Proyecto Final-Detección de fraude
Temas Selectos en Biomatemáticas

Brian Eduardo Fuentes Hérnandez[1]

[1] *Facultad de Ciencias, Universidad Nacional Autónoma de México, Ciudad de México C.P. 04510, Mexico**

(Dated: 13 de junio de 2025)

Este proyecto aborda la detección de transacciones fraudulentas mediante técnicas de aprendizaje automático, con un enfoque principal en la regresión logística. Se utilizaron dos conjuntos de datos distintos: uno proveniente de Kaggle y otro del repositorio de GitHub, con información sobre métodos de pago digitales. Cada conjunto fue sometido a un análisis exploratorio de datos, preprocesamiento (incluyendo normalización y codificación de variables categóricas) y estrategias para mitigar el desbalance de clases, como el submuestreo y el ajuste de pesos en el modelo. El modelo de regresión logística fue entrenado y evaluado en ambos casos, demostrando alta precisión cuando los datos fueron equilibrados y adecuadamente tratados.

I. INTRODUCCIÓN

La detección de fraude es un proceso esencial para proteger los sistemas financieros y comerciales contra actividades ilícitas que comprometen recursos, dinero o información confidencial. De acuerdo con IBM[1] , este proceso se basa en el uso de tecnologías avanzadas para identificar comportamientos sospechosos en tiempo real, tales como transacciones inusuales, accesos no autorizados o patrones anómalos que puedan representar un riesgo. En un entorno digital donde la automatización y las operaciones en línea son la norma, el fraude representa una amenaza constante, no solo por el impacto económico que conlleva —empresas pueden llegar a perder hasta el 5 por ciento de sus ingresos anuales, sino también por sus consecuencias legales, regulatorias y reputacionales.[1]

Para combatir esta amenaza de manera efectiva, las organizaciones han adoptado soluciones basadas en inteligencia artificial, análisis estadístico y aprendizaje automático. Estas herramientas permiten analizar grandes volúmenes de datos históricos, detectar desviaciones respecto a patrones normales y adaptarse dinámicamente a nuevas estrategias fraudulentas. En este contexto, modelos como la regresión logística y las redes neuronales se han consolidado como enfoques clave en la detección de fraude, debido a su capacidad para clasificar eventos, estimar probabilidades y detectar anomalías.

Un **modelo de regresión** es una herramienta estadística que permite investigar la relación matemática entre una variable Y (dependiente) y una o varias variables X (independientes). A través del análisis de datos observados, el modelo estima parámetros como el intercepto A y la pendiente B, representando el cambio esperado en Y ante variaciones en X, manteniendo las demás variables constantes (*ceteris paribus*) [2].

La forma clásica del modelo de regresión lineal simple es:

$$Y = A + BX + u$$

- Y: Variable dependiente (lo que se busca predecir).
- X: Variable independiente (predictora).
- A: Intercepto (valor de Y cuando $X = 0$).
- B: Pendiente (cambio en Y por unidad de X).
- u: Término de error (captura la variabilidad no explicada por el modelo).

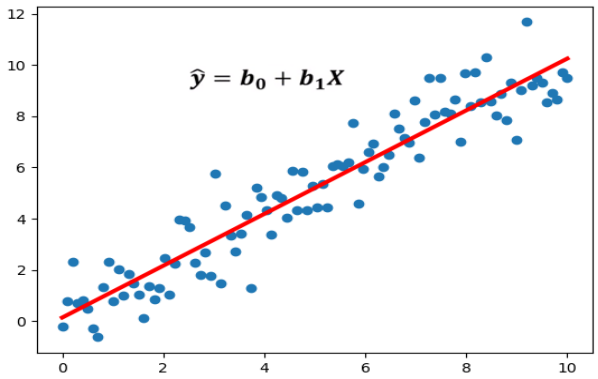

FIG. 1. Regresión lineal simple: $\hat{y} = b_0 + b_1 X$

La **regresión logística** es un algoritmo de aprendizaje supervisado utilizado para tareas de clasificación binaria, cuyo objetivo es predecir la probabilidad de que ocurra un evento entre dos posibles resultados (por ejemplo, fraude vs. no fraude). A diferencia de la regresión

* brianfuentes106@ciencias.unam.mx

lineal, que genera predicciones continuas, la regresión logística aplica la *función sigmoide* para convertir una combinación lineal de variables independientes en una probabilidad entre 0 y 1 [3].

