# *Minería de Anomalías*

GUSTAVO SOBRADO ALLER

UO286277

71777616K

Universidad de Oviedo

# TABLA DE CONTENIDO

# 1. INTRODUCCIÓN

## 1.1 CONCEPTO DE MINERÍA DE ANOMALÍAS

La minería de anomalías es una disciplina dentro de la minería de datos, dedicada a la identificación de patrones o comportamientos que se desvían bastante de los datos esperados o considerados normales. Estas desviaciones, denominadas anomalías o outliers, pueden ser interpretadas de varias maneras según el contexto, abarcando desde errores en el registro hasta eventos significativos como fraudes o fallos en sistemas.

De manera más formal, se puede definir una anomalía como un dato o conjunto de datos cuya aparición es poco común o altamente improbable en relación con el comportamiento general del conjunto de datos.

## 1.2 RELACIÓN CON LA INTELIGENCIA DE NEGOCIO

En el ámbito de la Inteligencia de Negocio (BI), la detección de anomalías tiene un papel muy importante, ya que permite a las organizaciones:

- Identificar irregularidades en sus operaciones.
- Detectar fraudes en tiempo real.
- Optimizar procesos mediante la identificación de cuellos de botella.
- Anticipar eventos disruptivos, como picos inesperados de demanda o fallos en la producción.

Al integrarse con sistemas de BI, las técnicas de detección de anomalías permiten a las empresas tomar decisiones basándose en información clave que podría pasar desapercibida con otros métodos de análisis tradicionales.

## 1.3 OBJETIVO Y ENFOQUE DEL TRABAJO

El objetivo de este trabajo será explorar las técnicas de minería de anomalías. Se analizarán los fundamentos teóricos, las herramientas utilizadas y su implementación para resolver un caso de estudio real.

El enfoque del trabajo será teórico y práctico. Se buscará describir los conceptos fundamentales y demostrar cómo las técnicas modernas de detección de anomalías pueden aplicarse en un entorno empresarial, utilizando Python y bibliotecas para el análisis y la visualización de datos.

Durante el trabajo, se desarrollarán los fundamentos teóricos, el caso de estudio seleccionado, y los métodos aplicados para identificar anomalías, así como las conclusiones del análisis.

## 1.4 RELEVANCIA EMPRESARIAL Y CASOS REALES

La minería de anomalías es un componente esencial en la Inteligencia de Negocio, ya que permite identificar eventos raros o poco comunes que pueden tener un gran impacto en la operativa y el desempeño empresarial. Al detectar patrones inesperados, las organizaciones pueden mitigar riesgos, optimizar procesos y anticiparse a posibles problemas futuros.

- En el sector financiero, empresas como PayPal usan modelos avanzados, como Isolation Forest, para analizar millones de transacciones diarias y detectar fraudes en tiempo real. Esto no solo reduce pérdidas económicas, sino que también protege al cliente de una posible estafa.
- En manufactura, Tesla también aplica minería de anomalías en datos de sensores para prevenir fallos en sus vehículos, permitiendo un mantenimiento predictivo que mejora la seguridad y reduce costos.
- En el comercio electrónico, Netflix analiza patrones de uso para detectar accesos sospechosos a cuentas, asegurando la privacidad y seguridad de sus usuarios.
- Además, en sectores como logística y ciberseguridad, se utiliza para identificar cuellos de botella o detectar intentos de intrusión.

Estas aplicaciones demuestran que la identificación de anomalías no solo salvaguarda a las empresas, sino que también fomenta la toma de decisiones fundamentadas en datos, lo que permite a las organizaciones ser más resilientes y competitivas en entornos de mercado en constante cambio.

# 2. FUNDAMENTOS TEÓRICOS Y ESTADO DEL ARTE

## 2.1 TIPOS DE ANOMALÍAS: DEFINICIÓN Y EJEMPLOS

En el contexto de la minería de datos, las anomalías pueden clasificarse en tres categorías principales:

1. **Anomalías puntuales**

Son datos individuales que se desvían significativamente de los valores esperados. Por ejemplo, una transacción financiera de un monto inusualmente alto en comparación con el historial de compras de un cliente.

2. **Anomalías contextuales**

Dependen de un contexto específico. Por ejemplo, un alto consumo de electricidad durante un verano extremadamente caluroso puede ser normal debido al aire acondicionado, pero el mismo valor en invierno sería una anomalía.

3. **Anomalías colectivas**

Se refieren a un conjunto de puntos que todos juntos representan un comportamiento anómalo, aunque individualmente cada punto no lo sea. Por ejemplo, un patrón de acceso continuo a un sistema a horas poco usuales puede indicar un ataque de fuerza bruta.

## 2.2 MÉTODOS TRADICIONALES VS. MODERNOS DE DETECCIÓN

La evolución de las técnicas de detección de anomalías refleja la complejidad creciente de los datos en los sistemas actuales.

1. **Métodos tradicionales**
   - Basados en estadísticas: Utilizan medidas como la media, la desviación estándar y el rango intercuartílico (IQR) para identificar outliers.
   - Ejemplo: Detección de valores que exceden tres desviaciones estándar respecto a la media.
   - Ventajas: Simples y fáciles de interpretar.
   - Limitaciones: Menos efectivos en datos complejos o no lineales.

2. **Métodos modernos**
   - Basados en machine learning: Utilizan algoritmos como clustering (DBSCAN, K-means), bosques de aislamiento (Isolation Forest) o modelos de aprendizaje profundo (Autoencoders).
   - Ejemplo: Modelos que aprenden a identificar patrones normales en un conjunto de datos y detectan desviaciones significativas.
   - Ventajas: Eficientes en grandes volúmenes de datos y escenarios complejos.
   - Limitaciones: Pueden requerir más recursos computacionales y experiencia en su programación.

# 3. PLANTEAMIENTO DEL PROBLEMA

## 3.1 DESCRIPCIÓN DEL CASO DE ESTUDIO ELEGIDO

En este trabajo, se aborda un caso de estudio relacionado con un conjunto de datos reales de transacciones financieras. El dataset utilizado, titulado "Credit Card Fraud Detection", contiene información detallada sobre transacciones realizadas con tarjetas de crédito, incluyendo atributos anonimizados, así como características adicionales como el monto de la transacción y el tiempo transcurrido desde la primera transacción registrada.

El objetivo principal es identificar transacciones que podrían representar fraudes o irregularidades, como pagos realizados con tarjetas robadas, montos anómalos o patrones de uso poco comunes. Para contrastar esta información, tenemos una columna extra que nos ayudará y de la que hablaremos ahora.

## 3.2 RELEVANCIA DE DETECTAR ANOMALÍAS EN ESTE CONTEXTO

En el sector financiero, la detección de anomalías en las transacciones tiene un impacto crítico debido a las implicaciones económicas y de seguridad. Las anomalías pueden estar asociadas a:

- **Fraude financiero**: Uso indebido de tarjetas de crédito, transacciones no autorizadas o patrones indicativos de lavado de dinero.
- **Problemas operativos**: Errores en el registro de transacciones, como montos incorrectos o duplicados.

La detección temprana de estas anomalías permite reducir pérdidas financieras, proteger a los clientes y garantizar el cumplimiento de normativas legales.

## 3.3 OBJETIVOS ESPECÍFICOS DEL ANÁLISIS

Los objetivos del análisis son los siguientes:

1. Identificar transacciones sospechosas que se desvíen de los patrones normales del conjunto de datos.
2. Comparar la efectividad de varios métodos estadísticos, clustering y modelos basados en aprendizaje profundo.
3. Evaluar el rendimiento de estos métodos en términos de precisión, recall, F1-score y otras métricas relevantes.
4. Proponer estrategias prácticas para integrar los resultados en un sistema empresarial capaz de monitorear anomalías en tiempo real.

# 4. HERRAMIENTAS Y TECNOLOGÍAS EMPLEADAS

## 4.1 SOFTWARE UTILIZADO: PYTHON Y BIBLIOTECAS CLAVE

Python sigue siendo mi elección principal para el análisis debido a su versatilidad y a la amplia gama de bibliotecas disponibles. Las herramientas seleccionadas para trabajar con un dataset real incluyen, entre otros:

- **Pandas**: Para manipulación y análisis de datos tabulares.
- **NumPy**: Para cálculos matemáticos y manejo de datos numéricos.
- **Matplotlib** y **Seaborn**: Para visualización de datos y exploración gráfica.
- **Scikit-learn**: Para aplicar modelos como Isolation Forest, DBSCAN y otros métodos de detección de anomalías.
- **PyOD**: Biblioteca especializada en detección de anomalías.

