Negative Rate
MCC	Matthews Correlation Coefficient
PTDR	Porcentaje de Transacciones Diarias Revisadas.
GAFI	Grupo de Acción Financiera Internacional.
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
LSTM	Long Short-Term Memory
SVM	Support Vector Machine

Resumen

El monitoreo de transacciones financieras es una obligación crucial 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 presentan ciertas desventajas: mientras que los modelos no supervisados pueden detectar nuevos patrones anómalos, suelen generar un alto número de falsas alarmas; los modelos supervisados, por otro lado, ofrecen una mayor tasa de detección, pero requieren una gran cantidad de datos etiquetados para alcanzar dicho rendimiento.

En esta 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. Este enfoque busca aumentar la precisión de la detección y reducir los costos de gestión del cumplimiento. Para ello, utilizamos un conjunto de datos sintético que simula transacciones de clientes que operan en mercados de capitales internacionales.

Los resultados muestran que el modelo híbrido mantiene un excelente rendimiento, con un F1 Score de alrededor del 90%, minimiza los falsos positivos casi a cero y reduce significativamente la carga de trabajo para los analistas del área de cumplimiento.

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

El código del proyecto se encuentra disponible en el repositorio de GitHub asociado (https://github.com/HerrSebas/monografia/tree/main)

Abstract

Monitoring financial transactions is a crucial 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 required to manually review all alerts. However, machine learning-based solutions also have certain drawbacks: while unsupervised models can detect new anomalous patterns, they often generate a high number of false alarms; supervised models, on the other hand, offer a higher detection rate but require a large amount of labeled data to achieve that performance.

In this research, we propose an approach that integrates active learning for anomaly detection, combining unsupervised and supervised learning techniques to improve transaction monitoring processes. This approach aims to increase detection accuracy and reduce compliance management costs. For this purpose, we use a synthetic dataset that simulates transactions of clients operating in international capital markets.

The results show that the hybrid model maintains excellent performance, with an F1 Score of around 90%, minimizes false positives almost to zero, and significantly reduces the workload for compliance analysts.

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

The code of this project is available in the associated GitHub repository (https://github.com/HerrSebas/monografia/tree/main)

1. Descripción del problema

1.1. 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 (United Nations Office on Drugs and Crime [UNODC], 2011). 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.

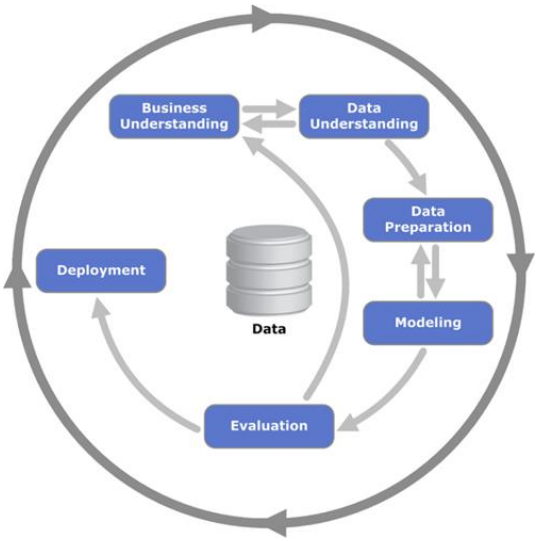
1.2. Aproximación desde la analítica de datos.

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 hibrido (Seung, Sompolinsky, & Tishby, 1992) 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 **puntajes** 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

1.3. 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 que simula perfiles de transacciones de clientes que realizan operaciones en mercados de capital internacional.

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.

1.4. 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 (Google, s.f.).

$$Accuracy = \frac{Verdaderos\ positivos\ (TP) + Verdaderos\ negativos\ (TN)}{Total\ de\ predicciones} \quad (1)$$

- **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 (Google, s.f.).

$$Precision = \frac{Verdaderos\ positivos\ (TP)}{Verdaderos\ positivos\ (TP) + Falsos\ positivos\ (FP)} \quad (2)$$

- **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 (Google, s.f.).

$$Recall = \frac{Verdaderos\ positivos\ (TP)}{Verdaderos\ positivos\ (TP) + Falsos\ negativos\ (FN)} \quad (3)$$

- **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 (Google, s.f.).

$$FPR = \frac{Falsos\ positivos\ (FP)}{Falsos\ positivos\ (FP) + Verdaderos\ negativos\ (TN)} \quad (4)$$

- **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 extremo entre las clases (Google, s.f.).

$$F - Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (5)$$

- **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) (Google, s.f.).

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (6)$$

- **PTDR (Porcentaje de transacciones diarias revisadas):** Mediante esta métrica se pretende evaluar qué porcentaje de las transacciones diarias deben ser etiquetas por los analistas.

$$PTDR = \frac{Transacciones\ revisadas}{Numero\ de\ transacciones\ diarias} \quad (7)$$

2. Objetivos

2.1. Objetivo general.

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

2.2. Objetivos específicos.

- Proponer un sistema híbrido de detección de anomalías que combine modelos de aprendizaje automático supervisado y no supervisado, con el fin de identificar tanto patrones de transacción conocidos como nuevos comportamientos inusuales.

- Mejorar 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 el costo de la revisión manual de transacciones, manteniendo un alto desempeño en términos de precisión y tasa de verdaderos positivos (TPR).

3. Datos

3.1. Datos originales.

La base de datos 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 de la base de datos original.*

Variable	Descripción	Tipo
TransID	Identificador único de la transacción, usado para rastrear y diferenciar cada operación en el sistema	Texto
ClienteOrigen	Código único que identifica al cliente que origina la transacción	Texto (Categoría)
IDAlternativoOrigen	Identificador adicional del cliente origen, utilizado para referencias alternativas o internas	Texto
FechaRegistro	Fecha y hora exacta en la que la transacción fue registrada, útil para el análisis temporal	Fecha y hora (Datetime)
TipoOperacion	Indica si la transacción fue una compra ("Buy") o una venta ("Sell")	Texto (Categoría)
Mercado	Nombre del mercado donde se ejecutó la transacción, como "Market1" o "Market2"	Texto (Categoría)
CodigoISINProducto	Código ISIN del producto negociado, identificador único del instrumento financiero	Texto
TipoProducto	Especifica el tipo de producto, como "FX" (Foreign Exchange), "FutureCommodity"	Texto (Categoría)
ClaseProducto	Clasificación general del producto, como "Trade" o "ADR Conversion"	Texto (Categoría)
MontoNormalizado	Valor normalizado de la transacción, en unidades estándar para facilitar la comparación	Numérico (Decimal)
Moneda	Moneda en la que se realizó la transacción, como "Currency1" o "Currency2"	Texto (Categoría)
IndicadorAnomalia	Etiqueta de anomalía: 0 para transacciones legítimas; 1-5 para categorías de anomalías	Numérico (Entero)

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 de la base de datos original.*

Número de Clientes	Número de Atributos	Número de Transacciones (T)	Transacciones Normales	Transacciones Anomalas (A)	Ratio (A/T)
40	12	29,704,090	29,622,822	81,262	0.27%

Los conjuntos de datos financieros suelen estar extremadamente desequilibrados y, por lo general, contienen entre un 0,1 % y un 1 % de transacciones anómalas (Carminati, Polino, Continella, Lanzi, Maggi, & Zanero, 2018). 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 (Financial Action Task Force [FATF], 2018) basadas en patrones sospechosos sugeridos por el GAFI (Grupo de Acción Financiera Internacional), 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).