Matemáticamente, la función principal está dada por:

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m)}}$$

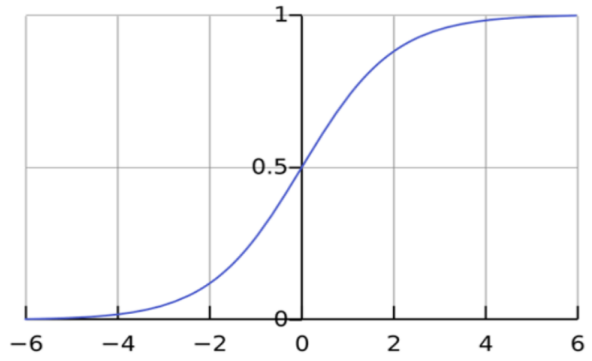

FIG. 2. Regresión logistica

II. OBJETIVOS

Implementar una solución integral basada en técnicas de Machine Learning, específicamente mediante el uso de regresión logística, con el propósito de analizar y detectar posibles casos de fraude en dos conjuntos de datos distintos.

III. DESARROLLO EXPERIMENTAL

La primera base de datos fue obtenida de `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data` y contiene un total de 284,807 transacciones con tarjeta de crédito realizadas en Europa. Cada registro está anonimizado y representado mediante variables transformadas mediante análisis de componentes principales (de `V1` a `V28`), junto con las variables originales `Time`, `Amount` y la variable objetivo `Class` (0 para transacción legítima y 1 para fraude).

No se detectaron valores nulos, lo que permitió trabajar directamente con los datos originales. Un primer análisis de la variable objetivo reveló un fuerte desbalance, con menos del 0.2 % de transacciones fraudulentas.

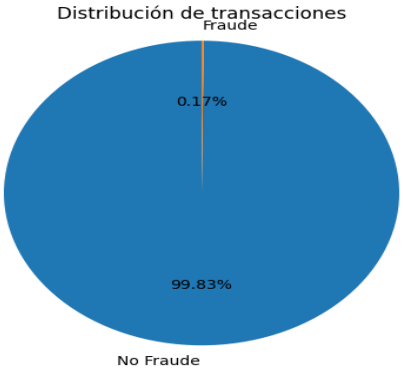

FIG. 3. Distribución de clases

También se analizaron otras variables relevantes:

- **Distribución del monto:** se generó un histograma del atributo `Amount`, observando una gran concentración de transacciones de bajo valor y algunos valores extremos.

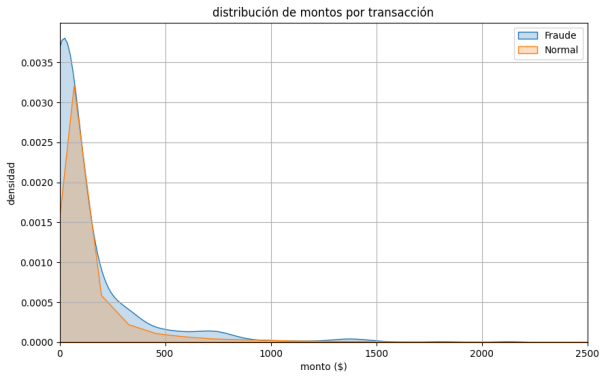

FIG. 4. Distribución de monto

- **Boxplots por clase:** se graficaron los montos para transacciones legítimas y fraudulentas, evidenciando que las transacciones fraudulentas tienden a concentrarse en ciertos rangos monetarios.