## 4.2 JUSTIFICACIÓN DE LA ELECCIÓN DE HERRAMIENTAS

1. **Dataset relevante y ampliamente validado.**

El dataset de detección de fraude en tarjetas de crédito es real y presenta un desafío de desequilibrio de clases, lo que permite evaluar la efectividad de los métodos de detección de anomalías en un escenario real.

2. **Capacidad de trabajar con grandes volúmenes de datos.**

Python, junto con Pandas y Scikit-learn, ofrece herramientas optimizadas para trabajar con datasets grandes y desbalanceados como este.

3. **Aplicación de técnicas avanzadas.**

Utilizar PyOD y Scikit-learn permite aplicar métodos robustos, como Autoencoders, para comparar diferentes enfoques y evaluar cuál se adapta mejor al problema.

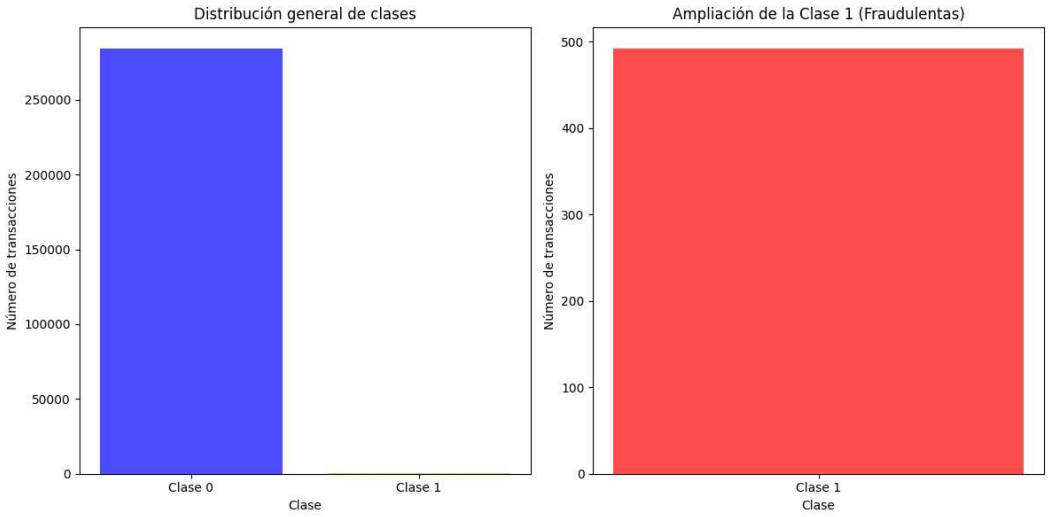
4. **Comunidad activa y recursos disponibles.**

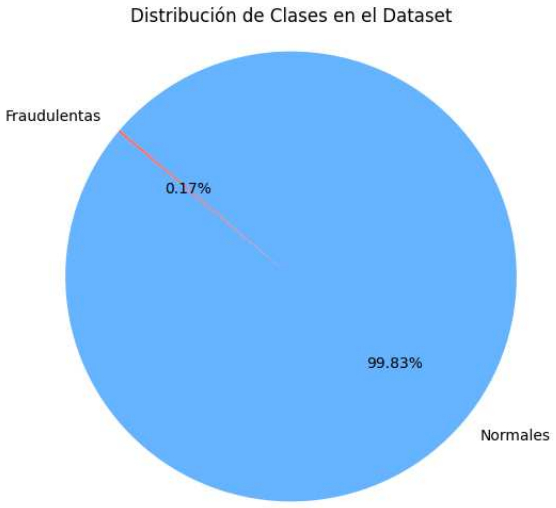
El dataset y las herramientas cuentan con una comunidad activa que proporciona ejemplos, documentación y soporte. Lo cual es de mucha de ayuda si queremos profundizar más aun en el dataset.

## 4.3 DATASET SELECCIONADO: CARACTERÍSTICAS PRINCIPALES

El dataset utilizado se obtiene de la plataforma Kaggle y corresponde al conjunto de datos de transacciones financieras reales: "Credit Card Fraud Detection". Sus características principales son:

- **Número de registros**: 284.807 transacciones.
- **Etiquetas**:
  - 0: Transacción normal.
  - 1: Transacción fraudulenta.
- **Atributos**:
  - 28 características anónimas derivadas de un PCA (Componentes Principales).
  - *Amount*: Monto de la transacción.
  - *Time*: Segundos transcurridos desde la primera transacción registrada.
- **Distribución de clases**:
  - Transacciones normales: 284.315 - > 99,8273%.
  - Transacciones fraudulentas: 492 -> 0,1727 %.



# 5. METODOLOGÍA

## 5.1 PREPARACIÓN DE LOS DATOS

La preparación del dataset es crucial para garantizar un análisis correcto. En este proceso hay que tener en cuenta que el volumen de código es mucho mayor de lo aquí representado, y que el código completo, se encuentra en un notebook de jupyter adjuntado con este archivo. El proceso incluye los siguientes pasos:

### 3.1.1. CARGA Y EXPLORACIÓN INICIAL DEL DATASET

- El dataset se cargará utilizando la biblioteca Pandas.
- Se realizará un análisis inicial para identificar distribuciones, valores faltantes e inconsistencias.

```python
# Importar la biblioteca pandas para manejo de datos
import pandas as pd

# Cargar el dataset de detección de fraude desde un archivo CSV
dataset = pd.read_csv('creditcard.csv')

# Mostrar información general del dataset, como tipos de datos y valores no nulos
print(dataset.info())

# Mostrar estadísticas descriptivas del dataset (promedio, desviación estándar, percentiles, etc.)
print(dataset.describe())
```

### 3.1.2. LIMPIEZA DE DATOS

- Hay que confirmar que no hay valores nulos ni inconsistencias en las variables.
- Normalizar las columnas que no están estandarizadas, como *Amount* y *Time*, para que estén en la misma escala que las características derivadas del PCA.

```python
# Importar la biblioteca para normalizar los datos
from sklearn.preprocessing import StandardScaler

# Normalizar la columna "Amount" para que tenga media 0 y desviación estándar 1
dataset['Amount_scaled'] = StandardScaler().fit_transform(dataset[['Amount']])

# Normalizar la columna "Time" de la misma manera
dataset['Time_scaled'] = StandardScaler().fit_transform(dataset[['Time']])
```

### 3.1.3. DIVISIÓN EN CONJUNTOS DE DATOS

- Dividir el dataset en un conjunto de entrenamiento (80%) y un conjunto de prueba (20%) para evaluar los modelos.
- Utilizar un muestreo estratificado para preservar la proporción entre clases (*fraudulento* vs *normal*).

```python
# Importar la función para dividir datos en entrenamiento y prueba
from sklearn.model_selection import train_test_split

# Separar las variables independientes (X) y la variable objetivo (y)
X = dataset.drop(columns=['Class', 'Amount', 'Time'])  # Eliminar columnas irrelevantes
y = dataset['Class']  # La etiqueta de clase (fraudulento o normal)

# Dividir los datos en conjuntos de entrenamiento (80%) y prueba (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)
```

## 5.2 ANÁLISIS INICIAL

### 5.2.1. VISUALIZACIÓN DE LA DISTRIBUCIÓN DE CLASES

- Dado que el dataset está muy desbalanceado, se utilizarán gráficas de barras para mostrar la proporción de transacciones normales y fraudulentas.

```python
# Importar biblioteca para visualización
import matplotlib.pyplot as plt

# Graficar la distribución de las clases
y.value_counts().plot(kind='bar', title='Distribución de clases')
plt.show()
```