3.2. Base de datos.

3.2.1. Construcción de la base de datos de Entrenamiento y Validación.

Para crear la base de datos agregada 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 (Veeramachaneni, Arnaldo, Korrapati, Bassias, & Li, 2016) 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.

3.2.2. 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

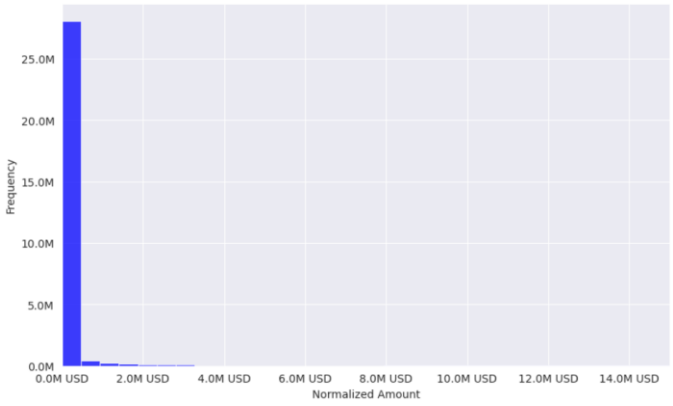
Tabla 3. *Estructura de la base de datos agregada resultante*.

Variable	Tipo	Ejemplo 1	Ejemplo 2
Originator	Texto (Categoría)	32	5
Weekday	Numérico (Entero)	2	5
Hour	Numérico (Entero)	9	14
Morning	Numérico (Binario)	1	0
Evening	Numérico (Binario)	0	1
Night	Numérico (Binario)	0	0
Anomaly	Numérico (Categoría)	0	4
Num_Transactions	Numérico (Entero)	5	10
Total_Amount_Traded	Numérico (Decimal)	1436.05	3545.56
Transactions_Count_Small_Amount	Numérico (Entero)	3	0
Transactions_Count_Round_Amount	Numérico (Entero)	4	15
Transactions_Count_Sell	Numérico (Entero)	3	0
Transactions_Count_Buy	Numérico (Entero)	0	5
Cash_out / Withdrawal / Security_out	Numérico (Decimal)	1549.50	340.00
Simple_Transfer	Numérico (Decimal)	12900.00	0.00
InputOutput_Delta	Numérico (Entero)	4	-10

3.3. Analítica descriptiva.

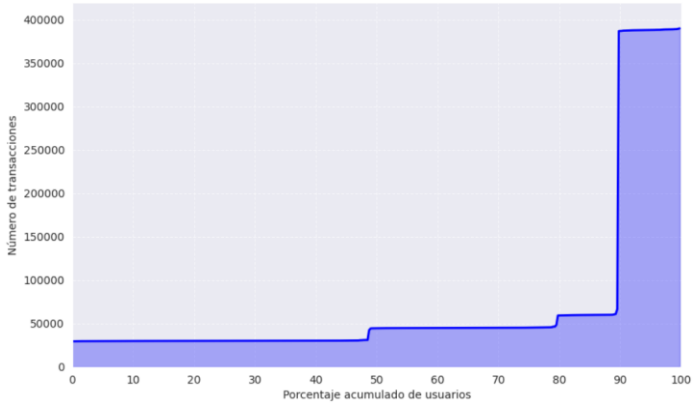
La base de datos inicial no posee valores duplicados ni nulos, y la única variable numérica está normalizada para facilitar comparaciones. Los datos transaccionales contienen información rica y detallada, pero con características complejas que dificultan identificar patrones o distribuciones estadísticas específicas. Sin embargo, es posible extraer algunos datos interesantes.

Figura 2. *Histograma de frecuencias del monto normalizado 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.

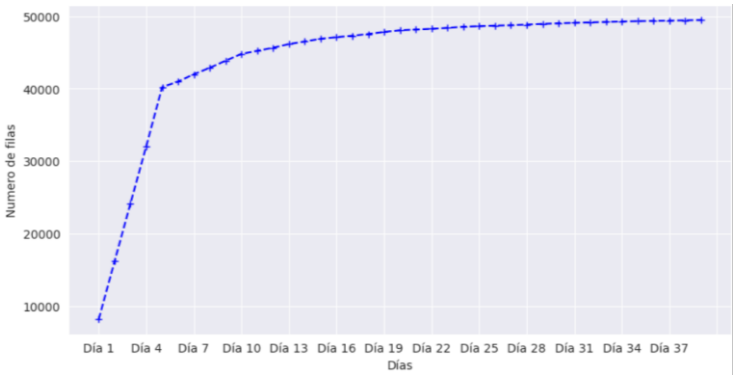
Figura 3. *Distribución acumulada del número de transacciones por usuario.*



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.

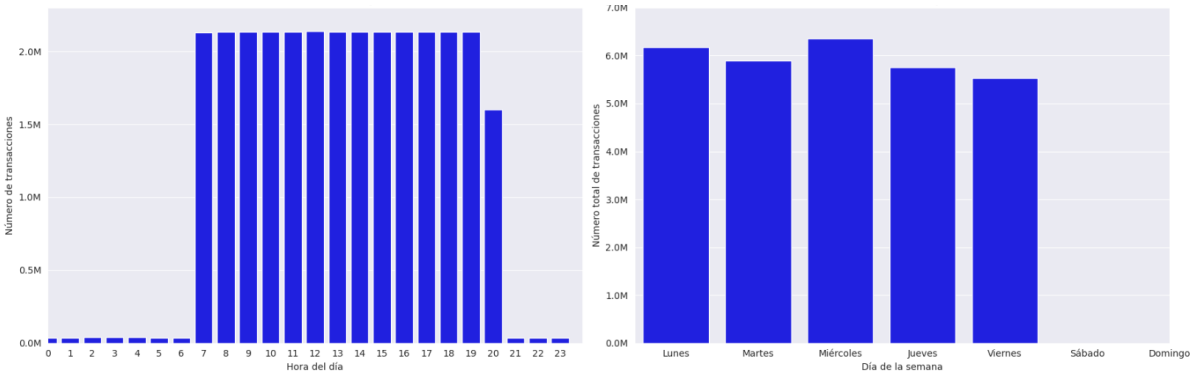
Al agregar la base de datos que se utilizará para entrenar los modelos, es posible extraer características representativas de los datos de manera más efectiva. Además, la agregación de datos no solo mejora la capacidad de los modelos para capturar patrones relevantes, sino que también optimiza el proceso de entrenamiento al reducir la cantidad y complejidad de los datos, permitiendo que el modelo se enfoque en las características más significativas para la predicción. Como se observa en la Figura 4, el crecimiento de la base de datos agregada no es lineal a medida que se incorporan datos adicionales, sino que se estabiliza en aproximadamente 50,000 filas(datos).