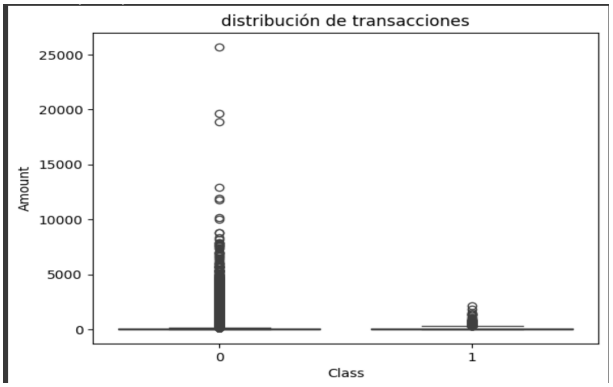

FIG. 5. Boxplot de monto

- **Correlación:** se generó una matriz de correlación entre las variables transformadas y `Class`, destacando algunas relaciones débiles pero informativas (como `V17` o `V14`).

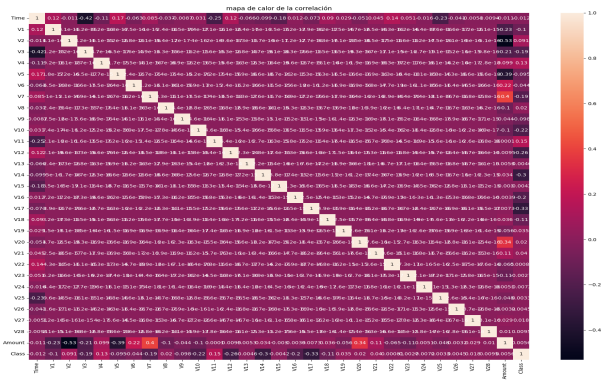

FIG. 6. mapa de calor de la correlación

- **Dispersión:** se analizó la relación entre la hora de la transacción) y el monto involucrado, diferenciando entre transacciones fraudulentas y normales. Como se observa, no hay un patrón claro que relacione el tiempo con la clase de transacción. Los fraudes ocurren a lo largo de todo el rango temporal, sin una concentración horaria específica.

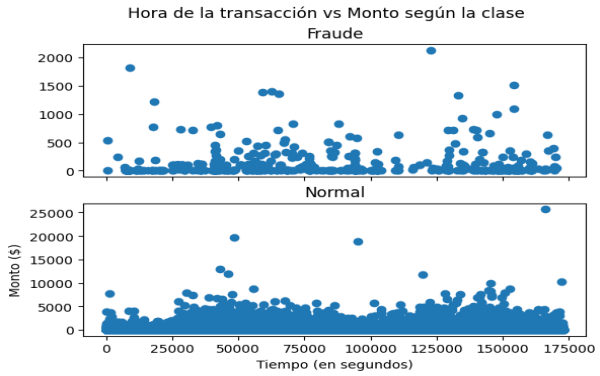

FIG. 7. Hora de la transacción vs. Monto según clase (Fraude vs Normal)

A. Base de datos 2

La segunda base de datos fue obtenida del repositorio en GitHub: `https://github.com/amankharwal/Website-data/blob/7ce2fbb5a5f9900cc0e28b7c34d29f4c202dc273/payment_fraud.csv`. Esta base contiene información relacionada con transacciones en línea, incluyendo atributos como:

- `accountAgeDays`: antigüedad de la cuenta.
- `numItems`: número de artículos comprados.
- `paymentMethod`: método de pago empleado (por ejemplo, `paypal`, `creditcard`, etc.).
- `paymentMethodAgeDays`: antigüedad del método de pago.
- `localTime`: hora local de la transacción.
- `label`: variable objetivo binaria (0 = transacción legítima, 1 = fraude).

