



Grupo 27

## Entrega 2

Jhon Farley Adarve Díaz

[j.adarve@uniandes.edu.co](mailto:j.adarve@uniandes.edu.co)

Julian Alberto Gomez Gomez

[j.gomez24@uniandes.edu.co](mailto:j.gomez24@uniandes.edu.co)

Giovany Andres Cardona Rojo

[g.cardonar@uniandes.edu.co](mailto:g.cardonar@uniandes.edu.co)

## Contenido

Resumen .....	3
Problema.....	3
Pregunta de negocio y alcance del proyecto. ....	3
Descripción de conjuntos de datos.....	4
Modelamiento.....	4
Evaluación y selección del modelo .....	5
Evaluación.....	5
Selección .....	5
Tablero .....	6
Descripción .....	6
Funcionalidades .....	6
Conclusiones.....	7
Repositorios y anexos .....	8
Enlace al repositorio GitHub .....	8
Fuente de los modelos.....	8
Fuente al tablero .....	8
Experimentación .....	8
Conexión máquina virtual EC2 para los modelos en MLFLOW .....	9
Tablero.....	10
Reporte de trabajo en equipo .....	11

# Identificación de transacciones financieras fraudulentas mediante técnicas de machine learning

## Resumen

### Problema

En nuestro proyecto, abordamos uno de los desafíos más críticos que enfrentan las instituciones financieras en la actualidad: la detección de fraude en transacciones bancarias. A medida que los sistemas financieros se digitalizan cada vez más, los métodos fraudulentos evolucionan en sofisticación, lo que hace necesario el desarrollo de sistemas de detección más robustos y adaptables.

Nos enfrentamos a un problema complejo caracterizado por:

- Datos altamente desbalanceados, donde las transacciones fraudulentas son eventos raros
- Necesidad de procesamiento en tiempo real o casi real
- Requerimiento de alta precisión para minimizar falsos positivos que podrían afectar a clientes legítimos
- Necesidad de adaptación continua a nuevos patrones de fraude

### Pregunta de negocio y alcance del proyecto.

Nuestra pregunta de negocio es ¿Cómo podemos desarrollar un sistema de detección de fraude que identifique eficazmente transacciones fraudulentas? Para abordarlo analíticamente estudiamos ¿Cuáles son los modelos de machine learning supervisados más efectivos para predecir transacciones fraudulentas, y qué variables (monto de la transacción, ubicación, categoría, etc.) son los predictores más importantes en la detección de fraude?

El alcance de nuestro proyecto incluye:

- Desarrollo de modelos de machine learning para clasificación de transacciones
- Implementación de técnicas de análisis de datos para identificación de patrones
- Evaluación de rendimiento usando métricas específicas para datos desbalanceados
- Validación del modelo en un contexto simulado de transacciones bancarias
- Visualización mediante un tablero de control

## Descripción de conjuntos de datos

El conjunto de datos utilizado en este proyecto proviene de BankSim, un simulador que genera datos basados en transacciones bancarias reales de un banco español, con un total de 594,643 registros. Este dataset incluye 587,443 transacciones normales y 7,200 transacciones fraudulentas, representando un 1.2% del total, y abarca un periodo simulado de 180 días (aproximadamente 6 meses). Las características del fraude simulado reflejan un escenario en el que los ladrones comprometen un promedio de tres tarjetas por ejecución, realizando alrededor de dos transacciones fraudulentas por día por tarjeta. Este enfoque simulado permite trabajar con datos estadísticamente representativos sin comprometer información confidencial, cumpliendo con las normativas de privacidad.