Figura 4. *Crecimiento del tamaño de la base de datos agregada.*



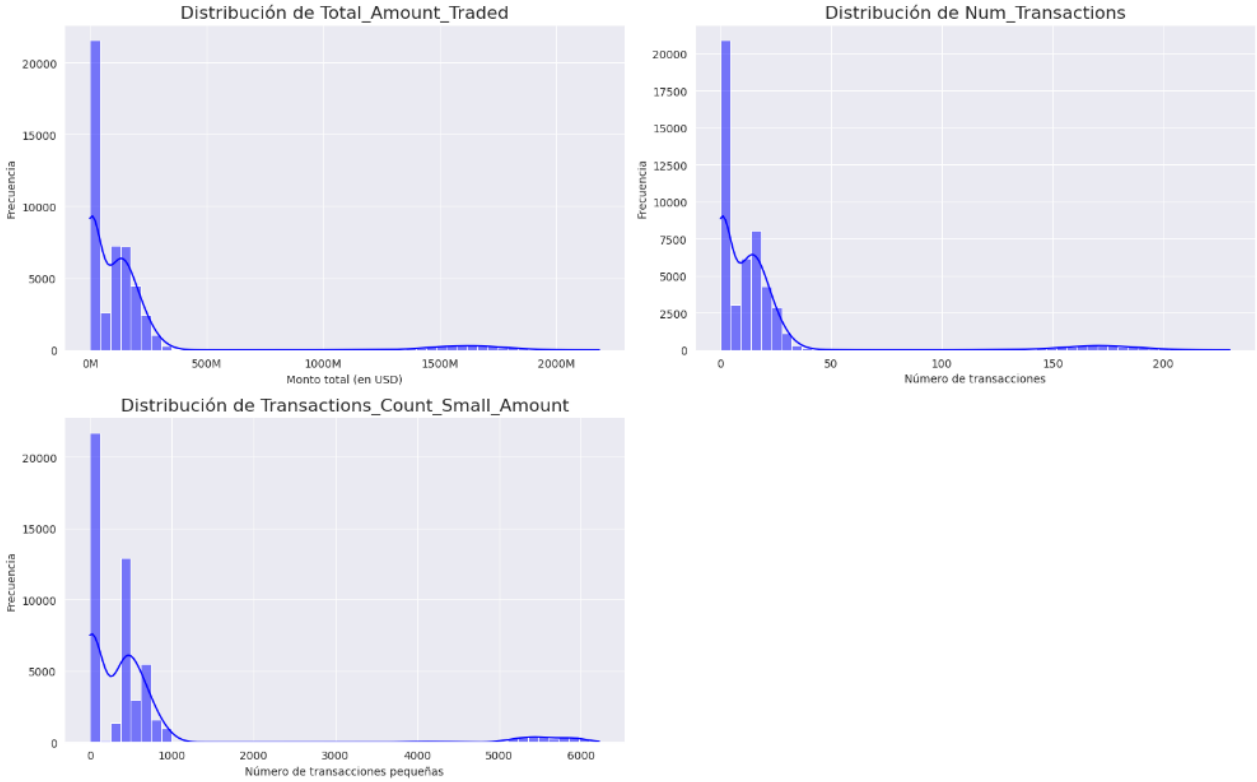
Las transacciones se distribuyen uniformemente tanto en número como en monto total negociado durante el horario de apertura del mercado de valores, mientras que solo un pequeño porcentaje se ejecuta durante las primeras horas de la mañana y al final del día.

Figura 5. *Distribución de transacciones por hora del día y 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 de algunas variables en la base de datos agregada.*



Cuando se agrupan los datos en ventanas de tiempo, los patrones subyacentes tienden a volverse más evidentes. Esta técnica facilita la identificación de patrones repetitivos o comportamientos anómalos que podrían no ser obvios en una visualización global de los datos. Es importante destacar que, en muchos contextos, los valores atípicos (Barnett, 1994), como los que se observan al final de las colas de los histogramas, suelen considerarse errores y, por lo tanto, se eliminan durante el preprocesamiento de los datos.

Sin embargo, en problemas de detección de anomalías, los valores atípicos no son un problema por resolver, sino el objetivo del análisis. Estos valores representan desviaciones significativas de los patrones normales de comportamiento y, en casos como el lavado de dinero o fraude, suelen ser precisamente donde se esconden las actividades sospechosas. Eliminar estos valores atípicos durante el preprocesamiento, como se haría en un análisis tradicional para "limpiar" los datos, iría en contra del propósito central de la detección de anomalías.

En lugar de eliminar los valores atípicos, en la detección de anomalías se busca identificarlos y analizarlos para comprender si corresponden a comportamientos legítimos o si, por el contrario, reflejan actividades fraudulentas o ilícitas. De hecho, los valores atípicos pueden ser señales cruciales que indican que algo fuera de lo común está ocurriendo, y estos son precisamente los casos que los modelos de detección de anomalías buscan identificar.

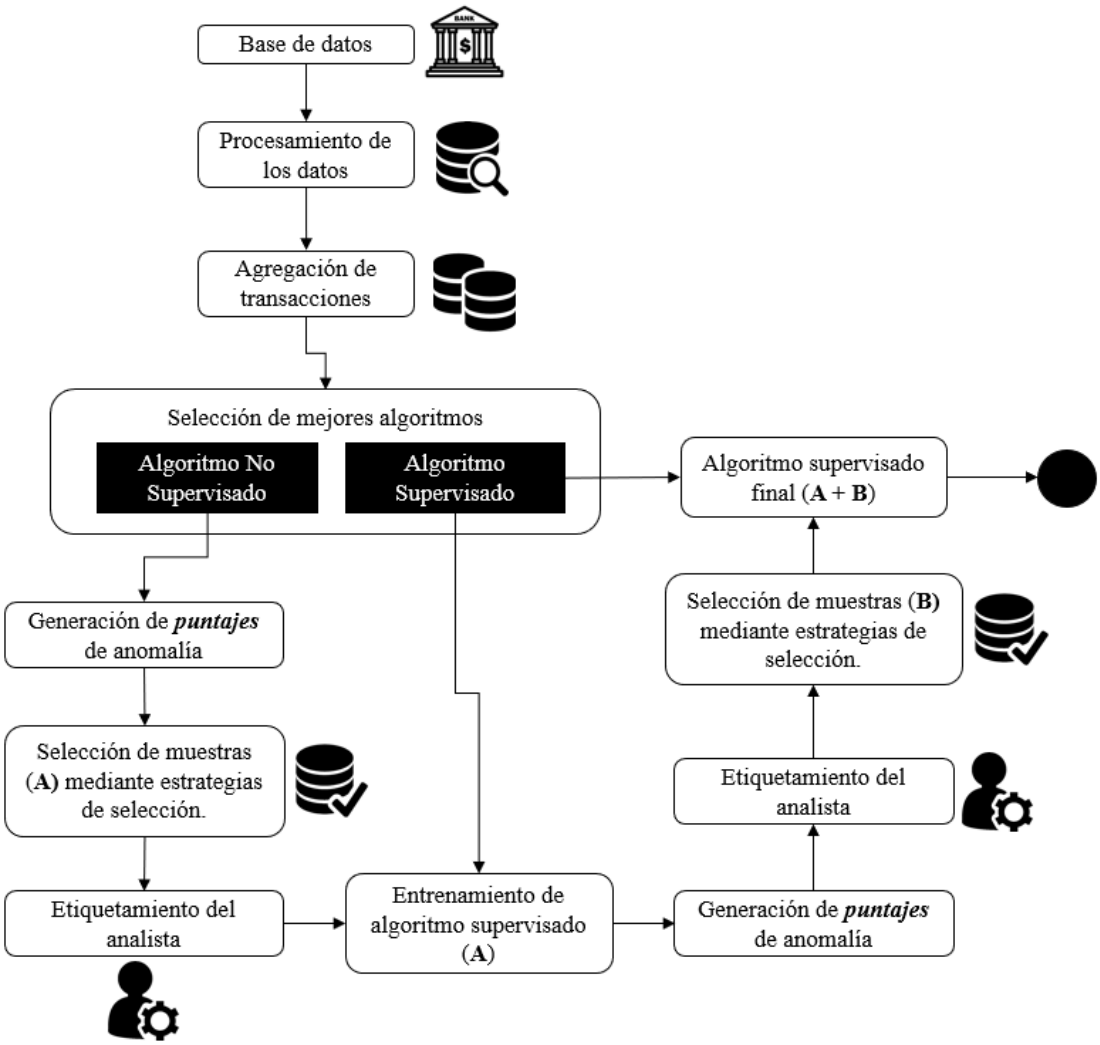
Tabla 4. *Número total de anomalías en los datos.*

