

# La Analítica de Datos y el Comportamiento del Fraude Bancario

Jonathan Estiven Fontalvo Aparicio  
Departamento de Ingeniería de  
Sistemas y Computación  
Universidad del Norte  
Barranquilla, Colombia  
jfontalvoe@uninorte.edu.co

Ehider Andres Guarin Ortega  
Departamento de Ingeniería de  
Sistemas y Computación  
Universidad del Norte  
Barranquilla, Colombia  
ehiderg@uninorte.edu.co

Mario David Diaz Antequera  
Departamento de Ingeniería de  
Sistemas y Computación  
Universidad del Norte  
Barranquilla, Colombia  
mantequerad@uninorte.edu.co

## I. INTRODUCTION

En un mundo globalizado por el internet, los pagos en línea se han convertido en algo tan normal como lo ha sido el pago en efectivo, pero esto a causado una gran preocupación debido a que los delitos financieros, definidos como prácticas manipuladoras utilizadas para obtener beneficios financieros, se han convertido recientemente en un problema tan omnipresente en empresas y organizaciones (Abdulalem, Shukor, et al, 2022). Esto hace que la prevención y detección del fraude sean componentes críticos para la viabilidad a largo plazo de cualquier negocio de comercio electrónico (Tax, Vries, et al, 2021). Por esta razón, en actualidad el uso de herramientas de Machine Learning basadas en el análisis de la cantidad masiva de datos que se producen cada segundo es viable para la prevención y detección de fraudes. Las empresas han descubierto que es concebible aprovechar una gran cantidad de información para producir respuestas a problemas que tienen objetivos comerciales lejanos. Con la disponibilidad de datos ricos, algoritmos innovadores y métodos novedosos, las instituciones comerciales, financieras y de seguros continúan siendo algunos de los sectores más importantes con un potencial extremadamente alto para aprovechar el aprendizaje automático (ML) y la inteligencia artificial (IA) (Fraud Detection at Claims, 2023). El propósito de esta investigación es probar diferentes modelos y escoger el más acertado en términos de precisión con el fin de implementar un prototipo de servicio de prevención y detección de fraude en transacciones online, tales como: Pagos, transferencias, depósitos, retiros y débitos.

## II. PROBLEMA

El fraude en las transacciones en línea se ha convertido en una preocupación creciente tanto para empresas como para usuarios en la era digital. Esta problemática se manifiesta a través de diversas formas de engaño y manipulación, perpetradas por estafadores cada vez más hábiles y astutos. Estos individuos logran acceder a información sensible y financiera de manera ilegítima, lo que conlleva a considerables pérdidas económicas y daños en la reputación tanto de las empresas afectadas como de los usuarios involucrados.

Las causas del fraude en línea son variadas y complejas. Una de las principales radica en la falta de seguridad en los sistemas de pago utilizados en las plataformas digitales. Los métodos de autenticación de usuarios también presentan debilidades, lo que permite a los estafadores suplantar identidades con relativa facilidad. Además, la carencia de herramientas efectivas para la

detección temprana de fraudes contribuye a la proliferación de este problema.

La falta de conocimiento por parte de los usuarios acerca de las tácticas utilizadas por los estafadores es otro factor crucial. Muchos usuarios no están al tanto de las señales de alerta que podrían indicar una actividad fraudulenta, y algunos incluso caen en prácticas negligentes que facilitan la perpetración de fraudes. La continua evolución de las técnicas empleadas por los estafadores agrega otra capa de complejidad al problema, ya que las estrategias de fraude se adaptan constantemente para eludir las medidas de seguridad establecidas.

En definitiva, el fraude en las transacciones en línea es un desafío multifacético que requiere un enfoque integral para su mitigación. Es fundamental implementar medidas de seguridad robustas en los sistemas de pago y mejorar los protocolos de autenticación de usuarios. Asimismo, el desarrollo y la implementación de herramientas avanzadas de detección de fraudes son esenciales para hacer frente a esta creciente amenaza en el mundo digital.

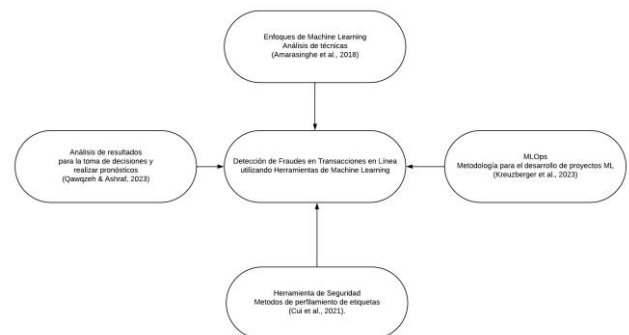


Fig. 1. Arbol del problema.

## III. JUSTIFICACIÓN

El trabajo investigativo para el desarrollo del sistema de detección de fraude de transacciones en línea con el uso de herramientas de Machine Learning se fundamenta en la creciente preocupación pública y la necesidad imperante de abordar el problema del fraude en transacciones financieras. Tradicionalmente, la detección de fraude ha recaído en equipos de auditoría que emplean técnicas manuales, lo cual puede resultar ineficiente y prolongado. Sin embargo, la evolución de la tecnología y el aumento en el volumen de datos han facilitado la aplicación de técnicas de Machine Learning (ML) y Deep Learning para automatizar el proceso de detección de fraudes.