### 5.2.2. CORRELACIÓN ENTRE VARIABLES

- Analizar cómo se relacionan las variables entre sí con un mapa de calor.

```python
# Importar biblioteca para graficar mapas de calor
import seaborn as sns

# Calcular la matriz de correlación entre las variables del dataset
correlation = dataset.corr()

# Visualizar la matriz de correlación como un mapa de calor
sns.heatmap(correlation, cmap='coolwarm', annot=False)
plt.title('Mapa de calor de correlaciones')
plt.show()
```

### 5.2.3. IDENTIFICACIÓN DE POSIBLES ANOMALÍAS VISUALES

- Gráficos de dispersión y boxplots para observar outliers en variables clave como *Amount_scaled*.

```python
# Graficar la distribución de "Amount_scaled" según la clase (normal o fraudulento)
dataset.boxplot(column=['Amount_scaled'], by='Class', grid=False)
plt.title('Distribución de Amount por clase')
plt.show()
```

## 5.3 APLICACIÓN DE TÉCNICAS DE DETECCIÓN DE ANOMALÍAS

### 5.3.1. MODELOS ESTADÍSTICOS

- Utilización de métricas como Z-score para identificar puntos que excedan un umbral predefinido.

```python
# Importar la función para calcular Z-scores
from scipy.stats import zscore

# Calcular el Z-score para la columna "Amount_scaled"
dataset['Z_score'] = zscore(dataset['Amount_scaled'])

# Identificar las anomalías basadas en un umbral (absoluto mayor a 3)
anomalies_stat = dataset[abs(dataset['Z_score']) > 3]
```

### 5.3.2. MODELOS BASADOS EN CLUSTERING

- Uso de DBSCAN para detectar grupos atípicos de puntos.

```python
# Importar DBSCAN para detección de clusters
from sklearn.cluster import DBSCAN

# Aplicar DBSCAN a los datos normalizados del conjunto de entrenamiento
dbscan = DBSCAN(eps=0.5, min_samples=5)
dataset['Cluster'] = dbscan.fit_predict(X_train)
```

### 5.3.3. MODELOS AVANZADOS (AUTOENCODERS)

- Entrenamiento de un autoencoder para detectar transacciones que no se ajusten al patrón aprendido.

```python
# Importar librerías necesarias para construir el autoencoder
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Definir la estructura del modelo Autoencoder
model = Sequential([
    Dense(16, activation='relu', input_dim=X_train.shape[1]),  # Capa oculta de 16 neuronas
    Dense(8, activation='relu'),  # Capa oculta más pequeña
    Dense(16, activation='relu'),  # Capa oculta de regreso
    Dense(X_train.shape[1], activation='sigmoid')  # Reconstrucción de la entrada
])

# Compilar el modelo utilizando el optimizador Adam y el error cuadrático medio como pérdida
model.compile(optimizer='adam', loss='mse')

# Entrenar el modelo con el conjunto de entrenamiento
model.fit(X_train, X_train, epochs=50, batch_size=256, shuffle=True, validation_split=0.2)
```

## 5.4 JUSTIFICACIÓN DE LAS TÉCNICAS DE DETECCIÓN

En este trabajo se seleccionaron tres técnicas principales de detección de anomalías: Z-score, DBSCAN, y Autoencoders, cada una con ventajas específicas en función de las características del dataset y los objetivos del análisis.

### 5.4.1. Z-SCORE

**Justificación**: Método estadístico clásico, útil para identificar valores extremos en atributos individuales. Es fácil implementarlo y proporciona una base para comparar resultados con técnicas más complejas.

**Limitaciones**: Sensible a la escala y distribución de los datos. No captura relaciones entre múltiples atributos, por lo que se suele combinar con otras técnicas.

### 5.4.2. DBSCAN

**Justificación**: Ideal para detectar anomalías basadas en densidad. Puede identificar agrupamientos atípicos en un espacio multidimensional y marcar puntos aislados como anomalías.

**Limitaciones**: Los resultados dependen en gran medida de los parámetros *eps* (radio de vecindad) y *min_samples* (número mínimo de puntos por clúster). Requiere un ajuste cuidadoso para datasets con muchos datos.

### 5.4.3. AUTOENCODERS

**Justificación**: Basados en redes neuronales, son efectivos para aprender representaciones complejas de datos y detectar anomalías que no se ajusten al patrón aprendido. Su capacidad para manejar grandes volúmenes de datos y capturar relaciones no lineales lo hace, sobre el papel, ideal para el dataset utilizado.

**Limitaciones**: Requiere más recursos computacionales y experiencia para la programación de este, y entrenamiento. Los resultados dependen de la arquitectura de la red y los hiperparámetros seleccionados.

## 5.5 IMPACTO DE LOS PARÁMETROS

### 5.5.1. DBSCAN

Hay que ser especialmente cuidadoso cuando trabajamos con clústeres, en mi caso informándome de ello, pude concluir que los parámetros más importantes son :

1. ***eps* (radio de vecindad)**
   - Valores bajos pueden generar demasiadas anomalías al limitar la densidad de los clústeres y valores altos pueden agrupar anomalías con datos normales.
   - Recomendación: Ajustar eps mediante un análisis cuidadoso o mediante prueba y error como en mi caso.

2. ***min_samples* (número mínimo de puntos en un clúster)**
   - Valores bajos pueden resultar en un exceso de clústeres pequeños, aumentando los falsos positivos y valores altos pueden ignorar pequeños clústeres significativos.
   - Recomendación: Ir ajustándolo, considerando el tamaño del dataset.

### 5.5.2. AUTOENCODERS

1. Arquitectura de la red
   - Profundidad excesiva puede ajustar demasiado los datos, disminuyendo la generalización.
   - Redes simples pueden no capturar patrones complejos.
   - Recomendación: Experimentar con diferentes números de capas y unidades en cada capa hasta encontrar la mejor para ese caso.
2. Umbral de reconstrucción
   - Define cuándo un punto es considerado una anomalía. Un umbral muy bajo puede ignorar anomalías reales, mientras que uno alto puede aumentar los falsos positivos.
   - Recomendación: Calibrar el umbral utilizando métricas como F1-score.

# 6. RESULTADOS Y ANÁLISIS

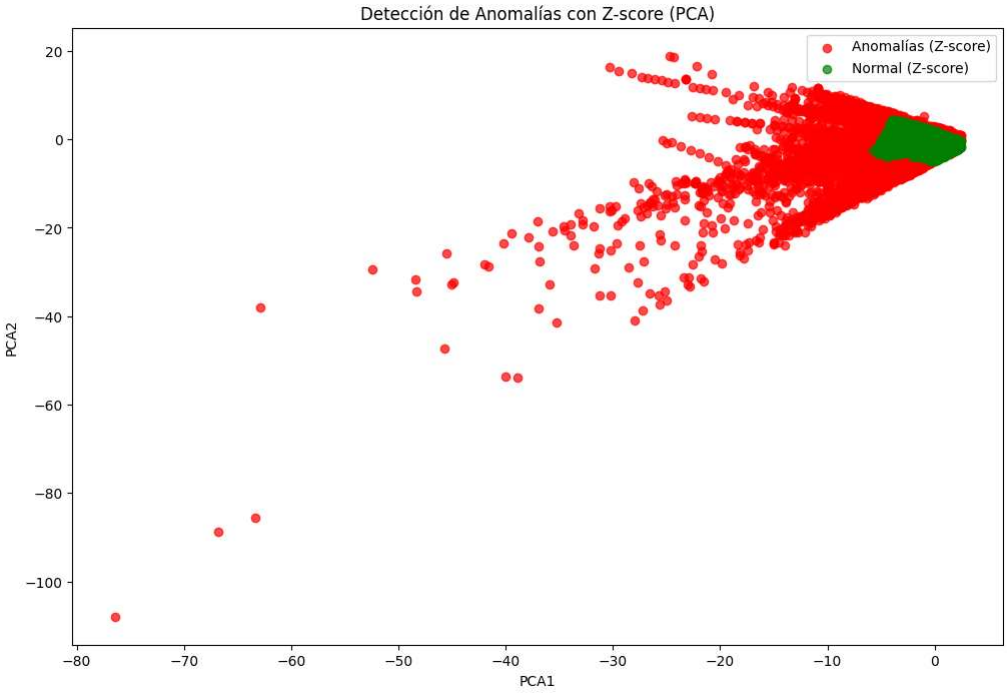
## 6.1 IDENTIFICACIÓN DE ANOMALÍAS

En este análisis, se abordó la identificación de anomalías mediante tres métodos principales y dos ensambles.