El conjunto incluye 9 variables predictoras y 1 variable objetivo. Entre estas, destacan variables temporales como el "Step", que representa el día desde el inicio de la simulación, y variables de identificación como "Customer", que permite rastrear el historial transaccional por cliente. Las variables geográficas, como "zipCodeOrigin" y "zipMerchant", capturan los códigos postales del origen y del comerciante, respectivamente, ayudando a identificar patrones espaciales. Variables demográficas como "Age" y "Gender" segmentan a los clientes por edad y tipo de cuenta, mientras que variables transaccionales como "Merchant", "Category" y "Amount" describen las características de cada operación. La variable objetivo, "Fraud", indica si una transacción es fraudulenta (1) o legítima (0), siendo esta última el foco del modelo de detección de fraude. Este dataset ofrece una representación robusta para analizar y modelar patrones de comportamiento transaccional y riesgos asociados al fraude.

## Modelamiento

Para el modelamiento se utilizó la metodología CRISP-DM, que permitió estructurar el proceso desde la comprensión del problema hasta la evaluación de los modelos. Se implementaron cinco modelos de machine learning supervisado: Regresión Logística, Random Forest, K-Nearest Neighbors (KNN) y XGBoost. Estos modelos fueron seleccionados por su capacidad para manejar datos desequilibrados, alta interpretabilidad y efectividad en la clasificación binaria, clave para la detección de fraude.

El proceso de entrenamiento incluyó la división del conjunto de datos en particiones de entrenamiento y prueba, el balanceo de las clases mediante SMOTE para abordar el desbalance en las transacciones fraudulentas, y la estandarización de variables numéricas. La selección de características incluyó variables clave como el monto, la categoría y la ubicación, consideradas relevantes para identificar patrones de fraude. Las métricas de evaluación seleccionadas fueron el F1-Score, el Recall y el AUC, priorizando la reducción de falsos negativos, ya que identificar fraudes de manera precisa es crítico para minimizar riesgos operativos y financieros.

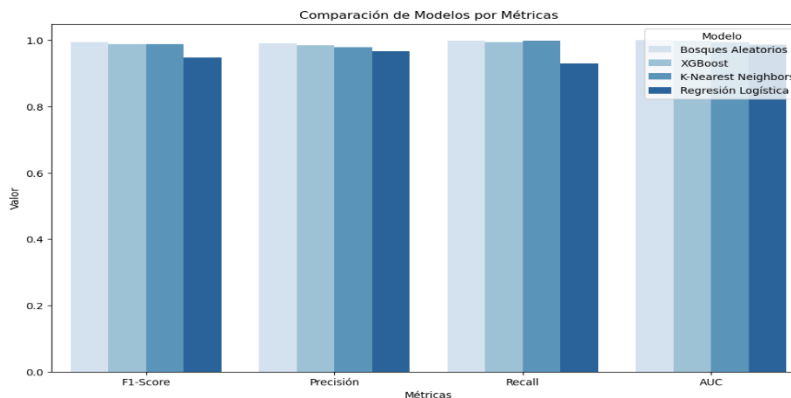
# Evaluación y selección del modelo

## Evaluación

### Modelos Evaluados

#	Modelo	Mejores Hiperparámetros	Precisión	Recall	F1-Score	AUC	Exactitud
1	Bosques Aleatorios	{'max_depth': None, 'n_estimators': 100}	0.991539	0.998551	0.995033	0.999728	0.995021
3	XGBoost	{'learning_rate': 0.1, 'n_estimators': 100}	0.985021	0.994086	0.989533	0.999201	0.989497
2	K-Nearest Neighbors	{'n_neighbors': 3, 'weights': 'uniform'}	0.978296	0.999386	0.988729	0.994062	0.988620
0	Regresión Logística	{'C': 1, 'solver': 'liblinear'}	0.966779	0.930194	0.948134	0.987098	0.949174

## Selección



El análisis comparativo de las métricas muestra que el modelo Bosques Aleatorios destaca frente a los demás por su alto desempeño en todas las métricas clave: precisión (0.9915), recall (0.9986), F1-Score (0.9950), AUC (0.9997) y exactitud (0.9950). Este modelo supera a XGBoost y K-Nearest Neighbors, que también presentan métricas sólidas pero ligeramente inferiores, especialmente en términos de F1-Score y AUC. Por otro lado, la Regresión Logística, aunque es un modelo más interpretable, presenta las métricas más bajas, con un recall de 0.9302 y un F1-Score de 0.9481, lo que la hace menos adecuada para tareas que priorizan el balance entre precisión y recall.