Una revisión exhaustiva de la literatura (Al-Hashedi & Magalingam, 2021; Abdulalem et al., 2022; Sanchez Marco et al., 2021) revela una tendencia hacia la aplicación de técnicas de ML en la detección de fraudes financieros. Estudios como el de Abdul Razaque et al. (2023) y Ansari et al. (2022) han demostrado el potencial de algoritmos de ML, como redes neuronales y algoritmos genéticos, para identificar patrones de fraude en transacciones de tarjetas de crédito y otros métodos de pago. Además, investigaciones como la de Kumar et al. (2022) han destacado la importancia de utilizar algoritmos de ML para analizar el comportamiento de los clientes y prevenir actividades fraudulentas en el sector financiero.

En este contexto, el presente estudio propone explorar y comparar diferentes métodos de detección de fraudes en transacciones financieras utilizando técnicas de ML. Se basará en trabajos previos (Sahin et al., 2013; Awoyemi et al., 2017; Khatri et al., 2020) que han evaluado algoritmos de aprendizaje supervisado, como árboles de decisión y métodos basados en reglas, para identificar transacciones fraudulentas. Esta investigación se justifica por la necesidad de seleccionar el modelo óptimo que se ajuste a la fuente de datos específica y así mejorar la gestión del análisis de los fraudes. Además, se realizará un análisis exhaustivo de las características de estos algoritmos y se evaluará su desempeño en la detección de fraudes. Este enfoque se adopta con el objetivo de proponer una estrategia efectiva y eficiente para mitigar el riesgo de fraude en el sector financiero.

#### IV. OBJETIVOS

##### Objetivo General

Diseñar e implementar un sistema inteligente basado en aprendizaje automático para la detección y prevención de fraudes en transacciones en línea, con el fin de proteger la integridad de las operaciones y garantizar la seguridad de los usuarios.

##### Objetivos Específicos

1. Modelar y diseñar un sistema de aprendizaje automático que permita procesar y analizar datos transaccionales para identificar patrones asociados con actividades fraudulentas en transacciones en línea
2. Desarrollar la arquitectura del sistema inteligente basado en aprendizaje automático que permita procesar y analizar datos transaccionales para identificar patrones asociados con actividades fraudulentas en transacciones en línea, con especial énfasis en la manipulación de transacciones como Cash In, Cash Out, Payment, Transfer and Debit..
3. Implementar el prototipo del sistema inteligente, incluyendo el modelo de aprendizaje automático y la API para su integración, asegurando su confiabilidad y facilidad de uso.
4. Validar el prototipo del sistema inteligente mediante pruebas en entornos simulados y reales, evaluando su efectividad en la detección y prevención de fraudes en transacciones en línea,

así como su capacidad para adaptarse a nuevas formas de fraude.

#### V. METODOLOGÍA

Para el desarrollo del sistema de prevención y detección de fraude en transacciones en línea con el uso de Machine Learning, se ha tenido en cuenta que para poder garantizar el éxito y la eficiencia dentro de la solución desde el momento en que se da lugar a la idea inicial hasta la implementación en producción y su posterior mantenimiento, es necesario el uso de una metodología que asegure dicho proceso. Por consiguiente, se propone hacer uso de la metodología MLOps (Machine Learning Operations), que combina prácticas de desarrollo de software (DevOps) con técnicas específicas de aprendizaje automático (Machine Learning) para mejorar y gestionar el ciclo de vida completo de los modelos de machine learning, desde su desarrollo y entrenamiento hasta su implementación, monitoreo y mantenimiento en producción (Testi et al., 2022).

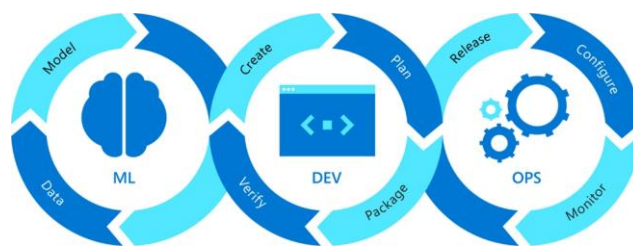


Fig. 2. Modelo MLOps (Tomado de Microcrost Azure Blog)

Para el desarrollo del presente proyecto se realizaron las siguientes fases en base a la metodología escogida con anterioridad:

##### 1. Gestión de datos:

1.1 Adquisición de datos: Recopilación de datos de transacciones en línea, incluidos detalles de transacciones, datos de clientes y cualquier otra información relevante.

1.2 Preprocesamiento de datos: Limpieza de datos, manejo de valores faltantes, normalización y codificación de características, y creación de conjuntos de datos balanceados.

1.3 Gestión de versiones de datos: Seguimiento de los cambios en los conjuntos de datos y aseguramiento de la consistencia entre entornos de desarrollo, prueba y producción.