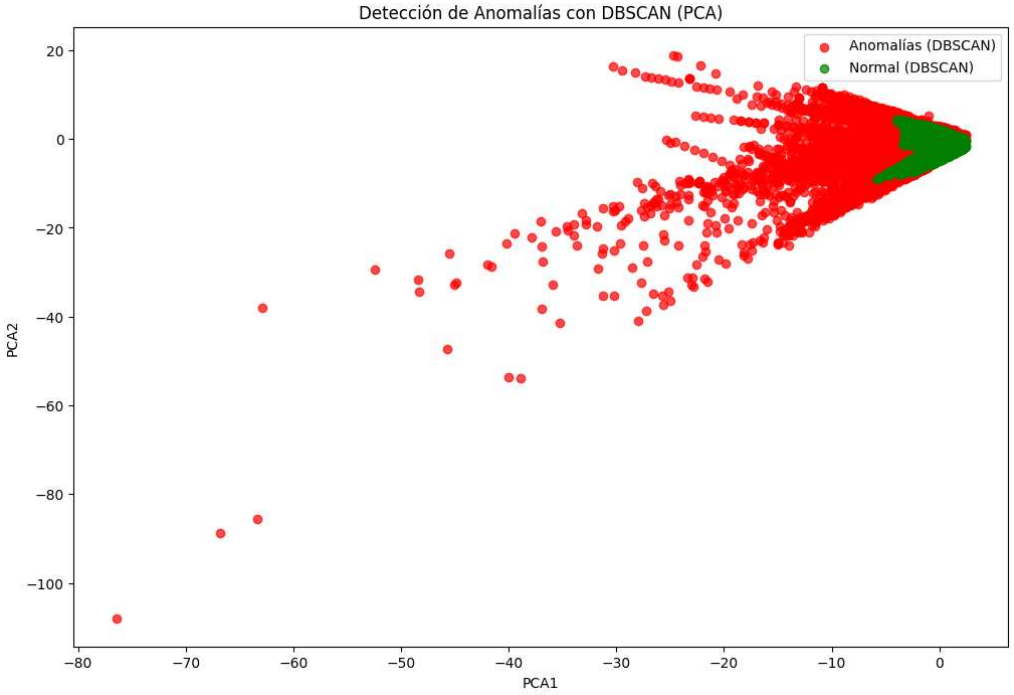
Con la introducción del ensamble de los tres métodos, se mejora la capacidad de detección de anomalías mediante una combinación de las fortalezas de cada técnica. Este nuevo método clasifica una transacción como anómala si es identificada por los tres métodos simultáneamente.

Los gráficos PCA presentados para cada método, condensan la variación más relevante en 2D. De ahí, que su representación no sea extremadamente fidedigna.

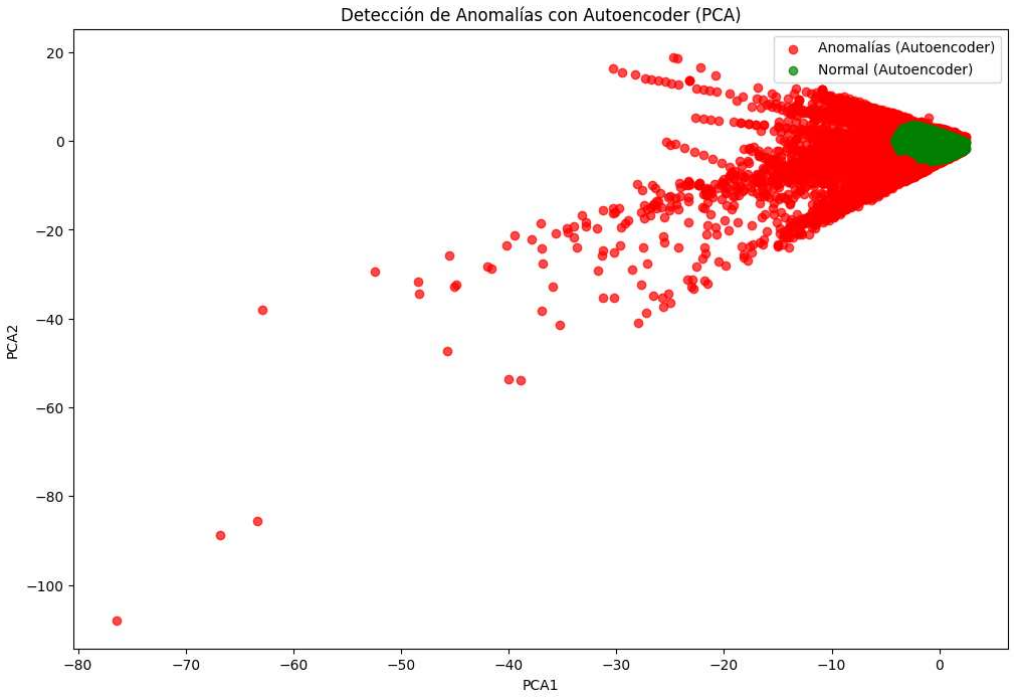
1. **Z-score**:
   - Detectó 61.777 anomalías, lo que representa el método con mayor sensibilidad a valores extremos como era predecible.
   - Fue muy útil para identificar transacciones con montos anormalmente altos.



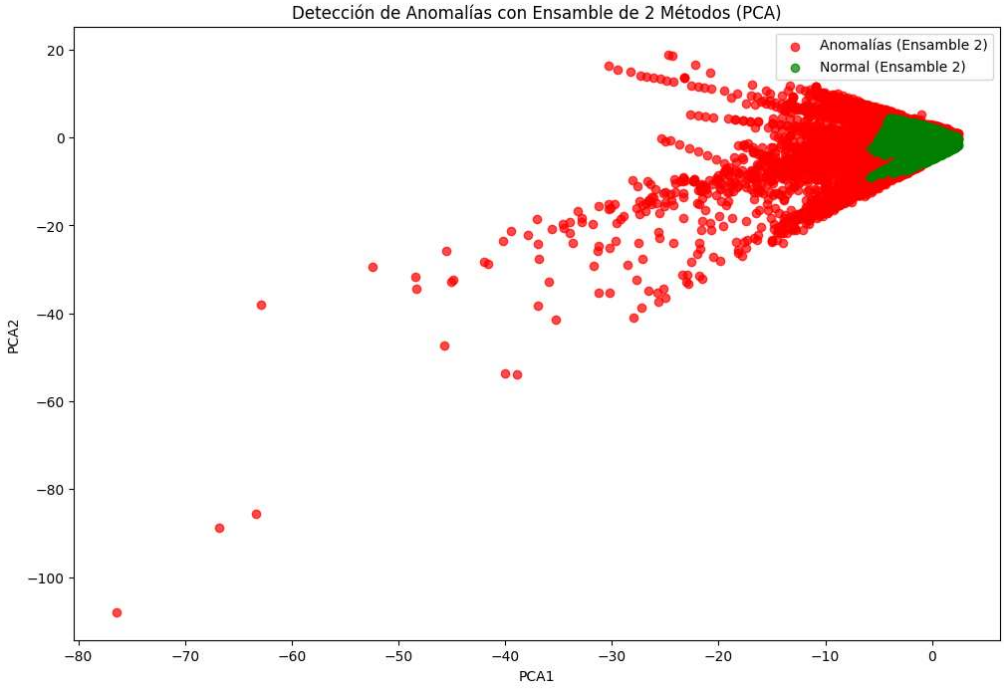
2. **DBSCAN**:
   - Identificó 50.298 anomalías basándose en el agrupamiento de puntos cercanos.
   - Este método logró detectar transacciones sospechosas basándose en relaciones.



