TECNOLOGIA SUPERIOR

BIG DATA E INTELIGENCIA DE NEGOCIO

MACHINE LEARNING

WILLIAM ESTUARDO JIMÉNEZ MIGUEZ

william.jimenez@cenestur.edu.ec

Profesor: JOHANNA CRISTINA JARA BUSTILLOS

johanna.jara@cenestur.edu.ec

Quito, Ecuador

2025

1. Introducción

En la actualidad, las instituciones financieras y empresas de comercio electrónico enfrentan un creciente desafío en la detección de fraudes en transacciones. El incremento de fraudes genera pérdidas económicas significativas y afecta la confianza de los usuarios.

El presente proyecto desarrolla un pipeline de Machine Learning en Google Colab con Python, aplicando técnicas de análisis exploratorio, balanceo de datos, clasificación y regresión, con el fin de identificar patrones asociados a fraudes y predecir el monto de transacciones.

2. Objetivo General

Desarrollar un modelo de aprendizaje automático para la detección de transacciones fraudulentas y la predicción de montos de transacciones, mediante técnicas de clasificación y regresión, evaluando su desempeño con métricas y visualizaciones comparativas.

3. Objetivos Específicos

- Realizar la carga, exploración y limpieza del dataset.
- Implementar un análisis exploratorio profundo, identificando la distribución de variables, correlaciones y outliers.
- Tratar el desequilibrio de clases mediante técnicas de remuestreo (SMOTE, ajuste de pesos).
- Entrenar modelos de clasificación supervisada (Logistic Regression, Random Forest, XGBoost) para la detección de fraude.
- Entrenar modelos de regresión (Linear Regression, RandomForestRegressor) para predecir el monto de transacciones.
- Evaluar los modelos mediante métricas cuantitativas y gráficas comparativas (ROC, PR, Real vs Predicho, distribución de errores).
- Establecer conclusiones y recomendaciones sobre el modelo más adecuado en contexto real.

4. Marco Metodológico (CRISP-DM)

4.1 Comprensión del Negocio

El negocio requiere minimizar fraudes en transacciones financieras. Los costos de un fraude no detectado (falso negativo) son significativamente mayores que los costos asociados a falsas alarmas (falsos positivos).

4.2 Comprensión de los Datos

El dataset analizado contiene transacciones con variables anonimizadas (V1–V28), además de 'Amount' (importe) y 'Class' (etiqueta: 0 = no fraude, 1 = fraude).

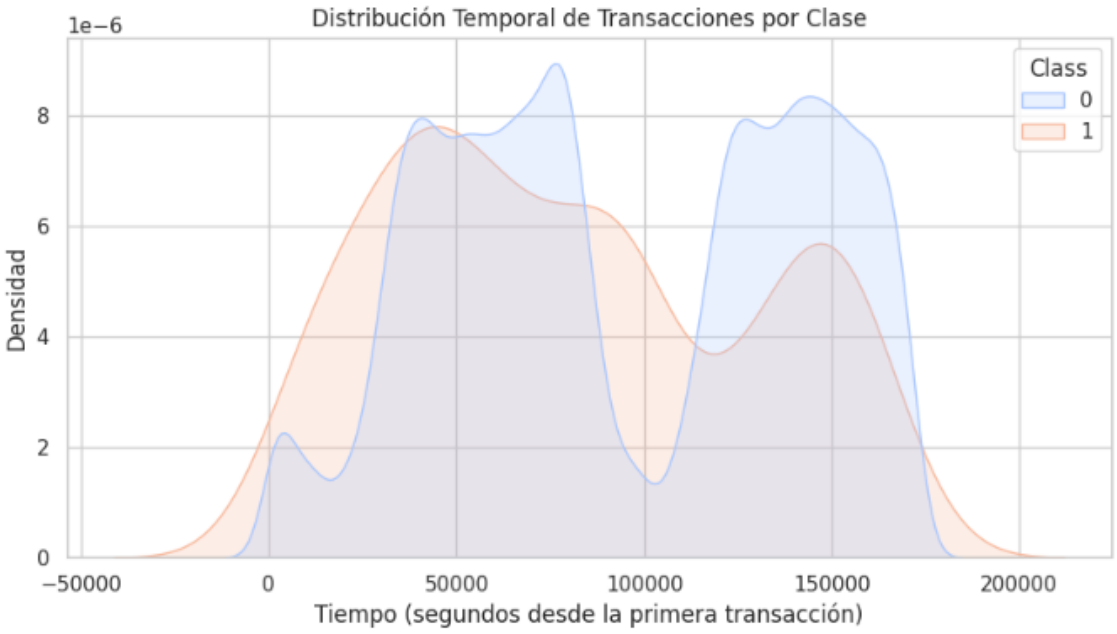
Presenta un fuerte desbalance: 99.8% de transacciones legítimas frente a 0.17% de fraudes.