##### 2. Modelado:

2.1 Desarrollo de modelos: Selección y entrenamiento de algoritmos de machine learning adecuados para la detección de fraudes.

2.2 Evaluación de modelos: Pruebas exhaustivas de los modelos utilizando métricas de rendimiento relevantes, como precisión, recall y F1-score, en conjuntos de datos de prueba y validación.

2.3 Optimización de modelos: Ajuste de hiperparámetros y técnicas de mejora del rendimiento del modelo para maximizar la precisión de la detección de fraudes.

### 3. Operacionalización:

3.1 Implementación de modelos: Despliegue de modelos entrenados en entornos de producción utilizando contenedores o servicios de infraestructura en la nube.

3.2 Monitorización de modelos: Seguimiento del rendimiento de los modelos en tiempo real en producción y detección de degradaciones o cambios en el comportamiento del modelo.

3.3 Retraining automático: Automatización del proceso de reentrenamiento del modelo con nuevos datos para mantener su rendimiento óptimo a lo largo del tiempo.

## VI. MARCO TEORICO

El fraude, que implica engañar para lograr el éxito, ha sido una preocupación constante en instituciones financieras. Tradicionalmente, los equipos de auditoría han empleado técnicas manuales para detectar el fraude, como señalan Qawqzeh y Ashraf (2023), un proceso que resulta lento, costoso, impreciso y poco práctico. En busca de eficiencia, se ha recurrido a herramientas de aprendizaje automático y aprendizaje profundo para automatizar estos procesos y reducir el tiempo de detección.

En los últimos años, el aumento significativo de transacciones electrónicas en la web, impulsado por el crecimiento del comercio electrónico, ha elevado la importancia de abordar los fraudes en estas transacciones (Orche & Bahaj, 2019). Este incremento ha dado lugar a un número significativo de casos de fraude, resultando en pérdidas monetarias sustanciales a nivel mundial. Por ende, se vuelve imperativo desarrollar y aplicar técnicas que contribuyan a la detección eficaz del fraude, según destacan Lima y Pereira (2012), lo que motiva la investigación en este campo.

En el escenario de la detección del fraude en los pagos en línea, las transacciones se caracterizan por atributos. Un sistema de detección de fraudes utiliza los atributos para construir un clasificador binario que distinga las transacciones fraudulentas de las legítimas. El factor clave que afecta a la calidad de un sistema de detección de fraudes es cómo extraer características útiles de los atributos de las transacciones. (Cui et al., 2021).

Con el avance de la inteligencia artificial (IA), el aprendizaje automático y la minería de datos se han utilizado para detectar actividades fraudulentas en el sector financiero. Se han empleado métodos supervisados y no supervisados para predecir las actividades fraudulentas (Ali et al., 2022).

De acuerdo con Cui et al. (2021), podemos distinguir entre dos tipos de métodos según la disponibilidad de etiquetas de transacciones, los basados en el aprendizaje supervisado y los basados en el aprendizaje no supervisado:

Los métodos de aprendizaje supervisado se basan en las características de los datos de las transacciones para construir un clasificador binario (Van Vlasselaer et al., 2015). Los modelos tradicionales de aprendizaje supervisado para la detección de fraudes incluyen la regresión logística (Shen et al., 2007), máquinas de

vectores de soporte (Rtayli & Enneya, 2020) y la red neuronal artificial (Khormuji et al., 2014).

Con el fin de mitigar el problema de sesgo en los datos, tanto el aprendizaje en conjunto como los modelos de aprendizaje sensible al costo se han utilizado ampliamente en la detección supervisada de fraudes. Bian et al. (2016) propusieron un nuevo enfoque de aprendizaje en conjunto para la detección de fraudes financieros mediante el uso simultáneo de los mecanismos de ensacado y aumento. Devi et al. (2018) entrenaron un bosque aleatorio ponderado sensible al costo para la detección de fraudes con tarjetas de crédito, y el método propuesto se beneficia de ambos mecanismos de aprendizaje.

También dentro del ámbito de aprendizaje supervisado, se han evidenciado resultados prometedores en términos de medidas de desempeño en el área de Árboles de Decisión. Qawqzeh y Ashraf (2023) destacan tales hallazgos en el ámbito de las transacciones en línea, demostrando como los Falsos Positivos y Falsos Negativos en una transacción o actividad en línea se pueden reducir en gran medida utilizando la clasificación basada en Árboles de Decisión.

Recientemente, el aprendizaje profundo ha demostrado su eficacia y potencial en la detección supervisada de fraudes. Por ejemplo, la red neuronal convolucional fue empleada por Fu et al. (2016). Para la detección de fraudes en tiempo real, Wang et al. (2017) adoptaron un método basado en redes neuronales recurrentes.

Dentro de las innovaciones más recientes se ha propuesto una Red neuronal de gráfico de doble aumento para tareas de detección de fraude (Li et al., 2022). En los que se buscan mejorar el desempeño generado por los detectores de fraude basados en las Redes neuronales de gráficos. Los resultados experimentales comparados con los modelos de última generación en dos conjuntos de datos del mundo real demuestran la superioridad del modelo propuesto.