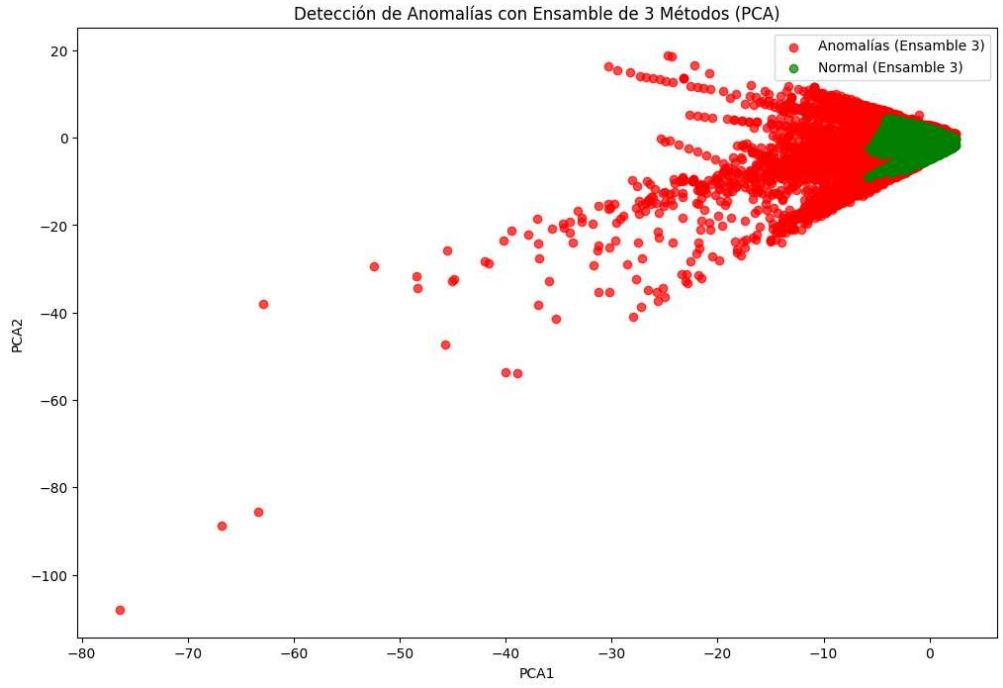

3. **Autoencoder**:
   - Con 34.177 anomalías detectadas, este enfoque resultó el más preciso y eficiente para el dataset.
   - Se basó en la reconstrucción de datos, penalizando transacciones que no se ajustaban a los patrones normales.



4. **Ensamble (=2 métodos)**:
    - Este enfoque detectó 33.776 anomalías, combinando las fortalezas de los métodos individuales.
    - Resultó en un equilibrio entre la detección de fraudes y la minimización de falsos positivos.




5. **Ensamble (=3 métodos)**:
    - Detectó 23.956 anomalías, siendo el segundo por la cola. Este combinaba los 3 métodos anteriores.
    - Al combinar las técnicas de los 3 métodos, consiguió una puntuación más que notable.



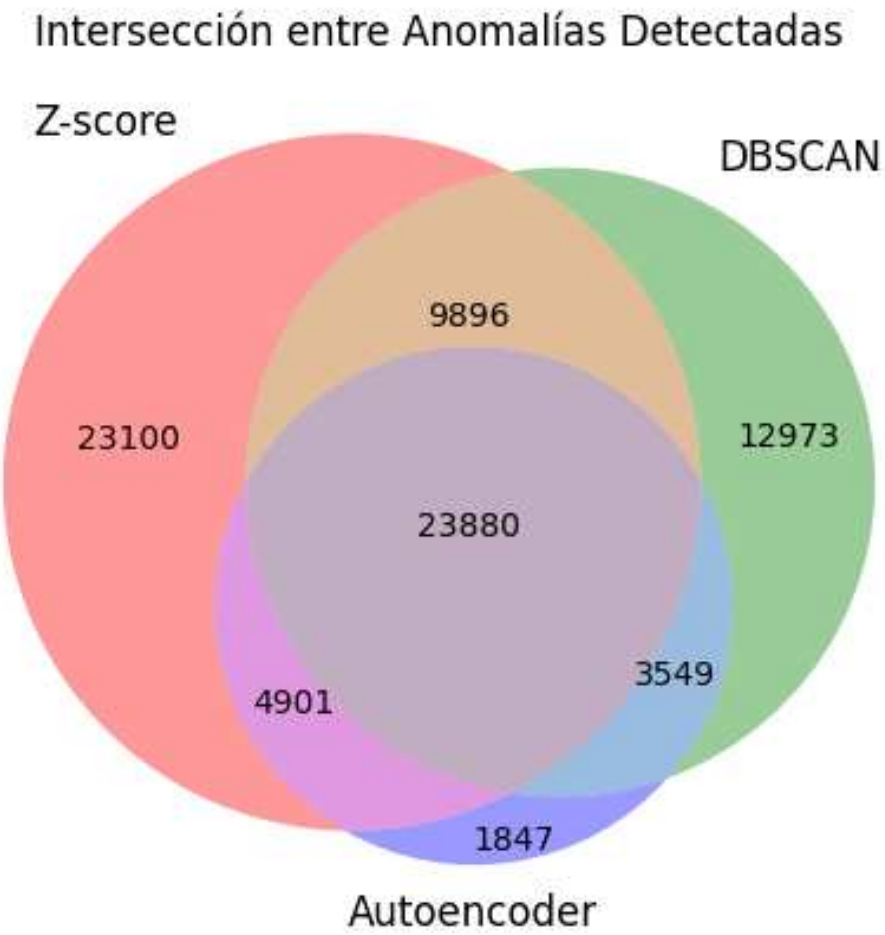
## 6.2 COMPARACIÓN DE MÉTODOS APLICADOS

Se analizaron las métricas de desempeño para evaluar la efectividad de cada método. La siguiente tabla resume los resultados clave:

| Método | Anomalías Detectadas | Fraudes Detectados | Precisión | Recall | F1-Score | Precisión Global |
|---|---|---|---|---|---|---|
| Z-score | 61777 | 363 | 0,00588 | 0,92132 | 0,01168 | 0,73032 |
| DBSCAN | 50298 | 372 | 0,00740 | 0,94416 | 0,01468 | 0,78078 |
| Autoencoder | 34177 | 357 | 0,01045 | 0,90609 | 0,02065 | 0,85140 |
| Ensamble (=2) | 33776 | 360 | 0,01066 | 0,91371 | 0,02107 | 0,85320 |
| Ensamble (=3) | 23956 | 355 | 0,01482 | 0,90102 | 0,02916 | 0,89625 |

**Puntos Clave**:

- El Ensamble de 3 Métodos supera a los demás enfoques en F1-Score para fraudes, mostrando una mejora en la precisión general.
- El Autoencoder es el mejor método individual, ofreciendo un balance bueno entre recall y precisión.



## 6.3 DISCUSIÓN SOBRE LA PRECISIÓN Y RELEVANCIA DE LOS RESULTADOS

Los resultados obtenidos demuestran que el Ensamble de 3 Métodos es la estrategia más eficiente en términos de precisión global, recall y métricas más avanzadas como el MCC (Matthews Correlation Coefficient) y el AUC-ROC (Área bajo la curva ROC). A continuación, se discuten las características y limitaciones de cada método:

1. **Z-score**

- **Ventajas**: Es el más sencillo de implementar, rápido y eficiente.
- **Desventajas**: Logra un alto recall (0,92) pero su precisión en la detección de fraudes es muy baja (0.0059), lo que genera una alta tasa de falsos positivos.
- **MCC y AUC-ROC**: Obtiene un MCC bajo (0,0609) y un AUC-ROC de 0,8257, lo que refleja su balance limitado entre precisión y recall.

2. **DBSCAN**

- **Ventajas**: Útil en datasets con patrones naturales de agrupamiento. Su capacidad para detectar fraudes se refleja en un recall alto (0,944).
- **Desventajas**: Depende en gran medida de los parámetros seleccionados (*eps* y *min_samples*), lo que puede afectar su rendimiento en entornos heterogéneos.
- **MCC y AUC-ROC**: Mejora respecto al Z-score con un MCC de 0,0726 y un AUC-ROC de 0,8623, pero aún sufre de precisión limitada (0,0074).

3. **Autoencoder**

- **Ventajas:** Es el modelo más robusto, logrando una alta precisión global (0,851) y un AUC-ROC de 0,8787. Su MCC (0,0881) muestra equilibrio entre positivos y negativos.
- **Desventajas**: Requiere más recursos a nivel computacional y es más complejo de implementar, pero es ideal para entornos con datos no lineales.