Anomalia 1	Anomalia 2	Anomalia 3	Anomalia 4	Anomalia 5
631	511	282	254	252

4. Proceso de analítica

4.1. Pipeline principal.

Figura 7. *Pipeline principal.*



4.2. 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 nuestra base de datos agregada. 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.

4.3. Modelos.

4.3.1. 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 el costo computacional temporal de varios algoritmos. Esto nos proporciona una visión

general de las opciones disponibles y nos ayuda a identificar cuáles podrían ser óptimas en términos de eficiencia computacional, tiempo de entrenamiento y capacidad de predicción.

Tabla 5. *Costo computacional temporal de algoritmos supervisados.*

Algoritmo	Costo	Descripción
Naive Bayes	$O(n \cdot m)$	*n:* Número de muestras, *m:* Número de características.
Decision Tree	$O(n \cdot log(n))$	*n:* Número de muestras.
Random Forest	$O(n \cdot log(n) \cdot m)$	*n:* Número de muestras, *m:* Número de características.
CatBoost	$O(n \cdot k)$	*n:* Número de muestras, *k:* Número de iteraciones.
SVM	$O(n^2 \cdot d)$	*n:* Número de muestras, *d:* Número de características.
LightGBM	$O(n \cdot log(n))$	*n:* Número de muestras.

En la tabla que resume el costo computacional (The Kernel Trip, s.f.), podemos observar que Naive Bayes es generalmente el algoritmo más rápido, con una complejidad de $O(n \cdot m)$ lo que lo hace adecuado para conjuntos de datos grandes con múltiples características. Por otro lado, Support Vector Machine (SVM) presenta la mayor complejidad, $O(n^2 \cdot d)$, 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 $O(n \cdot log(n) \cdot m)$ y $O(n \cdot log(n))$ 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 realizar la prueba, entrenamos y ajustamos todos los modelos utilizando el conjunto de datos de entrenamiento, que corresponde al 70% del total (20.792.863 transacciones), y evaluamos su rendimiento en el 30% restante (8.911.227 transacciones).

Tabla 6. *Comparativa del rendimiento de algoritmos supervisados en los datos.*

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 (Quinlan, 1986) se destaca como el mejor clasificador en términos generales según los resultados obtenidos. Presenta la mayor precisión (0.998), superando a los demás modelos en cuanto a exactitud en las predicciones. Además, muestra un excelente equilibrio entre las métricas de rendimiento, como el puntaje F1 (0.9702), uno de los más altos, 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 puntaje F1 (0.595), a pesar de ser el más rápido.

4.3.2. Comparativa de algoritmos no supervisados.

Utilizamos igualmente la tabla de complejidad temporal de algoritmos no supervisados (The Kernel Trip, s.f.) para ayudar a guiar en el proceso de selección del mejor algoritmo no supervisado.

Tabla 7. *Costo computacional temporal de algoritmos no supervisados.*

Algoritmo	Costo	Descripción
Isolation Forest	$O(n \cdot log(n))$	*n:* número de muestras, *log n:* profundidad de los árboles.
Autoencoder	$O(n \cdot m \cdot e)$	*n:* número de muestras, *m:* dimensiones de entrada, *e:* número de épocas de entrenamiento.
One-Class SVM	$O(n^3)$	*n:* número de muestras, lo que implica alta complejidad en grandes conjuntos de datos.
K-Nearest Neighbors	$O(n \cdot d)$	*n:* número de muestras, *d:* dimensiones de los datos; se calcula la distancia a todos los puntos.
Local Outlier Factor	$O(n \cdot d \cdot log(n))$	*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 $O(n \cdot m \cdot e)$, 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.

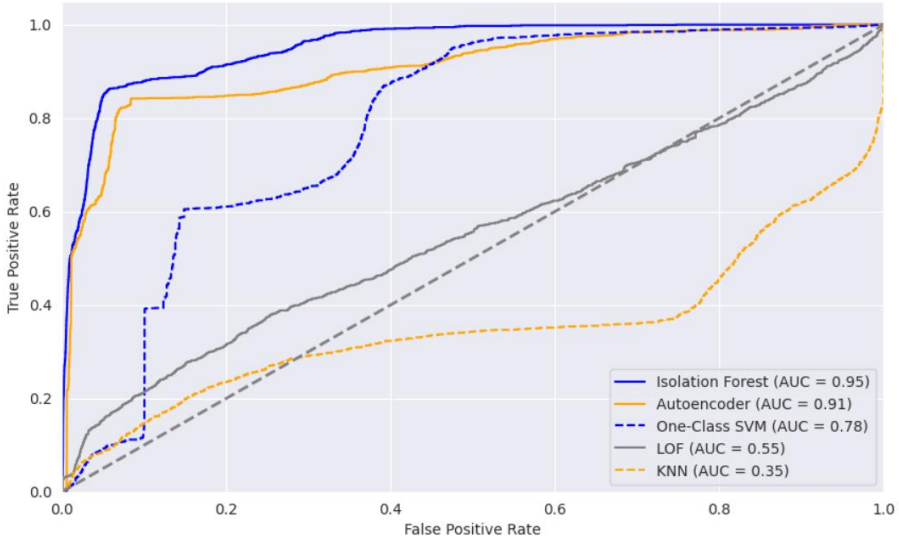
Como se observa en los resultados de la tabla 8, el Isolation Forest (Bosque de Aislamiento) es el algoritmo con el mejor desempeño general (Shokry, Rizka, & Labib, 2020). Presenta una alta exactitud de 0.9661, la mejor precision (0.7243) y puntaje F1 (0.7488) más alto, lo que indica su capacidad para 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.