Los métodos no supervisados suponen que los datos legítimos son de la clase mayoritaria, y funcionan modelando regularidades de los datos. Las técnicas en uso incluyen algoritmos de agrupamiento (Malini & Pushpa, 2017), mapas auto organizados (Saraswathi et al., 2019) y modelos de aprendizaje profundo (Abakarim et al., 2018).

En la mayoría de los casos, se espera que un conjunto de datos de transacciones de crédito tenga más transacciones normales que fraudulentas. Por lo tanto, la precisión de un sistema de detección de fraude depende de la construcción de un modelo que pueda manejar adecuadamente dicho conjunto de datos desbalanceado. Una de las técnicas para reequilibrar el conjunto de datos es la Técnica de Sobremuestreo de Minorías Sintéticas (SMOTE). (Alshameri & Xia, 2023)

La Técnica de Sobremuestreo de Minorías Sintéticas (SMOTE) es una de las técnicas más dominantes utilizadas para abordar el problema del desequilibrio de clases que se encuentra en conjuntos de datos, como los utilizados para construir modelos de detección de fraude con tarjetas de crédito basados en aprendizaje automático (Alshameri & Xia, 2023). El método SMOTE genera muestras de una clase específica conectando un punto de datos con sus  $k$  vecinos más cercanos. Este método genera puntos de datos sintéticos que no son una réplica directa de la instancia de

la clase minoritaria. Esto se hace para evitar el fenómeno del sobreajuste durante el proceso de entrenamiento (Ileberi et al., 2021).

## VII. MARCO CONCEPTUAL

### 1. Inteligencia Artificial

En su sentido más amplio, es la inteligencia que exhiben las máquinas, particularmente los sistemas informáticos, a diferencia de la inteligencia natural de los seres vivos. Es un campo de investigación en informática que desarrolla y estudia métodos y software que permiten a las máquinas percibir su entorno y utilizar el aprendizaje y la inteligencia para tomar acciones que maximicen sus posibilidades de lograr objetivos definidos. Sheikh et al. (2023) definen todas las aplicaciones que hoy calificamos como IA y da margen para cambios futuros en esa calificación, "Sistemas que muestran un comportamiento inteligente analizando su entorno y tomando acciones -con cierto grado de autonomía- para lograr objetivos específicos".

### 2. Machine Learning

El Machine Learning puede considerarse un subconjunto de la IA que abarca dos enfoques predictivos principales, es decir, el aprendizaje supervisado y no supervisado. En el aprendizaje supervisado, los conjuntos de datos contienen etiquetas conocidas como objetivo de predicción, mientras que, en el aprendizaje no supervisado, los conjuntos de datos no tienen etiquetas.

### 3. Data Analysis

El análisis de datos es el proceso de inspeccionar, limpiar, transformar y modelar datos con el objetivo de descubrir información útil, informar conclusiones y apoyar la toma de decisiones (Brown, 2014). El análisis de datos tiene múltiples facetas y enfoques, abarcando diversas técnicas bajo una variedad de nombres, y se utiliza en diferentes dominios de negocios, ciencia y ciencias sociales.

### 4. Exploratory data analysis

El análisis exploratorio de datos (EDA, por sus siglas en inglés) es utilizado por los científicos de datos para analizar e investigar conjuntos de datos y resumir sus principales características, a menudo empleando métodos de visualización de datos. (What Is Exploratory Data Analysis? | IBM, 2021)

### 5. Feature Engineering

La ingeniería de características es el proceso de crear nuevas características o transformar características existentes para mejorar el rendimiento de un modelo de aprendizaje automático. Involucra la selección de información relevante a partir de datos brutos y su transformación en un formato que pueda ser fácilmente entendido por un modelo. El objetivo es mejorar la precisión del modelo proporcionando información más significativa y relevante. (What Is A Feature Engineering? | IBM, 2024)

### 6. Jupyter Notebooks

Un Notebook es un documento generado por la aplicación Jupyter Notebook, que se puede compartir y

que combina código informático, descripciones en lenguaje sencillo, datos, visualizaciones enriquecidas como modelos 3D, cuadros, gráficos y figuras, y controles interactivos. Un cuaderno, junto con un editor (como JupyterLab), proporciona un entorno interactivo rápido para crear prototipos y explicar código, explorar y visualizar datos y compartir ideas con otros se define (Project Jupyter Documentation — Jupyter Documentation 4.1.1 Alpha Documentation, s. f.) en su documentación.

### 7. Docker

Docker es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones. Docker brinda la capacidad de empaquetar y ejecutar una aplicación en un entorno poco aislado llamado contenedor. El aislamiento y la seguridad le permiten ejecutar muchos contenedores simultáneamente en un host determinado. Los contenedores son livianos y contienen todo lo necesario para ejecutar la aplicación, por lo que no es necesario depender de lo que está instalado en el host se define («Docker Overview», 2023) en su página de documentación.