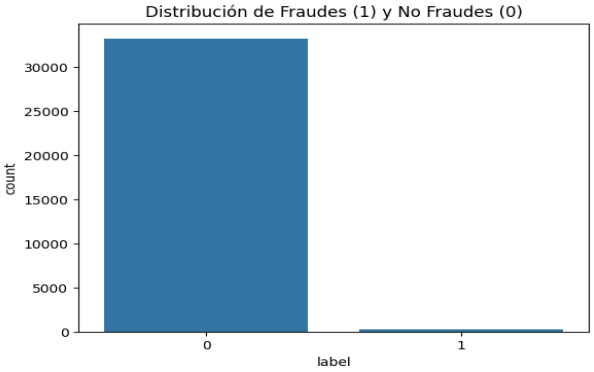

FIG. 8. Distribución de clases en la base de datos de métodos de pago

A partir del análisis visual se obtuvieron varios hallazgos relevantes:

- **Método de pago y fraude:** se observaron diferencias en la frecuencia de fraude según el método de pago utilizado.

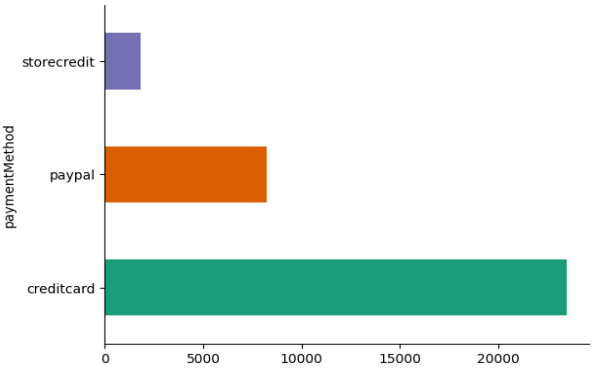

FIG. 9. Método de pago

- **Correlación:** si bien el número de variables es menor, se analizó la matriz de correlación entre las variables cuantitativas y la variable objetivo para identificar posibles relaciones útiles en el modelado.

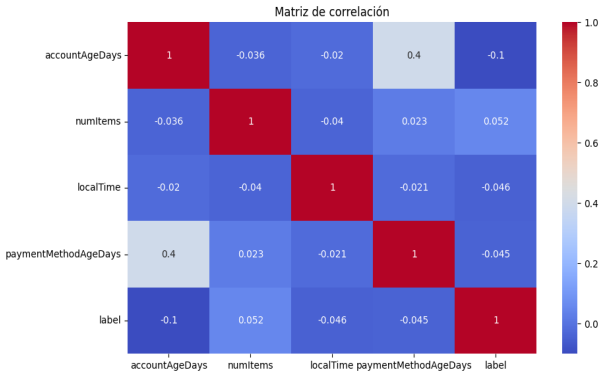

FIG. 10. Mapa de caor

Para la primer base de datos se procedió a realizar una verificación de valores nulos, confirmando que no existía ningún dato faltante.

Se utilizó el método `isnull().sum()` para verificar que todas las columnas estuvieran completas. Tambien se aplicó escalamiento mediante `MinMaxScaler` al conjunto de variables numéricas, con el fin de asegurar una magnitud comparable entre ellas y evitar sesgos durante el entrenamiento del modelo.

La segunda base de datos se le aplicó codificación tipo *dummy* para transformarlas en variables numéricas utilizables. Al igual que en la base anterior, se detectó un desbalance considerable en la variable `label`. Aunque en esta base tampoco se encontraron valores faltantes, se verificó mediante,de la misma manera que en el conjunto anterior, se utilizó `MinMaxScaler` para normalizar las columnas.

Una vez finalizada la preparación de datos, se procedió a implementar un modelo clasificación binaria: distinguir entre transacciones legítimas y transacciones fraudulentas. Para mantener la consistencia metodológica entre ambas bases de datos y facilitar la comparación, se seleccionó el modelo de **regresión logística** utilizando la biblioteca `scikit-learn`. Para entrenarlo, se dividieron los datos en conjuntos de entrenamiento y prueba utilizando la función `train_test_split` con una proporción del 80 % y 20 % respectivamente.