En este contexto, se justifica elegir Bosques Aleatorios como el modelo ideal debido a su excelente rendimiento general, especialmente en tareas críticas donde el recall y la exactitud son determinantes para minimizar errores. Además, su capacidad para manejar datos complejos y no lineales lo convierte en una opción robusta y confiable.

# Tablero

## Descripción

Nuestro tablero de control ha sido diseñado para ofrecer visualizaciones dinámicas e interactivas que faciliten el análisis y monitoreo de transacciones fraudulentas en tiempo real. Combina herramientas avanzadas como mapas de calor geográficos para identificar zonas de alto riesgo, gráficos de dispersión que relacionan montos y probabilidades de fraude, y series temporales que revelan patrones emergentes. Estas visualizaciones se integran con métricas clave de rendimiento del modelo, incluyendo tasa de detección de fraude (recall), precisión y F1-score, además de análisis demográficos detallados a través de gráficos de distribución.

La interfaz permite a los usuarios realizar análisis de profundidad (drill-down) mediante filtros interactivos por fecha, categoría de transacción, y rango de montos, agilizando la identificación de patrones sospechosos. Este diseño no solo mejora la usabilidad del sistema, sino que también optimiza la capacidad de respuesta frente a actividades fraudulentas.

El tablero permite a los equipos de monitoreo:

- Detectar nuevos patrones de fraude de manera proactiva.
- Evaluar la efectividad de las estrategias de prevención.
- Tomar decisiones informadas basadas en tendencias históricas y emergentes.
- Optimizar la asignación de recursos para investigar casos sospechosos.

Este enfoque integral no solo mejora la capacidad de detección de fraude, sino que también proporciona insights accionables para fortalecer medidas preventivas y minimizar pérdidas financieras asociadas con actividades fraudulentas.

## Funcionalidades

### 1. Métricas Clave (KPIs)

Indicadores de Rendimiento del Modelo:

- **Tasa de Detección de Fraude (Recall):** Porcentaje de fraudes detectados correctamente con tendencia histórica. Meta: >90%.
- **Precisión en Alertas:** Porcentaje de alertas verdaderas desglosadas por categoría de transacción. Meta: >85%.
- **F1-Score:** Equilibrio entre precisión y recall, con evolución temporal. Meta: >87%.

Indicadores Operativos:

- **Tiempo Promedio de Detección:** Desde la transacción hasta la alerta, categorizado por tipo de fraude.

## 2. Visualizaciones Principales

- **Mapa de Calor Geográfico:** Distribución de fraudes por código postal.
- **Gráfico de Dispersión Interactivo:** Relación entre monto y probabilidad de fraude.
- **Gráfica de Tendencia:** Visualización de tendencias de fraude por hora, día o mes.

## 3. Análisis de Variables Predictivas

- **Gráfico de Importancia de Variables:** Ranking de predictores más relevantes en la detección de fraude.
- **Matriz de Correlación:** Identificación de relaciones clave entre variables.

## 4. Segmentación y Perfiles

- **Gráficos de Distribución:** Análisis por edad, género, categoría de comercio y monto de transacción.
- **Filtros Interactivos:**
  - Rango de fechas.
  - Categoría de transacción.
  - Rango de montos.

Este tablero es una herramienta robusta y escalable que permite a las organizaciones maximizar la eficacia de sus estrategias de detección de fraude y fortalecer la toma de decisiones basada en datos.