Tabla 8. *Comparativa del rendimiento de algoritmos no supervisados en los datos.*

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

Aunque el Autoencoder también ofrece resultados sobresalientes en cuanto a False Negative Rate (0.393) y True Positive Rate (0.6070), su tiempo de entrenamiento es significativamente mayor, lo que lo hace menos eficiente en comparación con Isolation Forest.

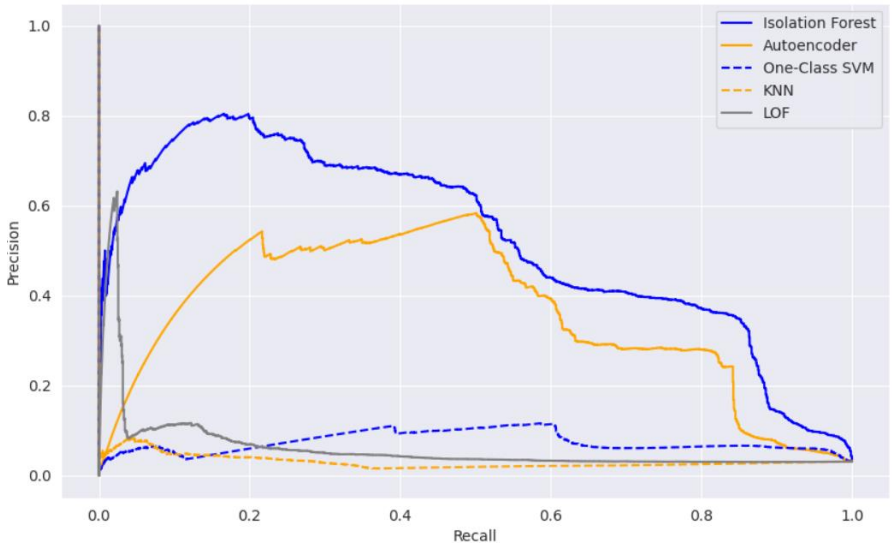
Figura 8. *Curvas ROC de algoritmos no supervisados.*



La Curva ROC, que muestra la capacidad de cada algoritmo para separar las clases de anomalías y transacciones normales. En ella, el Isolation Forest destaca como el mejor algoritmo con un AUC de 0.95, lo que indica una excelente capacidad para identificar anomalías con un bajo nivel de falsos positivos. El Autoencoder, con un AUC de 0.91, también muestra un rendimiento sólido,

aunque inferior al Isolation Forest. El One-Class SVM tiene un rendimiento moderado (AUC de 0.78), mientras que LOF y KNN, con AUC de 0.55 y 0.35 respectivamente, muestran un bajo poder de discriminación, similar al de un clasificador aleatorio.

Figura 9. *Curvas precision-recall de algoritmos no supervisados.*



Por otro lado, la Curva de Precisión-Recall, es más útil en el contexto de detección de anomalías donde las clases están desbalanceadas. En esta curva, el Isolation Forest mantiene nuevamente el mejor equilibrio, con una alta precisión y recall en la identificación de anomalías, lo que lo convierte en la opción más confiable. El Autoencoder muestra un buen rendimiento, pero su precisión disminuye conforme intenta identificar más anomalías (aumenta el recall), indicando que aumenta la tasa de falsos positivos en esos casos. Los algoritmos restantes, One-Class SVM, KNN y LOF, presentan curvas mucho más bajas, con bajos valores de precisión y recall, especialmente KNN y LOF, que tienen un rendimiento muy limitado en la detección de anomalías.

4.4. Métricas.

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

5. Metodología

5.1. 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 (Das, Islam, Jayakodi, & Doppa, 2019).

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

- **Primera estrategia:**

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 puntaje de anomalía entregado por el algoritmo Isolation Forest, seleccionando las K muestras con los puntajes de anomalía más altos (top) y las K con los más bajos (bottom) a partir del conjunto de datos U'.

a) Sea $S(x_i)$ el puntaje de anomalía de cada vector de alto nivel x_i en el conjunto de datos U_t.
b) Ordenamos U_t en orden descendente según $S(x_i)$:

$$U' = \{x_1, x_2, \dots, x_n\} : S(x_1) \geq S(x_2) \geq \cdots \geq S(x_n) \quad (7)$$

c) Seleccionar las K muestras más anómalas:

$$C_{top} = \{x_1, x_2, \dots, x_k\} \quad (8)$$

d) Seleccionar las K muestras menos anómalas:

$$C_{bottom} = \{x_{n-k+1}, x_{n-k+2}, \dots, x_n\} \quad (9)$$

e) Conjunto final de muestras para revisión es:

$$C = C_{top} \cup C_{bottom} \quad (10)$$

- **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. Llevamos a cabo agrupamiento mediante el algoritmo de HDBSCAN para reunir 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.

a) Filtrar las muestras con puntajes de anomalía altos. Seleccionamos el subconjunto $U_p \subset U_t$ que contiene solo las muestras x_i con un puntaje de anomalía $S(x_i)$ superior a un percentil definido p:

$$U_p = \{x_i \in U' : S(x_i) > percentil\,(S, p)\} \quad (11)$$

b) Aplicar clustering. Aplicamos un algoritmo de clustering (específicamente HDBSCAN) a U_p para dividirlo en n clústeres de acuerdo con su similitud:

$$U_p = \bigcup_{j=1}^{n} Cl_j \tag{12}$$

c) Ordenar clústeres por densidad. Denotamos la densidad de cada clúster j como D_{Clj}. Ordenamos los clústeres de menor a mayor densidad, ya que los clústeres menos densos son más propensos a contener anomalías:

$$clusters\ ordenados = \{cl_1, cl_2, \dots, cl_n\} : D(cl_1) \leq D(cl_2) \leq \cdots \leq D(cl_n) \tag{13}$$

d) Seleccionar muestras de cada clúster. A partir de los clústeres menos densos, seleccionamos una cantidad fija r de muestras de cada clúster hasta alcanzar el número total de muestras deseado K:

$$C_{diverse} = \bigcup_{j=1}^{m} \{x_{j,1}, x_{j,2}, \dots, x_{j,r}\} \tag{14}$$

Donde $x_{j,k} \in cl_j$ y m es el número de clústeres necesarios para cubrir el total K.