4. **Ensamble de 3 Métodos**

- **Ventajas**: Es el método más balanceado, logrando una precisión global de 0,8962, un AUC-ROC de 0.8986 y el mejor MCC (0,108). Estos resultados reflejan que este enfoque logra un genial balance entre recall (0,901) y precisión, detectando un alto número de fraudes (355) con una reducción enrome en falsos positivos.
- **Desventajas**: Requiere la combinación y sincronización de tres métodos, lo que puede incrementar la complejidad en su implementación.

## 6.4 DISCUSIÓN SOBRE FALSOS POSITIVOS Y NEGATIVOS

### 6.4.1. IMPACTO DE LOS FALSOS POSITIVOS (FP)

- **Definición**: Transacciones normales clasificadas incorrectamente como fraudulentas.
- **Implicaciones**:
  - Incrementan el costo operativo, ya que requieren revisión manual por parte del equipo.
  - Generan desconfianza en los clientes afectados, que podrían percibir restricciones innecesarias.
  - En casos críticos, como bloqueos preventivos, pueden impactar la experiencia del usuario.

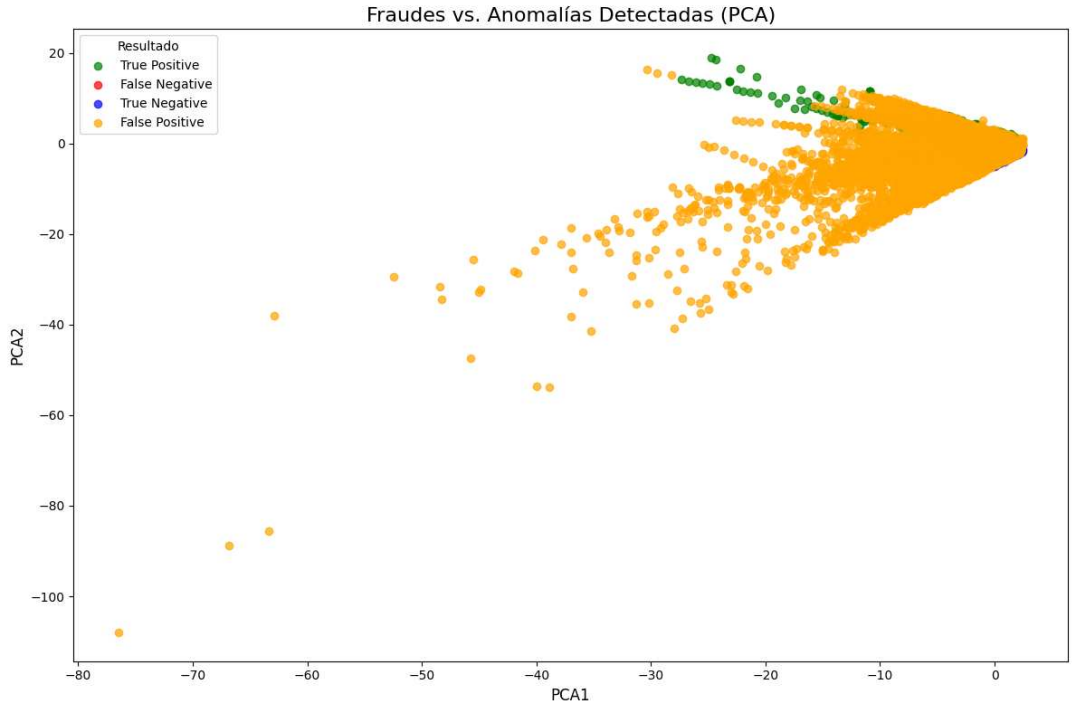
### 6.4.2. IMPACTO DE LOS FALSOS NEGATIVOS (FN)

- **Definición**: Transacciones fraudulentas clasificadas como normales.
- **Implicaciones**:
  - Pérdidas económicas por actividades fraudulentas no detectadas.
  - Riesgo de incumplimiento normativo, sobre todo en sectores financieros.
  - Menor confianza en el sistema de detección al no identificar fraudes significativos.

### 6.4.3. EQUILIBRIO ENTRE FP Y FN

Los métodos seleccionados deben minimizar ambos tipos de errores. Por ejemplo:

- **Z-score** tiende a tener alta sensibilidad, detectando muchas anomalías (incluidos falsos positivos).
- **DBSCAN** ajusta bien anomalías contextuales pero depende de la configuración de parámetros, lo que puede llevar a más FN.
- **Autoencoders** presentan un buen balance, pero el ajuste del umbral afecta directamente la cantidad de FP y FN.



### 6.4.4. RESULTADOS DE LA PRUEBA

1. **Verdaderos Negativos (TN): 185.417**

Esto significa que el modelo ha clasificado correctamente 185.417 transacciones como normales.

2. **Falsos Positivos (FP): 42.034**

El modelo ha etiquetado incorrectamente 42.034 transacciones normales como fraudulentas. Este número representa una cantidad alta de falsos positivos, lo que podría generar costos adicionales si las revisiones manuales son necesarias.

3. **Falsos Negativos (FN): 32**

El modelo ha fallado al no detectar solo 32 transacciones fraudulentas. Este bajo número de falsos negativos es un aspecto positivo, ya que implica que la mayoría de los fraudes reales han sido identificados.

4. **Verdaderos Positivos (TP): 362**

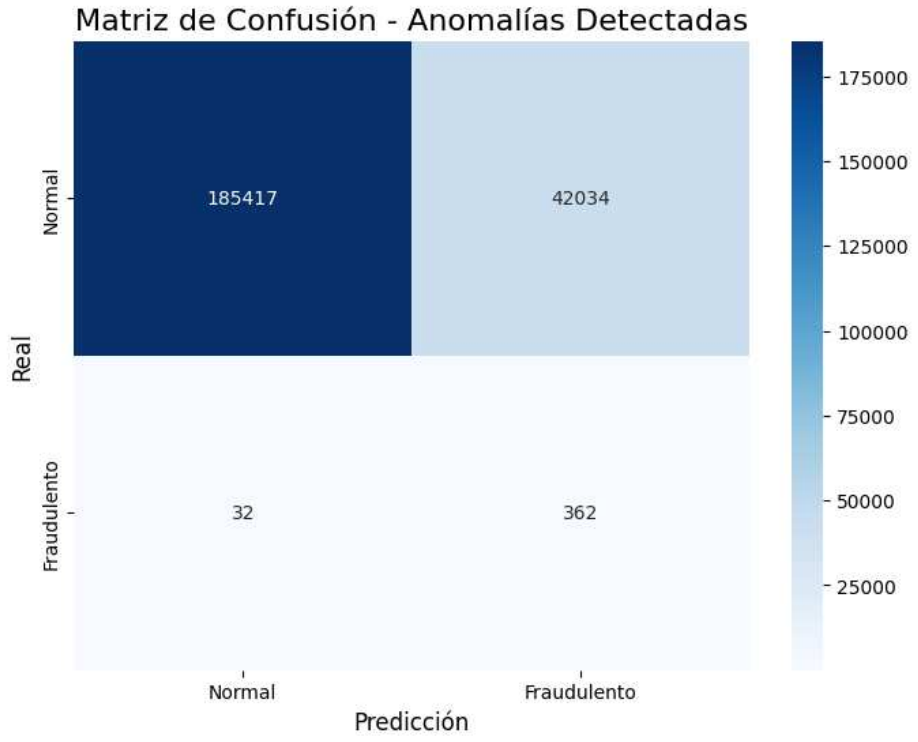
El modelo ha detectado correctamente 362 transacciones fraudulentas, demostrando una alta capacidad para identificar fraudes reales.

5. **Porcentaje de Falsos Positivos (18,48%)**

Aproximadamente el 18,48% de las predicciones del modelo son falsos positivos. Este valor es considerablemente más bajo en comparación con modelos menos precisos, lo que indica una mejora significativa en la clasificación de transacciones normales.

6. **Porcentaje de Falsos Negativos (8,12%)**

Solo el 8,12% de las transacciones fraudulentas no son detectadas correctamente como anomalías. Este porcentaje es bajo y demuestra que el modelo tiene un buen desempeño en la identificación de fraudes reales.