## Conclusiones

El modelo de Bosques Aleatorios demostró ser la solución más robusta y confiable para la detección de transacciones fraudulentas. Su capacidad para equilibrar un alto recall (0.998551), fundamental para minimizar fraudes no detectados, con una precisión sobresaliente (0.991539) que reduce los falsos positivos, lo convierte en una herramienta ideal para este contexto. Además, su flexibilidad en la selección de hiperparámetros y su escalabilidad lo hacen adecuado para manejar grandes volúmenes de datos transaccionales, garantizando un rendimiento óptimo en escenarios complejos y dinámicos como los de detección de fraude.

Por su parte, el tablero desarrollado responde eficazmente a la pregunta de negocio planteada: ¿Cómo podemos desarrollar un sistema de detección de fraude que identifique eficazmente transacciones fraudulentas? Al integrar las predicciones generadas por el modelo de machine learning supervisados, el tablero permite monitorear y analizar las transacciones en tiempo real. A nivel analítico, aborda la pregunta secundaria identificando que los Bosques Aleatorios son el modelo más efectivo para predecir transacciones fraudulentas, destacando la importancia de variables como el monto de la transacción y la categoría de comercio como los predictores más relevantes.

Adicionalmente, la interfaz del tablero está diseñada para facilitar el uso por parte de los equipos de monitoreo, ofreciendo métricas clave (recall, precisión, F1-Score) y visualizaciones dinámicas como mapas de calor, gráficos de dispersión y tendencias temporales. Los filtros interactivos por rango de fechas, categoría y montos permiten realizar análisis detallados y personalizables. Este sistema no solo cumple con los objetivos del proyecto al identificar patrones sospechosos de manera eficaz, sino que también proporciona una herramienta escalable y fácil de usar para optimizar la prevención y mitigación del fraude financiero.

## Repositorios y anexos

### Enlace al repositorio GitHub:

<https://github.com/JhonAdarve/Fraud-Detection-in-Financial-Transactions/tree/master>

### Fuente de los modelos:

<https://github.com/JhonAdarve/Fraud-Detection-in-Financial-Transactions/blob/master/notebooks/Modeling.ipynb>

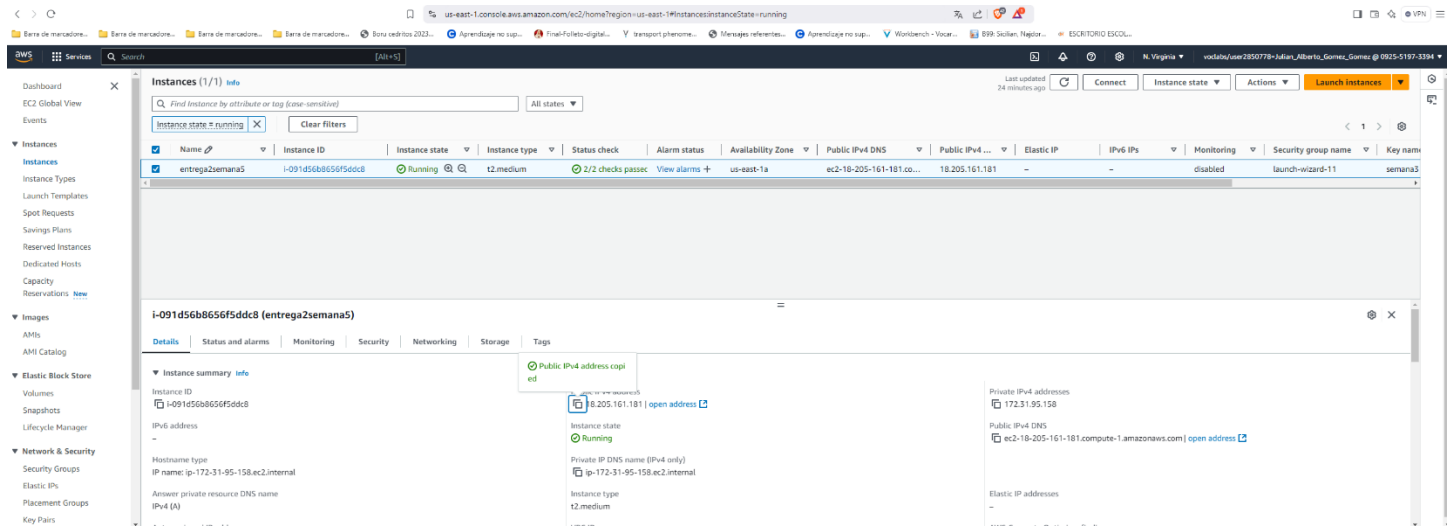
### Fuente al tablero:

[https://github.com/JhonAdarve/Fraud-Detection-in-Financial-Transactions/blob/master/dashboard/RadarFraud\\_Dashboard.py](https://github.com/JhonAdarve/Fraud-Detection-in-Financial-Transactions/blob/master/dashboard/RadarFraud_Dashboard.py)

### Experimentación:



## Conexión máquina virtual EC2 para los modelos en MLFLOW



The screenshot shows the AWS Management Console interface. On the left, there's a navigation menu with options like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The main area displays a list of EC2 instances. One instance, 'entrega2semana5', is highlighted. Below the list, the details for this instance are shown, including its ID (i-091d56b865f5ddc8), state (Running), type (t2.medium), and network configuration (Public IPv4 address 18.205.161.181). A tooltip is visible over the public IP address, showing additional details like the instance state (Running) and private IP DNS name (ip-172-31-95-158.ec2.internal).

```
(env-mlflow) ubuntu@ip-172-31-95-158:~$ sudo kill 1844
(env-mlflow) ubuntu@ip-172-31-95-158:~$ sudo lsof -i :8051
(env-mlflow) ubuntu@ip-172-31-95-158:~$ sudo lsof -i :8051
(env-mlflow) ubuntu@ip-172-31-95-158:~$ python3 Modelin_v1.py

/home/ubuntu/env-mlflow/lib/python3.12/site-packages/xgboost/core.py:158: UserWarning:

[16:53:09] WARNING: /workspace/src/learner.cc:740:
Parameters: { "use_label_encoder" } are not used.

Dash is running on http://0.0.0.0:8051/

* Serving Flask app 'Modelin_v1'
* Debug mode: on
/home/ubuntu/env-mlflow/lib/python3.12/site-packages/xgboost/core.py:158: UserWarning:

[16:54:10] WARNING: /workspace/src/learner.cc:740:
Parameters: { "use_label_encoder" } are not used.

186.155.14.225 -- [11/Nov/2024 16:54:40] code 400, message Bad request version ('~xÆ')
186.155.14.225 -- [11/Nov/2024 16:54:41] code 400, message Bad request version ("İ[$~µyâ'[^6q")
```

## Tablero



## Reporte de trabajo en equipo

1. **Jhon Adarve:** lideró la formulación de la pregunta de negocio, el alcance del proyecto y la definición del problema. También se enfocó en el desarrollo técnico del repositorio de Git, donde gestionó la versión inicial del código fuente del proyecto en general, desde los modelos hasta el tablero. Además, colaboró en la redacción de los informes técnicos y la documentación. Gestionará la presentación audiovisual del proyecto (narrativa y video).
2. **Julian Gomez:** fue responsable del desarrollo del tablero a su estado final, asegurándose de que cumpliera con los requisitos funcionales y visuales. Además, se encargará de la construcción del manual de usuario y de instalación de este.
3. **Giovanny Cardona:** se encargó de la implementación, experimentación y evaluación de modelos, utilizando herramientas como Python, AWS y MLflow para seleccionar el de mejor desempeño en la predicción del fraude, se encargará del despliegue de sus resultados a través del tablero en los servicios nube ya sea a nivel IaaS o PaaS, permitiendo la interacción del usuario con el modelo y la exploración de las visualizaciones.

Cada miembro del equipo trabajó en conjunto para cubrir los diferentes aspectos del proyecto, contribuyendo de manera efectiva al cumplimiento de los entregables.