- **Tercera estrategia:**

La tercera estrategia está orientada a reducir la incertidumbre en el modelo Random Forest, que se entrena inicialmente sobre el conjunto combinado U_c (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 *K* 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.

a) Definimos U_c como el conjunto combinado de muestras seleccionadas desde las muestras anteriores:

$$U_c = C_{top} \cup C_{bottom} \cup C_{diverse} \tag{15}$$

b) Entrenamos un modelo de Random Forest sobre U_c para obtener el puntaje de probabilidad $rf\_score(x_i)$ para cada muestra del conjunto de datos inicial $x_i \in U_t$

c) Definimos el puntaje de entropía para cada muestra x_i en U_t como la distancia a 0.5:

$$entropy\_score(x_i) = \left| rf\_score_{(x_i)} - 0{,}5 \right| \quad (16)$$

d) Ordenamos y seleccionamos las K muestras con mayor incertidumbre:

$$C_{entropy} = \{ x_1, x_2, \dots, x_k \} : entropy\_score(x_1) \leq \cdots \leq entropy\_score(x_k) \quad (17)$$

- **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.

a) Calculamos el puntaje de conflicto para cada muestra $x_i \in U_t$ basado en la discrepancia entre Isolation Forest y el Random Forest inicial entrenado sobre U_c:

$$conflict\_score(x_i) = | S(x_i) - rf\_score(x_i) | \quad (18)$$

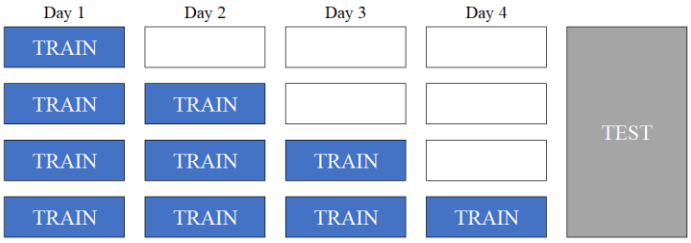
b) Ordenamos y seleccionamos las K muestras con mayor conflicto entre los modelos:

$$C_{conflict} = \{ x_1, x_2, \dots, x_k \} : conflict\_score(x_1) \geq \cdots \geq conflict\_score(x_k) \quad (19)$$

5.2. 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 11. 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 entrenamiento 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 experto, se procederá al entrenamiento en el módulo supervisado con las muestras etiquetadas.

Figura 10. *Representación gráfica de la evaluación del modelo.*



Una vez que los analistas revisan las muestras y están 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. En este contexto, se utilizarán los datos correspondientes a los dos primeros meses (60 días aproximadamente) para entrenar el modelo, mientras que el tercer mes se destinará a evaluar las predicciones y medir las métricas de desempeño.

5.3. Iteraciones y evolución.

El proceso iterativo buscó optimizar el rendimiento del modelo, controlando el número de transacciones revisadas por los analistas. Cada iteración evaluó no solo métricas de rendimiento descritas anteriormente, como el F1-Score, TPR, TNR y FPR, sino también la eficiencia que facilitara en la revisión de transacciones sospechosas, con el objetivo de reducir la carga de trabajo manual para los expertos sin perder precisión en la detección de anomalías.

Cada iteración nos permitió ajustar el tamaño y la composición de este conjunto de muestras revisadas, buscando mantener muy bajo el número de transacciones a revisar sin comprometer la sensibilidad ni la precisión del modelo. Inicialmente, el enfoque incluyó un mayor número de muestras para asegurar una cobertura amplia de anomalías, pero a medida que el modelo se entrenaba con más datos y ajustábamos las estrategias de selección, se redujo progresivamente la cantidad de datos que debían ser revisados por los analistas día tras día.

En cada iteración, el F1-Score fue nuestra métrica principal para evaluar el equilibrio general del modelo. Un aumento en el F1-Score indicaba que estábamos mejorando tanto en la detección de

anomalías como en la reducción de falsos positivos, sin sacrificar la precisión. Esperábamos observar una tendencia creciente en el F1-Score y otras métricas a medida que transcurrían los días o las iteraciones.

5.4. Herramientas.

Elegimos Python como el lenguaje principal de programación debido a su versatilidad y amplia adopción en el ámbito del machine learning y la ciencia de datos. Python cuenta con librerías robustas como *pandas* para la manipulación de datos, *sklearn* para la implementación de algoritmos de machine learning, y *HDBSCAN* para la detección de patrones en datos no supervisados.

Los notebooks de Kaggle se usaron para desarrollar y ejecutar el código en un entorno compartido y de fácil acceso. Kaggle proporciona un entorno de notebooks en la nube con GPU y TPU disponibles, lo que facilitó tanto la ejecución eficiente de los modelos como la colaboración en tiempo real.

Utilizamos GitHub para alojar el repositorio del proyecto, permitiendo un control de versiones eficiente y una colaboración fluida entre los miembros del equipo. La plataforma de GitHub facilitó la organización y gestión de cambios en el código, y permitió documentar el progreso del proyecto.

Google Drive se empleó para compartir y almacenar documentos del proyecto, incluyendo reportes intermedios, presentaciones y documentación técnica. La facilidad para gestionar permisos de acceso y colaborar en tiempo real hizo de Google Drive una herramienta ideal para la revisión y edición conjunta de documentos.

6. Resultados

6.1. Métricas.

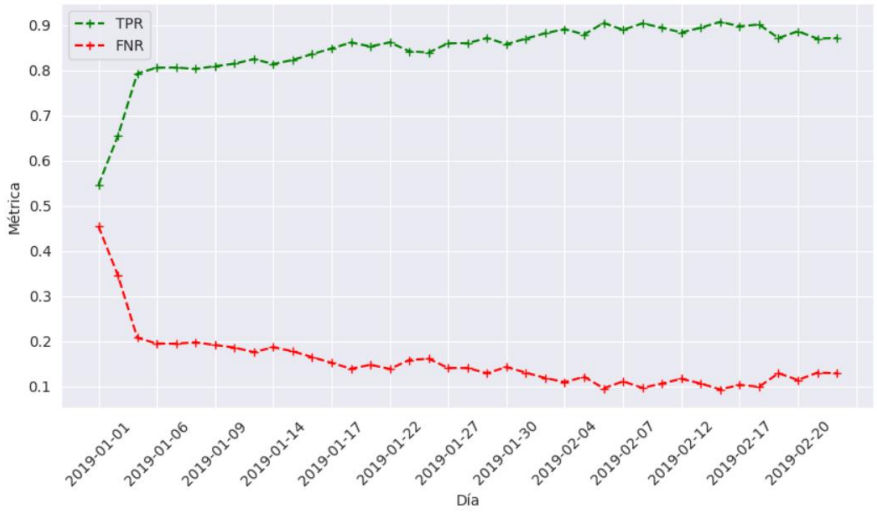
El gráfico muestra cómo evoluciona el F1 Score a medida que el sistema se entrena y se ajusta diariamente. Observamos un aumento significativo en la métrica desde un valor inicial bajo hasta estabilizarse en un valor alto cercano a 0.90. Este aumento rápido sugiere que el modelo está mejorando considerablemente en sus primeras etapas de entrenamiento, logrando un equilibrio entre la precisión y el recall. A partir del 7 de febrero, la curva se mantiene en un nivel constante, indicando una estabilización del modelo en cuanto a la predicción correcta de las clases positivas. Estos resultados reflejan que el sistema logra adaptarse y mejorar sus predicciones a lo largo del tiempo, optimizando el aprendizaje activo y reduciendo tanto los falsos positivos como los falsos negativos.