## VIII. ARQUITECTURA LÓGICA DE LA SOLUCIÓN

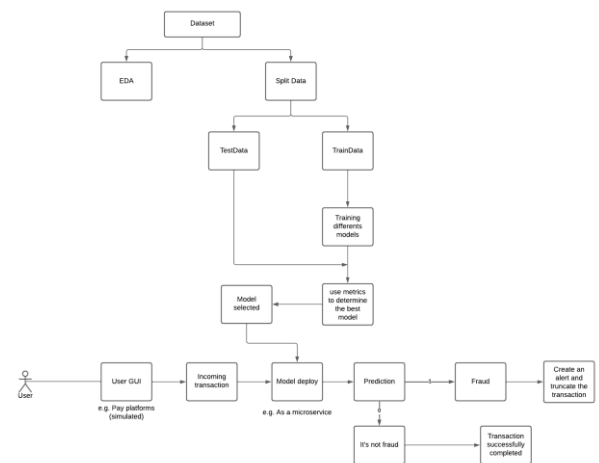


Fig. 3. Arquitectura Lógica de la Solución

## IX. ARQUITECTURA FÍSICA DE LA SOLUCIÓN

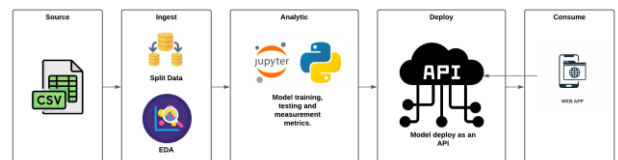


Fig. 4. Arquitectura Física de la Solución

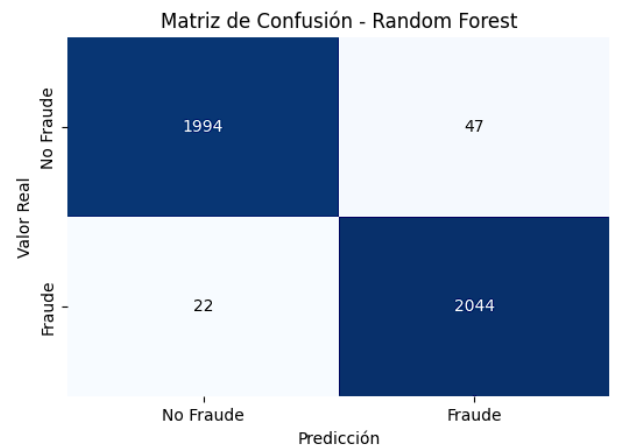
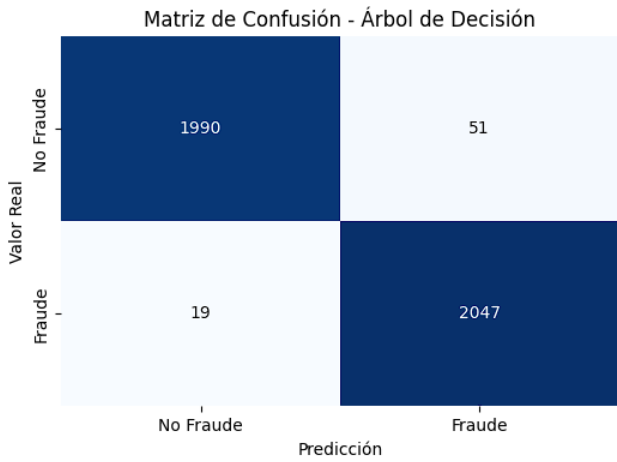
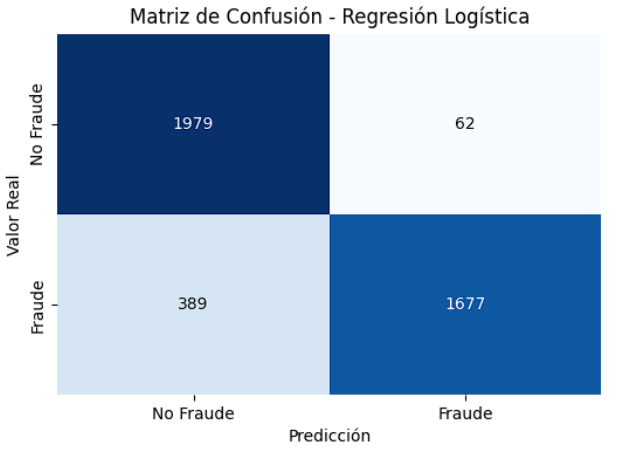
## X. PROTOTIPO

Para desarrollar un modelo eficiente en la predicción de transacciones fraudulentas, se llevó a cabo un exhaustivo trabajo de minería de datos para comprender la