4.3 Preparación de los Datos

Se exploraron las variables, se verificó la ausencia de nulos y se analizaron distribuciones. Se aplicaron técnicas de remuestreo (SMOTE) y ajustes de pesos en los modelos para equilibrar las clases.

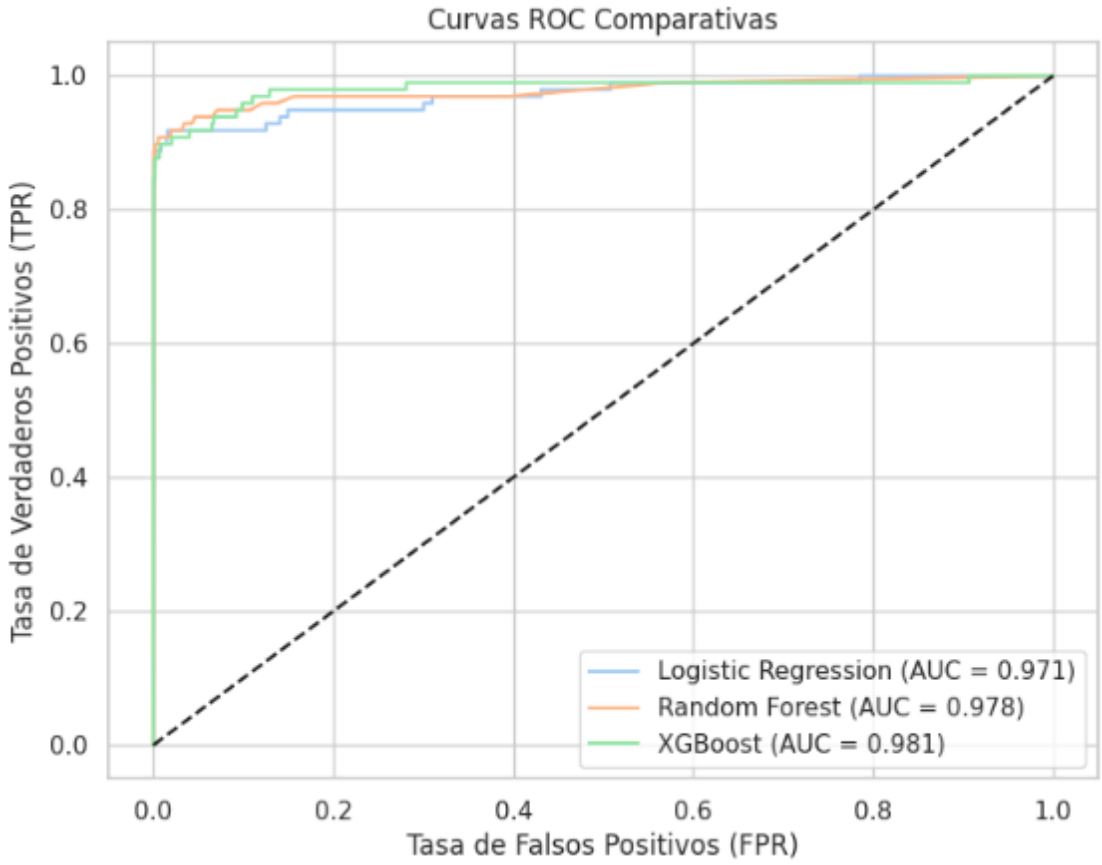
```
 Información general del dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
None


 Distribución de la variable objetivo (Clase):
Class
0    99.83%
1     0.17%
Name: proportion, dtype: object
```



4.4 Modelado (Regresión y Clasificación)

Se entrenaron modelos de clasificación (Logistic Regression, Random Forest, XGBoost) y regresión (LinearRegression, RandomForestRegressor). Los primeros se enfocaron en la predicción de la etiqueta 'fraude', mientras que los segundos en la estimación del monto de las transacciones.