Figura 11. *Evolución grafica del F1 Score a través de los días.*

El segundo gráfico ilustra la evolución del True Positive Rate (TPR) y el False Negative Rate (FNR) a lo largo del tiempo. La TPR mide la capacidad del modelo para detectar transacciones anómalas, mientras que la FNR indica los casos en que el modelo no detecta dichas anomalías. Un TPR creciente y un FNR decreciente reflejan que el modelo mejora su capacidad de identificar transacciones sospechosas, logrando así una mayor efectividad en la detección de anomalías.

Inicialmente, el TPR comienza en un nivel medio (cercano a 0.5) y aumenta rápidamente hasta estabilizarse alrededor de 0.85. Este comportamiento es consistente con la mejora del F1 Score, ya que una mayor tasa de verdaderos positivos contribuye a una mejor capacidad de predicción. Por otro lado, el FNR muestra una disminución constante desde un valor inicial más alto (aproximadamente 0.4) hasta alcanzar un valor bajo cercano a 0.1, lo que indica una reducción en los falsos negativos. Este balance entre TPR y FNR sugiere que el modelo está logrando identificar correctamente las instancias positivas con mayor frecuencia, mientras reduce los errores de omisión (falsos negativos).

Figura 12. *Evolución del TPR y FNR a través de los días.*

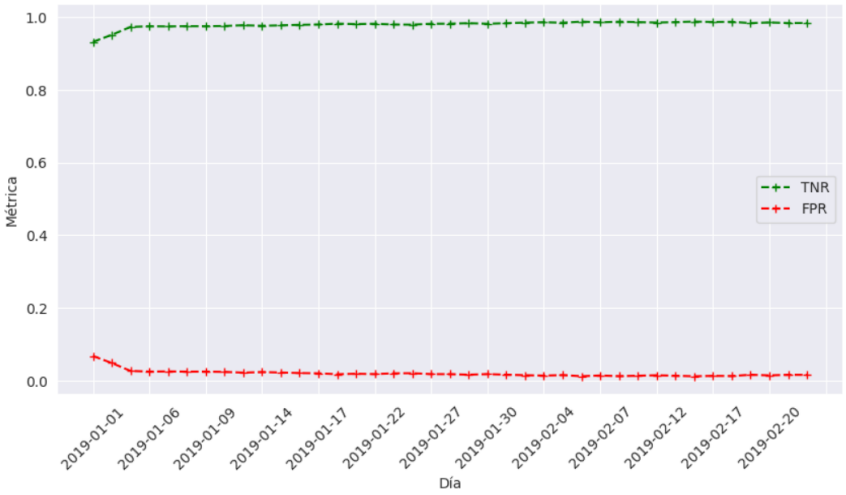


Este gráfico muestra cómo varían la tasa de verdaderos negativos (TNR) y la tasa de falsos positivos (FPR). La TNR representa la precisión del modelo al clasificar correctamente las transacciones no anómalas, mientras que la FPR indica la cantidad de transacciones normales clasificadas incorrectamente como anómalas.

La métrica TNR se mantiene consistentemente alta a lo largo de los días, con valores cercanos a 1. Esto indica que el modelo está logrando identificar correctamente como negativos la mayoría de los casos que realmente no son anómalos. Este comportamiento es importante, ya que implica una baja tasa de falsos positivos o, en otras palabras, pocas alarmas innecesarias para los analistas. Por otro lado, la métrica FPR también es excelente, lo que indica que el modelo casi no clasifica erróneamente los casos negativos como positivos; es decir, rara vez comete errores al marcar

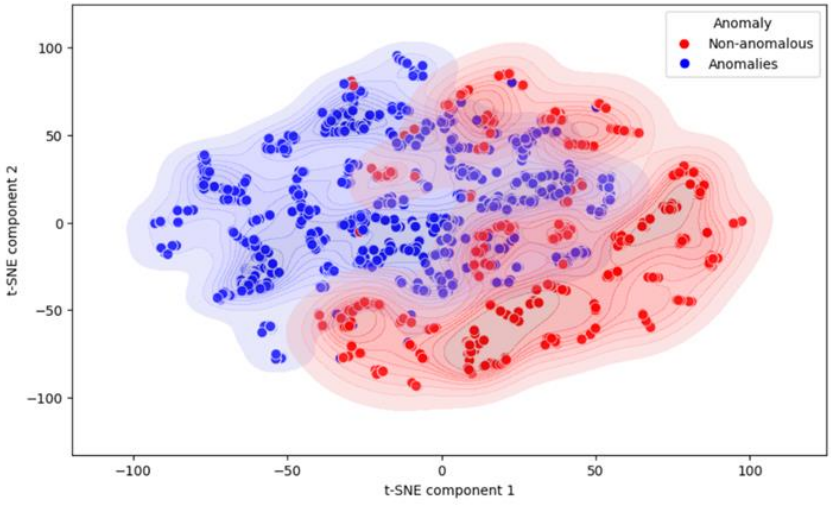
transacciones normales como anómalas. Esto resulta beneficioso en contextos donde los falsos positivos son costosos o indeseables, ya que reduce la cantidad de revisiones innecesarias.

Figura 13. *Evolución del TNR y FPR a través de los días.*



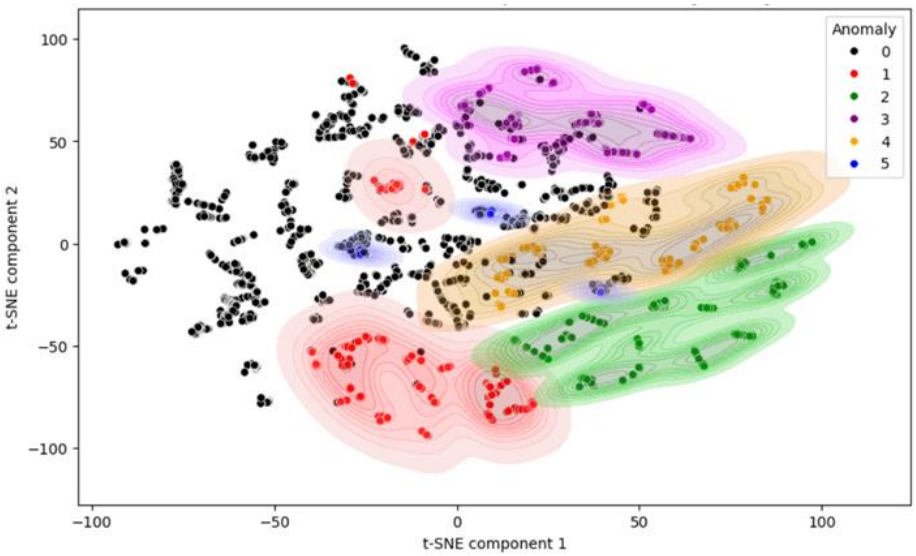
Al visualizar la distribución de las muestras utilizando el algoritmo t-SNE con 2 componentes y una perplejidad de 30, observamos que los puntos correspondientes a datos anómalos y no anómalos se encuentran distribuidos de forma equilibrada en el espacio bidimensional. La separación clara entre los puntos anómalos (en rojo) y los no anómalos (en azul) indica que el modelo ha logrado identificar con precisión las anomalías dentro de la muestra seleccionada. Esta visualización confirma que la muestra de datos final utilizada por el modelo ya no presenta un desbalance significativo entre datos anómalos y no anómalos, lo cual es esencial para mejorar la robustez del análisis y reducir el riesgo de sesgos en los resultados.