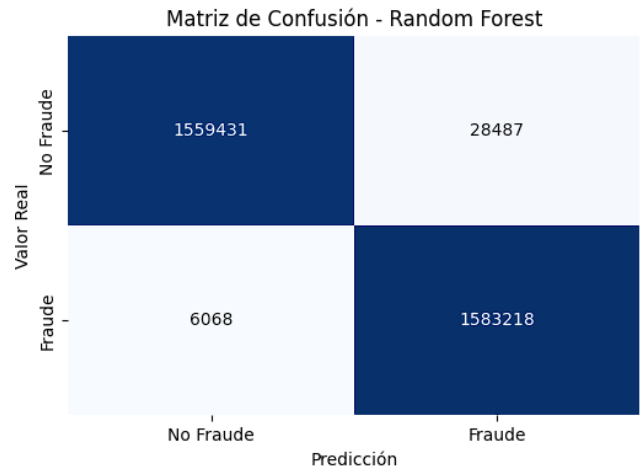
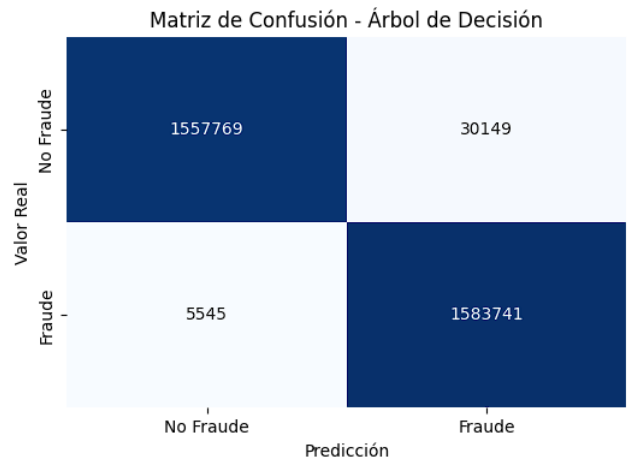
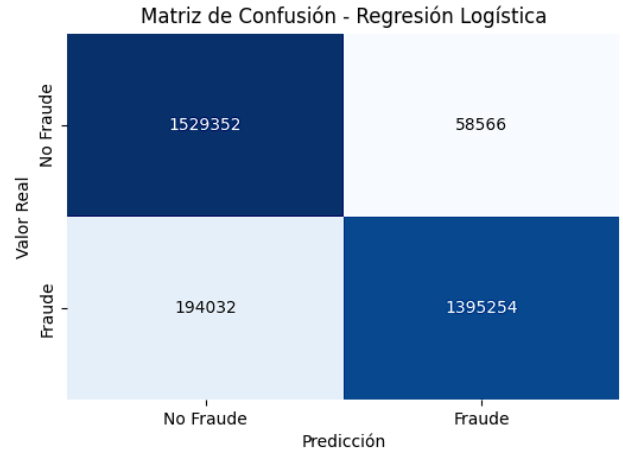
naturaleza de los datos y determinar qué técnicas de machine learning serían más efectivas para abordar este problema.

Se emplearon modelos de regresión logística, árboles de decisión y Random Forest, en combinación con dos técnicas para el manejo del desbalanceo de los datos: submuestreo aleatorio (Random Undersampling) y sobremuestreo sintético a través de la técnica SMOTE (Synthetic Minority Over-sampling Technique), Donde los mejores resultados se obtuvieron utilizando SMOTE, destacando especialmente el rendimiento de los árboles de decisión.

### 1. Métricas obtenidas utilizando Random Undersampling



### 2. Métricas obtenidas utilizando SMOTE



## XII. CONCLUSIÓN

A partir de la investigación realizada, basada en la revisión sistemática de la literatura y la metodología seleccionada (MLOps), diseñamos una arquitectura lógica y física para explicar la estructura de nuestro prototipo. Además, investigamos diversas herramientas que nos facilitarían una mejor fase de desarrollo, así como técnicas de análisis y predicción de datos que nos proporcionaron una mayor interpretación de los conjuntos de datos.

Pudimos observar cómo las diferentes características del conjunto de datos afectan la predicción realizada por el modelo. A través de esta observación, identificamos las variables más influyentes en la detección de fraudes y ajustamos nuestro modelo para mejorar su precisión y robustez.

El uso de técnicas como SMOTE fue crucial para abordar el desequilibrio de clases en los datos, lo que nos permitió mejorar significativamente el rendimiento del modelo sin caer en el sobreajuste. Esta técnica, junto con otras herramientas y enfoques implementados, contribuyó a la creación de un sistema de detección de fraudes más eficaz y confiable.

Concluimos que el análisis y la inclusión de datos con distribuciones sesgadas desempeñan un papel fundamental en la mejora de la claridad y precisión en el análisis del fraude bancario. Al reconocer y manejar adecuadamente estos datos sesgados, pudimos identificar patrones y tendencias que serían difíciles de detectar en un conjunto de datos balanceado. Este enfoque permitió una detección más precisa de comportamientos anómalos y fraudulentos, optimizando así la efectividad del sistema de detección de fraudes.

Después de evaluar los modelos, llegamos a la conclusión de que, aunque el problema es de clasificación binaria, la regresión logística no presenta un buen rendimiento en este conjunto de datos. Por otro lado, tanto el árbol de decisión (siendo este el de mejor performance) como el Random Forest muestran una precisión y sensibilidad superiores, métricas cruciales en la detección de fraude bancario. Basándonos en esto, podemos afirmar que el uso de técnicas de Machine Learning para la detección de fraudes en transacciones en línea es factible.