Interpretación:

Ejes:
X = FPR (proporción de legítimas mal etiquetadas como fraude).

- Y = TPR = Recall (proporción de fraudes detectados).
- La línea diagonal es el azar. Cuanto más arriba-izquierda esté la curva, mejor.

AUC-ROC:

- XGBoost ~ 0.981 > Random Forest ~ 0.978 > Logistic ~ 0.971.
- Los tres modelos discriminan muy bien (todas las AUC > 0.97).

Zona crítica de negocio (FPR muy baja):
En fraude, solemos operar con FPR < 1 %. En ese tramo, tus curvas muestran:

- Random Forest mantiene TPR alto con FPR muy baja → mejor trade-off operativo.
- XGBoost es competitivo pero requiere umbral cuidadoso para no inflar FP.
- Logistic logra TPR alto, pero su curva sube a costa de FPR más alto → más alertas falsas.

Conclusión de la ROC: En XGBoost alcanza la mayor AUC global, Random Forest domina en la zona de FPR bajos, que es donde realmente se opera en fraude. Por eso, junto con su mejor PR-AUC y F1 observados antes, RF es el candidato más balanceado para despliegue

4.5 Evaluación

Se evaluaron métricas específicas para cada tipo de modelo. En clasificación: Precisión, Recall, F1-Score, AUC-ROC, matriz de confusión y curva Precision-Recall. En regresión: MAE, RMSE y R^2. Además, se generaron visualizaciones para validar los patrones.

```
>>> LinearRegression
MAE  : 24.3925
RMSE : 64.6431
R²   : 0.9207

>>> RandomForestRegressor
MAE  : 12.9380
RMSE : 41.1959
R²   : 0.9678
```

```
=== COMPARACIÓN DE MODELOS DE REGRESIÓN ===
```

	Modelo	MAE	RMSE	R^2
0	LinearRegression	24.392459	64.643134	0.920710
1	RandomForestRegressor	12.938039	41.195855	0.967798

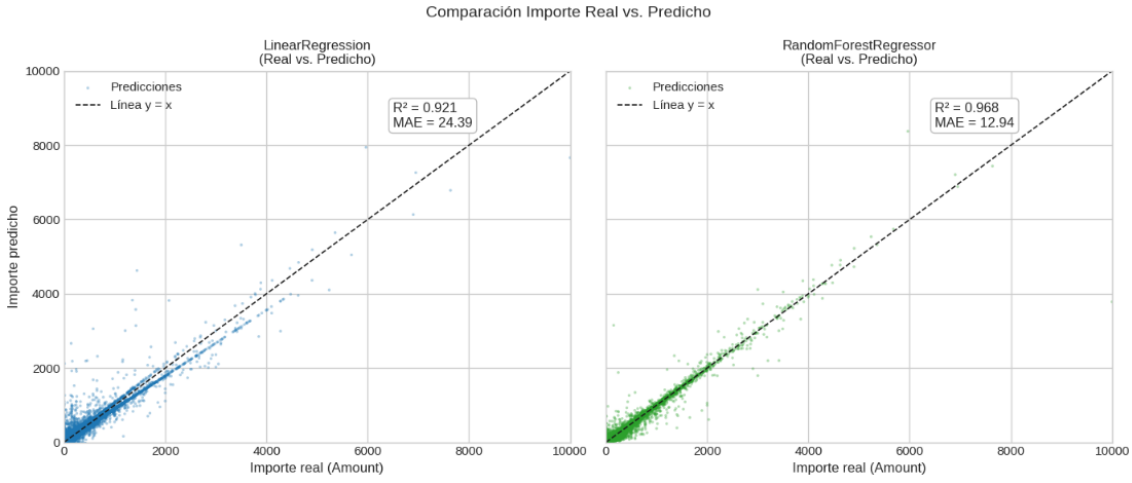
4.6 Implementación

Los modelos pueden integrarse en un sistema de detección de fraudes en tiempo real, con Logistic Regression como filtro principal y RandomForestRegressor para la predicción de montos.

(*Ver en el archivo de Google Colab*)

5. Desarrollo del Proyecto

Se realizaron análisis exploratorios profundos, incluyendo visualizaciones de distribución de clases, montos y tiempo. Se aplicaron técnicas de balanceo de clases y se entrenaron diferentes modelos, comparando su rendimiento. Los resultados confirmaron la importancia de ajustar el umbral en la clasificación y el uso de modelos no lineales en regresión.