Figura 14. *Grafica t-SNE de la muestra total seleccionada por el modelo.*



Finalmente, al graficar las muestras con las anomalías categorizadas (0, 1, 2, 3, 4 y 5), se aprecia que la mayoría de las categorías presentan una distribución bien definida y compacta en el espacio bidimensional. Esto sugiere que el modelo no solo identifica las anomalías de manera efectiva, sino que también es capaz de agruparlas según características distintivas. Sin embargo, se observa una excepción en la categoría de anomalías número 5, la cual muestra una dispersión notable y un menor número de muestras en comparación con las otras categorías. Esta dispersión podría indicar una mayor variabilidad dentro de esta categoría, lo que podría estar relacionado con una menor consistencia o un comportamiento más atípico en los datos asociados a esta anomalía.

Figura 15. *Grafica t-SNE de la distribución de las anomalías seleccionadas por el modelo.*



Este resultado es coherente con los resultados de las iteraciones previas, donde se identificó que esta categoría tiene un rendimiento inferior en comparación con las otras. La dificultad para agrupar esta anomalía de manera clara podría deberse a su naturaleza intrínsecamente variada o a la presencia de subgrupos no identificados en los datos.

Figura 16. *Muestra de los resultados de las ultimas iteraciones.*

```
Accuracy  0.9981393987540244
F1-score  0.9312014193606765
[[46798     3     0     8     0     0]
 [    5   321     0     0     0     0]
 [    0     0   295     0     0     0]
 [   10     0     0   156     0     0]
 [    1     0     0     0   115     0]
 [   62     0     0     0     0    60]]
```

6.2. Evaluación cualitativa.

El objetivo principal de este modelo es incrementar el rendimiento del negocio mediante la reducción de las muestras que deben ser analizadas por los analistas responsables de detectar transacciones sospechosas, sin sacrificar la eficacia en la detección de anomalías. Las métricas, como el F1 Score, mostraron resultados cercanos al 90%, lo cual indica una alta efectividad en la identificación de anomalías. Además, el modelo es excelente en la reducción de falsos positivos, lo que mejora la eficiencia en el etiquetado por parte de los analistas.

En cuanto al número de muestras revisadas por los analistas, diariamente representan menos del 1% del total de transacciones diarias. Esto, en términos de negocio, se traduce en una mejora significativa en la eficiencia operativa y en el uso de recursos.

Tabla 9. *Muestra de resultados diarios en la reducción de transacciones a revisar.*

Día	Transacciones Totales	Transacciones Revisadas	PTDR
1/01/2019	580,361	172	0.03%
2/01/2019	512,880	169	0.03%
3/01/2019	485,300	183	0.04%
6/01/2019	452,398	184	0.04%
7/01/2019	555,044	183	0.03%
8/01/2019	475,205	170	0.04%
9/01/2019	612,723	199	0.03%
10/01/2019	412,370	191	0.05%
13/01/2019	515,671	196	0.04%
14/01/2019	587,872	181	0.03%

6.3. Conclusiones y consideraciones de producción.

- El enfoque híbrido propuesto, basado en la integración de aprendizaje supervisado, no supervisado y aprendizaje activo, demuestra ser una solución efectiva para la detección de lavado de activos en el sector financiero. Este modelo logra un equilibrio entre la precisión

y la reducción de falsos positivos, alcanzando un puntaje F1 cercano al 90%. Además, reduce significativamente la carga de trabajo de los analistas, limitando las transacciones a revisar a menos del 1% del total diario. Además, el sistema puede ser capaz de adaptarse a nuevas anomalías y patrones de comportamiento, gracias a su componente iterativo y a la incorporación de datos revisados por expertos. Esto no solo optimiza la eficiencia operativa, sino que también mejora la robustez del modelo para detectar actividades sospechosas en entornos reales.

- En este trabajo, asumimos que los analistas que revisan las transacciones siempre etiquetan correctamente cada transacción a evaluar. No obstante, sería valioso estudiar el impacto de los errores de etiquetado y adaptar el modelo para manejar estos posibles errores, mejorando así su robustez y confiabilidad en entornos de producción.

- Además, sería interesante aplicar este modelo en diferentes sectores, ya que, dependiendo del sector, podrían ajustarse los umbrales del "anomaly score". Por ejemplo, en un sector de alto riesgo, como el financiero, se podría reducir el umbral a partir del cual una muestra se envía al analista para revisión, lo que aumentaría la cantidad de transacciones revisadas y ofrecería un enfoque más conservador.

- Finalmente, en este trabajo no se consideró el uso de algoritmos de *deep learning* debido a su alto costo computacional en comparación con los algoritmos de *machine learning* tradicionales. Investigaciones futuras podrían explorar la implementación de algoritmos como LSTM, los cuales permiten capturar correlaciones a largo plazo, aportando una perspectiva más profunda y efectiva en la detección de patrones complejos en los datos.

Referencias

United Nations Office on Drugs and Crime. (2011). Estimating illicit financial flows resulting from drug trafficking and other transnational organized crimes. United Nations. https://www.unodc.org/documents/data-and-analysis/Studies/Illicit_financial_flows_2011_web.pdf

The Kernel Trip. (s.f.). Computational complexity of learning algorithms. The Kernel Trip. https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/

FATF. (2018). Guidance for a risk-based approach: Securities sector. Financial Action Task Force. https://www.fatf-ga.org/media/fatf/documents/recommendations/pdfs/RBA-Securities-Sector.pdf

Barnett, T. L. V. (1994). Outliers in statistical data (3rd ed.). Wiley.

Carminati, M., Polino, M., Continella, A., Lanzi, A., Maggi, F., & Zanero, S. (2018). Security evaluation of a banking fraud analysis system. ACM Transactions on Privacy and Security, 21(3), 11:1–11:31. https://doi.org/10.1145/3178370

Seung, H. S., Sompolinsky, H., & Tishby, N. (1992). Statistical mechanics of learning from examples. Physical Review A: General Physics, 45, 6056–6091. https://doi.org/10.1103/PhysRevA.45.6056

Veeramachaneni, K., Arnaldo, I., Korrapati, V., Bassias, C., & Li, K. (2016). Ai2: Training a big data machine to defend. En Proceedings of the 2nd IEEE International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance Smart Computing (HPSC), IEEE International Conference on Intelligent Data Security (IDS) (pp. 49–54). IEEE. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.79

Shokry, A. E. M., Rizka, M. A., & Labib, N. M. (2020). Counterterrorism finance by detecting money laundering hidden networks using unsupervised machine learning algorithms. En Proceedings of the International Conference on ICT, Society, and Human Beings (pp. 89–97).

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106. https://doi.org/10.1023/A:1022643204877

Das, S., Islam, M. R., Jayakodi, N. K., & Doppa, J. R. (2019). Active anomaly detection via ensembles: Insights, algorithms, and interpretability. arXiv. https://arxiv.org/abs/1901.08930

Google. (s.f.). Accuracy, precision and recall. Google Developers. https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall?hl=es-419