Debido al fuerte desbalance en ambas bases de datos (los fraudes representan una minoría), se tomaron medidas específicas para asegurar que el modelo no estuviera sesgado hacia la clase mayoritaria:

- En el primer conjunto (Kaggle),Utilizamos autocodificadores para el modelo de detección de fraude. Mediante autocodificadores, entrenamos la base de datos únicamente para aprender la representación de las transacciones no fraudulentas.

  La razón detrás de la aplicación de este método es que permitiera que el modelo aprendiera la mejor representación de los casos no fraudulentos para que puediera distinguir automáticamente los otros casos.

- En el segundo conjunto (Payment Fraud), también se construyó una versión balanceada mediante submuestreo aleatorio, permitiendo comparar el desempeño frente al modelo entrenado con datos desbalanceados.

- Además, se exploró el uso del parámetro `class_weight='balanced'` para que el modelo penalizara más los errores cometidos en la clase minoritaria.

El modelo fue evaluado usando las métricas apropiadas para clasificación binaria: **precisión**, **recall**, **F1-score** y **exactitud**. Se priorizó el *recall* para la clase positiva (fraude), ya que en este tipo de problemas es preferible cometer falsos positivos que falsos negativos (dejar pasar un fraude).

También se analizó la matriz de confusión para identificar el comportamiento del clasificador en ambos contextos.

IV. RESULTADOS Y DISCUSIÓN

A. Base de datos 1: creditcard.csv (Kaggle)

El resultado fue una buena capacidad de predicción general:

```
Reporte de clasificación:
              precision    recall  f1-score   support

         0.0       0.94      1.00      0.97       751
         1.0       1.00      0.59      0.74       122

    accuracy                           0.94       873
   macro avg       0.97      0.80      0.86       873
weighted avg       0.95      0.94      0.94       873


Precisión del modelo (accuracy): 0.9427262313860252
```

FIG. 11. Reporte de clasificación para creditcard.csv

Se obtuvo una precisión general (`accuracy`) del 94.1 %, con un F1-score superior a 0.74 para ambas clases. Esto indica un rendimiento adecuado y un buen equilibrio entre sensibilidad y especificidad. El uso de datos balanceados ayudó a evitar que el modelo se sesgara completamente hacia la clase mayoritaria (no fraude).

B. Base de datos 2: payment_fraud.csv

Para este conjunto, se realizó primero una evaluación con los datos desbalanceados. Aunque el modelo arrojó

una precisión del 99 %, el rendimiento en la clase de fraude fue prácticamente nulo, tal como se muestra en la siguiente captura:

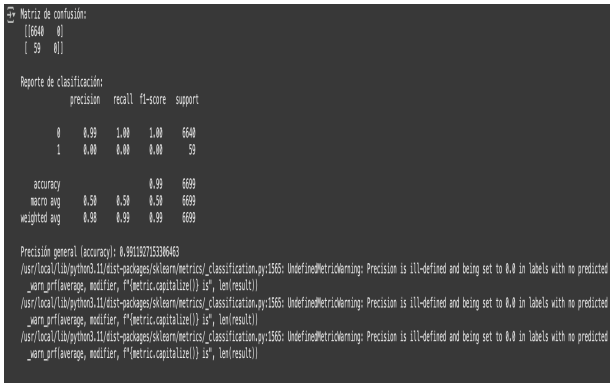

FIG. 12. Reporte de clasificación con datos desbalanceados

Posteriormente, se aplicó submuestreo de la clase mayoritaria y se reentrenó el modelo con la opción `class_weight='balanced'`. El resultado fue una mejora sustancial en la detección de fraudes, aunque con una ligera caída en la precisión general.