## REFERENCES

A Cost-sensitive weighted Random Forest Technique for Credit Card Fraud Detection. (2019, 1 de julio). IEEE Conference Publication | IEEE Xplore.  
<https://ieeexplore.ieee.org/document/8944885>

Abakarim, Y., Limam, M., & Attiou, A. (2018). An Efficient Real Time Model For Credit Card Fraud Detection Based On Deep Learning. ACM.  
<https://doi.org/10.1145/3289402.3289530>

Abdulalem, A., Shukor, A. R., Hajar, O. S., Elfadil, E. T., Arafat, A.-D., Maged, N., Tusneem, E., Hashim, E., & Abdu, S. (2022). Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. Applied Sciences, 12, 9637.  
<https://doi.org/10.3390/app12199637>

## 3. Entrenamiento de Modelos: Fragmento Clave

En este apartado se presenta un fragmento de código esencial para el entrenamiento de los modelos de machine learning del proyecto.

```
logreg_cv = LogisticRegression(solver='liblinear', random_state=123)
dt_cv = DecisionTreeClassifier(random_state=123, max_depth=8, min_samples_leaf=4, min_samples_split=2)
rf_cv = RandomForestClassifier(random_state=123, max_depth=8, min_samples_leaf=2, min_samples_split=5, n_estimators=100)
cv_dict = {0: 'Regresión Logística', 1: 'Árbol de Decisión', 2: 'Random Forest'}
cv_models = [Logreg_cv, dt_cv, rf_cv]

# Iterar sobre los modelos y calcular las métricas
for i, model in enumerate(cv_models):
    model.fit(X_train_, y_train_) # Entrenar el modelo
    y_pred_ = model.predict(X_test_) # Predecir con el conjunto de prueba
    report = classification_report(y_test_, y_pred_) # Calcular el reporte de clasificación
    print(f"--- {cv_dict[i]} ---")
    print(report)
    cri_ = classification_report_imbalanced(y_test_, y_pred_, target_names=['No Fraude', 'Fraude'])
    print(cri_)
    cm = confusion_matrix(y_test_, y_pred_) # Calcular la matriz de confusión
    # Crear un heatmap de la matriz de confusión
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False,
                xticklabels=['No Fraude', 'Fraude'], yticklabels=['No Fraude', 'Fraude'])
    plt.title(f"Matriz de Confusión - {cv_dict[i]}")
    plt.xlabel('Predicción')
    plt.ylabel('Valor Real')
    plt.show()
```

## XI. TABLA DE VALORACIÓN DEL PROTOTIPO

Características	Definición o descripción	1	2	3	4	5
Understandability	¿Fácil de comprender?					5
Documentation	¿Documentación de usuario completa, apropiada y bien estructurada?					5
Buildability	¿Fácil de construir en un sistema compatible? (Close-Open)					5
Installability	¿Fácil de instalar en un sistema compatible?					5
Learnability	¿Fácil de aprender a usar sus funciones?				4	
Identity	¿La identidad del proyecto / software es clara y única?				4	
Copyright	¿Es fácil ver quién posee el proyecto / software?				4	
Community	¿Evidencia de comunidad actual / futura?				4	
Testability	¿Fácil de probar la corrección de las funciones caja negra?			3		
	Modelo de evaluación basado en el estándar ISO 9126 ISO 15504 + ESCALA DE LIKERT					