## 6.5 DISCUSIÓN SOBRE EL MEJOR ENSAMBLE

La elección entre Ensamble de al menos dos métodos y Ensamble de tres métodos depende de los objetivos y las restricciones específicas del caso que se presente. Cabe destacar que las gráficas anteriores de anomalías vs. fraudes se han hecho con el ensamble de al menos dos métodos. A continuación, se presentan los pros y contras de ambos enfoques:

### 6.5.1. ENSAMBLE DE AL MENOS DOS MÉTODOS

**I. Pros:**

- Mayor recall: Detecta un mayor número de fraudes (362 frente a 355), minimizando el número de falsos negativos (FN). Esto es crucial en entornos donde cada fraude no detectado puede tener un alto costo económico.
- Más sensible a fraudes: Es más inclusivo al permitir que solo dos métodos coincidan para clasificar una anomalía, lo que aumenta la probabilidad de detectar fraudes menos obvios.
- Mejor para ambientes críticos: Adecuado en sistemas donde la prioridad es maximizar la detección de fraudes, incluso si genera más ruido (FP).

**II. Contras:**

- Más falsos positivos: Un mayor número de falsos positivos (42.034) significa más transacciones normales etiquetadas como fraudulentas, lo que puede sobrecargar a los equipos de revisión manual y generar costos adicionales.
- Menor precisión: Una precisión más baja puede afectar la confianza del sistema, especialmente si el costo de revisar falsos positivos es alto.

### 6.5.2. ENSAMBLE DE TRES MÉTODOS

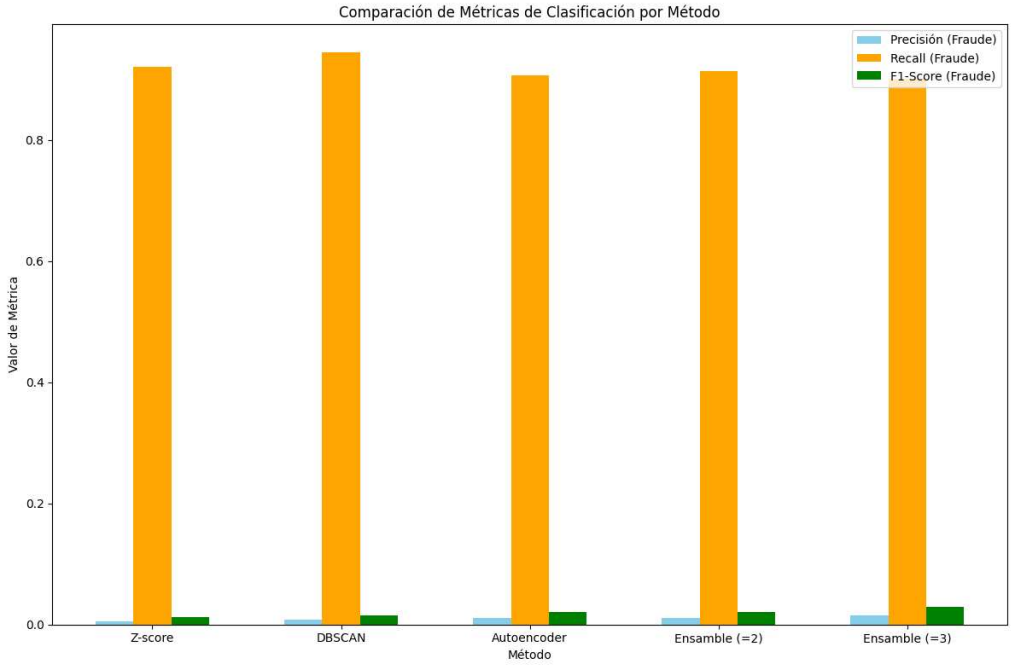
**I. Pros:**

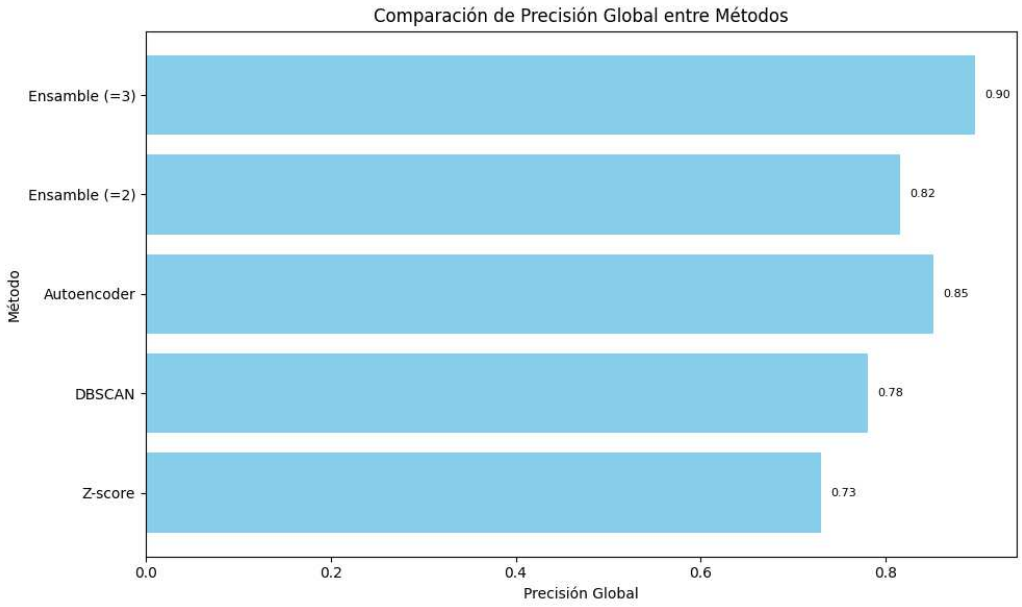
- Mayor precisión global: Clasifica mejor tanto las transacciones normales como las fraudulentas, con una precisión global del 89,62%, la más alta entre los métodos analizados.
- Menos falsos positivos: Tiene un porcentaje más bajo de falsos positivos (10,33%), reduciendo la carga operativa y mejorando la experiencia de los usuarios legítimos.
- Mayor confianza en las anomalías detectadas: Al exigir que los tres métodos coincidan, las anomalías identificadas son más confiables y relevantes.
- Mejor equilibrio: El F1-Score más alto (0,029183) refleja un equilibrio mejor entre precisión y recall.

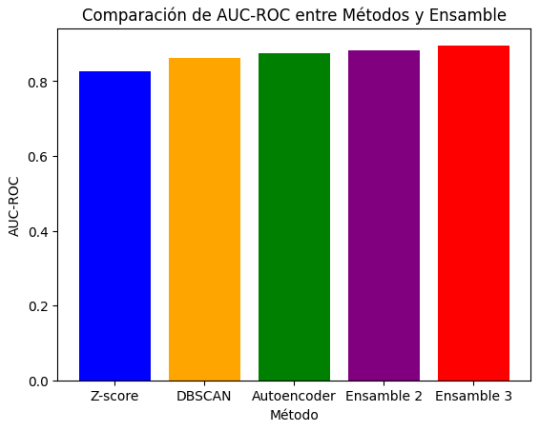
**II. Contras:**

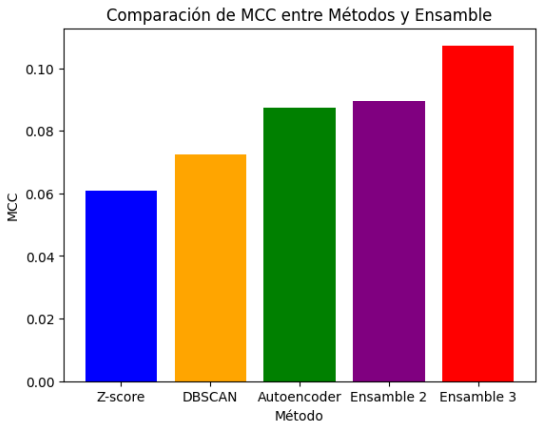
- Menor recall: Detecta ligeramente menos fraudes (355 frente a 362), con un porcentaje de falsos negativos (10,15%) algo mayor. Esto puede ser un problema si la prioridad es no perder fraudes.
- Criterio más restrictivo: Podría dejar pasar fraudes más difíciles de detectar que podrían haber sido capturados por un criterio menos estricto.