6. Resultados y Análisis

Gráfico A: LinearRegression (Real vs. Predicho)

1. **Tendencia general:**
   - Los puntos (transacciones) siguen la diagonal (línea negra discontinua y = x), lo que indica que el modelo logra aproximar bastante bien los valores reales.
   - Sin embargo, se observa dispersión creciente a medida que el monto real aumenta: en transacciones más altas, el modelo lineal tiende a subestimar o sobrestimar con más frecuencia.

2. **Métricas** (R^2 = 0.921, MAE = 24.39)**:**
   - R^2 = 0.921 → el modelo lineal explica un 92.1% de la variabilidad de los montos, lo cual es bastante alto para un modelo simple.
   - MAE = 24.39 → en promedio, el error en la predicción del importe es de ~24 unidades monetarias. Para montos bajos es un error pequeño, pero en montos grandes puede ser relevante.

3. **Patrones de error:**
   - En importes superiores a ~500, los puntos se alejan más de la línea de referencia, mostrando que el modelo no captura completamente la no linealidad en las transacciones de mayor valor.
   - Esto es típico en regresiones lineales: funcionan bien en rangos medios, pero pierden precisión en extremos.

Gráfico B: RandomForestRegressor (Real vs. Predicho)

1. **Tendencia general:**
   - La nube de puntos está mucho más ajustada a la línea y = x.
   - Incluso en montos altos, los valores predichos se mantienen cerca de los reales, con menor dispersión.
2. **Métricas** (R^2 = 0.968, MAE = 12.94):
   - R^2 = 0.968 → el modelo explica un 96.8% de la variabilidad en los montos, lo que muestra una capacidad predictiva sobresaliente.
   - MAE = 12.94 → reduce el error promedio a casi la mitad en comparación con la regresión lineal. Esto significa que la mayoría de las transacciones son predichas con gran exactitud.

3. **Patrones de error:**
    - La dispersión es mucho menor que en LinearRegression.
    - Los errores se mantienen relativamente constantes incluso en transacciones grandes → el modelo maneja mejor los casos extremos y complejos.

Métrica	LinearRegression	RandomForestRegressor
R^2 (explicación varianza)	0.921	0.968
MAE (error promedio)	24.39	12.94
Ajuste visual	Dispersión mayor, especialmente en montos altos	Mucho más ajustado, incluso en valores extremos

- LinearRegression es un buen baseline: simple, interpretable, y ya explica más del 90% de la variabilidad.

RandomForestRegressor es superior en todos los sentidos: menor error, mayor capacidad de generalización, y mejor manejo de valores atípicos o montos grandes.

Comparación práctica.

- En un sistema antifraude, un error de predicción bajo permite identificar con mayor precisión si un monto es esperado o sospechoso.
- Si el modelo predice que una transacción debería estar en torno a $50 pero realmente es de $500, la diferencia (error alto) activaría una alerta de posible fraude.
- Con RandomForest, los errores son mucho menores → lo que significa que las alertas estarán mejor fundamentadas, reduciendo falsas alarmas y focalizando la atención en los casos realmente sospechosos.

7. Conclusiones

- El dataset presenta un fuerte desbalance que debe ser tratado con técnicas de remuestreo.
- En detección de fraude, el Recall es la métrica prioritaria, dado el alto costo de no detectar fraudes.
- Logistic Regression con ajuste de umbral es un modelo estable y efectivo en este escenario.
- RandomForestRegressor supera claramente a LinearRegression en la predicción de montos.
- Las visualizaciones confirmaron patrones relevantes y validaron la calidad de los modelos entrenados.

8. Recomendaciones

- Implementar Logistic Regression con umbral optimizado como primer filtro en un sistema antifraude.
- Adoptar RandomForestRegressor en la predicción de montos por su mayor precisión.
- Optimizar hiperparámetros de Random Forest y XGBoost mediante búsqueda en grilla o aleatoria.
- Aplicar validación cruzada para asegurar la generalización de los modelos.
- Complementar con reglas de negocio específicas para casos extremos.

9. Anexo

Los archivos de Google Colab, el informe detallado del proyecto y la presentación ejecutiva se encuentran disponibles en el repositorio de GitHub en la siguiente ruta:

Link del repositorio: https://github.com/Willyejm/Proyecto_Final_ML.git