- Ali, A., Razak, S. A., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., Elhassan, T., Elshafie, H., & Saif, A. (2022). Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. *Applied Sciences*, 12(19), 9637. <https://doi.org/10.3390/app12199637>
- Analysis on credit card fraud identification techniques based on KNN and outlier detection. (2017, 1 de febrero). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/7972424>
- Application of Classification Models on Credit Card Fraud Detection. (2007, 1 de junio). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/4280163>
- Bian, Y., Cheng, M., Yang, C., Yuan, Y., Li, Q., Zhao, J. L., & Liang, L. (s. f.). FINANCIAL FRAUD DETECTION: a NEW ENSEMBLE LEARNING APPROACH FOR IMBALANCED DATA. AIS Electronic Library (AISeL). <https://aisel.aisnet.org/pacis2016/315>
- Credit Card Fraud Prediction And Detection using Artificial Neural Network And Self-Organizing Maps. (2019, 1 de marzo). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/8819758>
- Cui, J., Yan, C., & Cheng, W. (2021). A Credible Individual Behavior Profiling Method for Online Payment Fraud Detection. *ACM*. <https://doi.org/10.1145/3456146.3456151>
- Detection. (2022, 6 de julio). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/9862930>
- Fraud Detection at Claims. (s.f.). Recuperado el 1 de enero de 2023, de <https://www.friss.com>
- Khormuji, M. K., Bazrafkan, M., Sharifian, M., Mirabedini, S. J., & Harounabadi, A. (2014, 6 de junio). Credit Card Fraud Detection with a Cascade Artificial Neural Network and Imperialist Competitive Algorithm. <https://www.ijcaonline.org/archives/volume96/number25/16947-6736/>
- Lima, R. A. F., & Pereira, A. C. M. (2012). Fraud detection in web transactions. *ACM*. <https://doi.org/10.1145/2382636.2382695>
- Orche, A. E., & Bahaj, M. (2019). Approach to use ontology based on electronic payment system and machine learning to prevent Fraud. *ACM*. <https://doi.org/10.1145/3320326.3320369>
- Qawqzeh, Y., & Ashraf, M. (2023). A Fraud Detection System Using Decision Trees Classification in An Online Transactions. *ACM*. <https://doi.org/10.1145/3587828.3587860>
- Rtayli, N., & Enneya, N. (2020). Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. *Journal Of Information Security And Applications*, 55, 102596. <https://doi.org/10.1016/j.jisa.2020.102596>
- Tax, N., Vries, K., Jong, M. d., Dosoula, N., Akker, B. D., Smith, J., Thoung, O., & Bernardi, L. (2021). Machine Learning for Fraud Detection in E-Commerce: A Research Agenda. *arXiv*, abs/2107.01979.
- Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2015). APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, 75, 38-48. <https://doi.org/10.1016/j.dss.2015.04.013>
- Wang, S., Liu, C., Gao, X., Qu, H., & Xu, W. (2017). Session-Based Fraud Detection in Online E-Commerce Transactions Using Recurrent Neural Networks. *En Lecture Notes in Computer Science* (pp. 241-252). [https://doi.org/10.1007/978-3-319-71273-4\\_20](https://doi.org/10.1007/978-3-319-71273-4_20)
- Sheikh, H. I., Prins, C., & Schrijvers, E. (2023). Mission AI. *En Research for policy*. <https://doi.org/10.1007/978-3-031-21448-6>
- Project Jupyter Documentation — Jupyter Documentation 4.1.1 alpha documentation. (s. f.). <https://docs.jupyter.org/en/latest/>
- «Docker overview». (2023, 22 noviembre). Docker Documentation. <https://docs.docker.com/get-started/overview/>
- Testi, M., Ballabio, M., Frontoni, E., Iannello, G., Moccia, S., Soda, P., & Vessio, G. (2022). MLOPs: A Taxonomy and a Methodology. *IEEE Access*, 10, 63606–63618. <https://doi.org/10.1109/access.2022.3181730>
- Higuchi, T. (2023, 11 mayo). MLOps Blog Series Part 1: The art of testing machine learning systems using MLOps. *Microsoft Azure Blog*. <https://azure.microsoft.com/en-us/blog/mlops-blog-series-part-1-the-art-of-testing-machine-learning-systems-using-mlops/>
- Kreuzberger, D., Kuehl, N., & Hirschl, S. (2023). Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access*, 11, 31866-31879. <https://doi.org/10.1109/access.2023.3262138>
- Hu, J., Hu, R., Wang, Z., Li, D., Wu, J., Ren, L., Zang, Y., Huang, Z., & Mei, W. (2023). Collaborative Fraud Detection: How Collaboration Impacts Fraud Detection. *En Proceedings of the 2021 1st International Conference on Control and Intelligent Robotics*. <https://doi.org/10.1145/3581783.3613780>
- Amarasinghe, T., Aponso, A., & Krishnarajah, N. (2018). Critical Analysis of Machine Learning Based Approaches

for Fraud Detection in Financial Transactions. En Proceedings of the 2021 1st International Conference on Control and Intelligent Robotics.

<https://doi.org/10.1145/3231884.3231894>

Li, Q., He, Y., Xu, C., Wu, F., Gao, J., & Li, Z. (2022). Dual-Augment Graph Neural Network for Fraud Detection. Proceedings Of The 31st ACM International Conference On Information & Knowledge Management. <https://doi.org/10.1145/3511808.3557586>

Alshameri, F., & Xia, R. (2023). Credit card fraud detection: an evaluation of SMOTE resampling and machine learning model performance. International Journal Of Business Intelligence And Data Mining, 23(1), 1-13. <https://doi.org/10.1504/ijbidm.2023.131791>

Elreedy, D., & Atiya, A. F. (2019). A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. Information Sciences, 505, 32-64. <https://doi.org/10.1016/j.ins.2019.07.070>

Ileberi, E., Yanxia, S., & Wang, Z. (2021). Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost. IEEE Access, 9, 165286-165294. <https://doi.org/10.1109/access.2021.3134330>

Brown, M. S. (2014). Transforming Unstructured Data into Useful Information. En Auerbach Publications eBooks (pp. 227-246). <https://doi.org/10.1201/b16666-14>

What is a feature engineering? | IBM. (2024, 24 enero). Recuperado 26 de mayo de 2024, de <https://www.ibm.com/topics/feature-engineering>

What is Exploratory Data Analysis? | IBM. (2021, 4 octubre). Recuperado 26 de mayo de 2024, de <https://www.ibm.com/topics/exploratory-data-analysis>