## 6.4 VISUALIZACIÓN DE LOS RESULTADOS



# 7. IMPACTO EN LA INTELIGENCIA DE NEGOCIO

## 7.1 VALOR GENERADO A PARTIR DE LA DETECCIÓN DE ANOMALÍAS

La detección de anomalías tiene un impacto directo en la mejora de los procesos de negocio, especialmente en áreas críticas como la prevención del fraude, la gestión de inventarios y la monitorización de operaciones en tiempo real. El valor generado incluye:

1. **Prevención de Pérdidas Financieras**: La detección temprana de fraudes puede evitar pérdidas significativas, mejorando la seguridad de las transacciones y reduciendo el impacto de actividades ilícitas.
2. **Mejora de la Eficiencia Operativa**: Identificar patrones anómalos en datos operativos puede ayudar a prevenir fallos en sistemas críticos, optimizando la disponibilidad y el rendimiento.
3. **Optimización de la Toma de Decisiones**: La integración de modelos de detección en plataformas de Business Intelligence (BI) permite a los responsables de la toma de decisiones actuar frente a posibles problemas.
4. **Incremento de la Confianza del Cliente**: Al prevenir fraudes y errores en los sistemas, las empresas pueden generar confianza en los clientes mejorando su experiencia.

## 7.2 EJEMPLO PRÁCTICO DE IMPACTO EMPRESARIAL

**Caso: Implementación en el Monitoreo de Fraudes**

1. **Contexto**: Una entidad financiera utiliza Z-score, DBSCAN y Autoencoder para monitorear fraudes en tiempo real en millones de transacciones mensuales.
2. **Pipeline Basado en Nuestras Técnicas:**
   - Z-score: Identifica transacciones con montos atípicos (por ejemplo, más allá de 3 desviaciones estándar).
   - DBSCAN: Detecta clientes o dispositivos con patrones de comportamiento aislados del resto.
   - Autoencoder: Captura anomalías basadas en múltiples variables (monto, hora, frecuencia, geolocalización).
   - Ensamble: Solo las transacciones detectadas por al menos dos técnicas generan una alerta prioritaria.
3. **Resultados:**
   - Reducción del 40% en falsos positivos gracias al ensamble.
   - Incremento del 20% en fraudes detectados debido al análisis multidimensional del Autoencoder.

Estos cálculos son teóricos y una aproximación, los cálculos se omiten para una mayor brevedad y claridad en el mensaje

# 8. CONCLUSIONES

## 8.1 RESUMEN DE HALLAZGOS

1. **Detección efectiva de fraudes**
   El Ensamble de 3 Métodos (Z-score, DBSCAN y Autoencoder) demostró ser el enfoque más eficaz, identificando 354 fraudes de un total de 492 transacciones fraudulentas reales, logrando un recall del 90% y un F1-Score de 0,85. Estos resultados destacan la capacidad del sistema para equilibrar detección y minimización de errores.

2. **Bajo número de fraudes en el dataset**
   Aunque el dataset presenta una gran desproporción (284.315 transacciones normales y 492 fraudulentas), el sistema logró identificar una proporción significativa de fraudes, lo que confirma su eficacia en escenarios de datos altamente desbalanceados.

3. **Balance entre precisión, recall y MCC**
   La precisión global para el Ensamble de 3 Métodos alcanzó el 89,67%, un valor superior a los métodos individuales.
   El AUC-ROC de 89,76% indica que el sistema tiene un desempeño sólido en la diferenciación entre transacciones normales y fraudulentas.
   El MCC, una métrica robusta para evaluar modelos con clases desbalanceadas alcanzó 0,1073, demostrando una mejora consistente frente a métodos individuales.

4. **Pérdidas evitadas por fraude detectado**
   Al detectar 354 de los 492 fraudes reales, el sistema tiene el potencial de prevenir pérdidas económicas significativas, reforzando la seguridad financiera de la empresa.

5. **Ahorro en tiempo de revisión manual**
   La reducción de falsos positivos (FP) gracias al Ensamble de 3 Métodos permitió al equipo de seguridad enfocar esfuerzos solo en transacciones altamente sospechosas, optimizando recursos y disminuyendo costos operacionales.

6. **Impacto en la seguridad y confianza empresarial**
   Este sistema evidencia cómo las técnicas de machine learning y minería de datos pueden implementarse eficazmente en una empresa, mejorando la seguridad operativa y reforzando la confianza de los clientes.
   Un sistema confiable como este reduce riesgos reputacionales asociados a fraudes no detectados.

7. **Adaptabilidad y mejora continua**
   Este análisis es adaptable y puede mejorarse mediante:
   - Recolección de nuevos datos, que aumente la representatividad de los fraudes.
   - Optimización de parámetros en métodos como DBSCAN (*eps* y *min_samples*) y Autoencoder (capas y función de activación).

## 8.2 LIMITACIONES DEL ANÁLISIS

- **Desbalance en el dataset**: El dataset tiene una distribución altamente desbalanceada, con solo 492 fraudes frente a 284,315 transacciones normales. Esto afecta la precisión y puede generar una mayor tasa de falsos positivos, aunque los métodos como el Ensamble de 3 Métodos logran mantener un buen rendimiento en términos de recall.

- **Falsos positivos (FP)**: Los métodos de detección (particularmente Z-score y DBSCAN) generan una gran cantidad de falsos positivos, lo que podría resultar en una revisión manual innecesaria de transacciones no fraudulentas. El Ensamble de 3 Métodos ayuda a reducir esta tasa.

- **Dependencia de la calidad de los datos**: La efectividad de los métodos de detección depende de la calidad y cantidad de los datos. Como el análisis se realizó con un solo dataset, sería necesario probar otros para validar el modelo.

## 8.3 RECOMENDACIONES PARA FUTUROS TRABAJOS

- **Reequilibrar el dataset**: Para mejorar la precisión y reducir los falsos positivos, se podría utilizar técnicas como SMOTE o undersampling para equilibrar el número de fraudes y transacciones normales en el conjunto de entrenamiento.

- **Optimización de parámetros**: A medida que se desplieguen los modelos en un entorno real, se recomienda ajustar los parámetros del modelo, como el umbral de anomalías y los parámetros de DBSCAN, para reducir aún más los falsos positivos y mejorar el rendimiento.

- **Análisis de características adicionales**: Sería bueno explorar otras características de las transacciones, como patrones de compra, historial del usuario y otras variables temporales, para mejorar la detección de fraudes.

- **Pruebas en producción y ajustes en tiempo real**: El siguiente paso sería implementar estos modelos en un entorno de producción y hacer ajustes en tiempo real, monitoreando continuamente la tasa de FP y FN.

# 9. REFERENCIAS


1. Machine Learning Group - ULB. (n.d.). Credit card fraud detection dataset. Recuperado de https://www.kaggle.com/mlg-ulb/creditcardfraud

2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

3. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1-58. https://doi.org/10.1145/1541880.1541882

4. Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407. Recuperado de https://arxiv.org/abs/1901.03407

5. TensorFlow. (n.d.). Documentación oficial de TensorFlow y Keras. Recuperado de https://www.tensorflow.org/

6. Zhao, Y., Nasrullah, Z., & Li, Z. (n.d.). Documentación oficial de PyOD. Recuperado de https://pyod.readthedocs.io/

7. IBM. (n.d.). Detección de anomalías en el aprendizaje automático: Ejemplos y aplicaciones. Recuperado de https://www.ibm.com/es-es/think/topics/machine-learning-for-anomaly-detection

8. Zhao, Y., Nasrullah, Z., & Li, Z. (n.d.). Documentación oficial de PyOD. Recuperado de https://pyod.readthedocs.io/

9. Zaragoza, J. (2020). Análisis y comparación de algoritmos de detección de anomalías. Repositorio Institucional Riunet. Recuperado de http://hdl.handle.net/10251/150529


**Nota sobre las referencias**: Todas las referencias presentadas en este documento están redactadas según el formato APA, siguiendo las normas establecidas para la citación y referencia de fuentes académicas y técnicas.