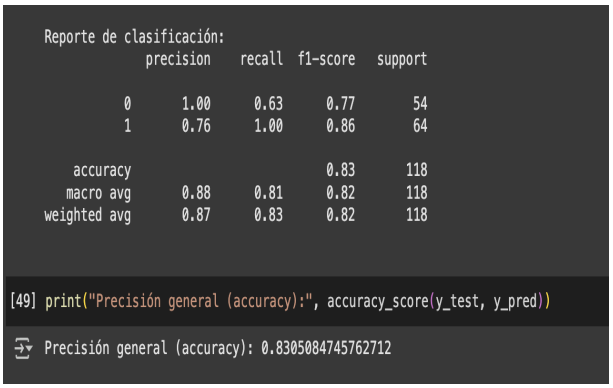

FIG. 13. Reporte de clasificación con datos balanceados

En términos generales, el modelo desarrollado presenta un rendimiento satisfactorio considerando la simplicidad del algoritmo y la complejidad del problema. En trabajos futuros, podrían explorarse con redes neuronales o modelos basados en árboles de decisión.

V. CONCLUSIONES

El algoritmo de regresión logística fue seleccionado por su capacidad interpretativa y demostró ser una herramienta eficaz.

En ambos conjuntos de datos, se evidenció que el desbalance de clases representa un desafío crítico, ya que puede llevar a modelos con alta precisión aparente pero sin capacidad real de detección de fraudes. Al aplicar submuestreo de la clase mayoritaria y ajustar los pesos del modelo, se logró mejorar el *recall* y el *F1-score* de la clase positiva (fraude), sin comprometer gravemente la precisión general.

Los resultados obtenidos permiten concluir que, incluso con modelos lineales como la regresión logística, es posible alcanzar un rendimiento aceptable en la detección de anomalías, siempre que se atienda cuidadosamente la estructura del conjunto de datos.

Finalmente, integrar este modelo dentro de un flujo de trabajo automatizado con actualización en tiempo real representaría un gran desafío y proyecto si es que aun no existe.

[1] IBM Corporation, "¿qué es la detección de fraude?" `https://www.ibm.com/mx-es/topics/fraud-detection` (2024), consultado el 7 de junio de 2025.

[2] Economipedia, "Modelo de regresión - qué es, definición y concepto," `https://economipedia.com/definiciones/modelo-de-regresion.html` (2024), consultado el 7 de junio de 2025.

[3] IBM Corporation, "¿qué es la regresión logística?" `https://www.ibm.com/mx-es/think/topics/logistic-regression` (2025), consultado el 13 de junio de 2025.

A. Anexo

El presente proyecto fue desarrollado tomando como referencia y guía técnica dos proyectos previos disponibles públicamente, que sirvieron como apoyo para comprender el enfoque de modelado y preprocesamiento más adecuado frente al problema de detección de fraude en transacciones financieras.

Base de datos 1: Kaggle - Credit Card Fraud Detection

Para la primera base de datos, se tomó como inspiración el notebook desarrollado por Khwaish Saxena, disponible en la plataforma Kaggle bajo el título *Credit Card Fraud Detection*. Este recurso proporcionó una visión clara sobre el uso de regresión logística, estrategias de muestreo para clases desbalanceadas, normalización y análisis exploratorio básico. La guía fue particularmente útil en la organización de la secuencia de pasos y en la interpretación de métricas como el *recall* y el *F1-score* para la clase positiva (fraude).

- Enlace del proyecto: `https://www.kaggle.com/code/khwaishsaxena/credit-card-fraud-detection#Model-Evaluation`

Base de datos 2: Blog Aman Kharwal - Fraud Detection

En el caso del segundo conjunto de datos, se siguió el enfoque publicado por Aman Kharwal en su blog personal. Su publicación fue útil para entender el comportamiento de variables categóricas como `paymentMethod`, la importancia del balanceo en el conjunto y el uso de regresión logística como modelo base. Además, la forma en que se estructuró la codificación de variables y las gráficas de distribución guiaron varios de los pasos de este proyecto.

- Enlace del proyecto: `https://amanxai.com/2020/08/04/fraud-detection-with-machine-learning/`