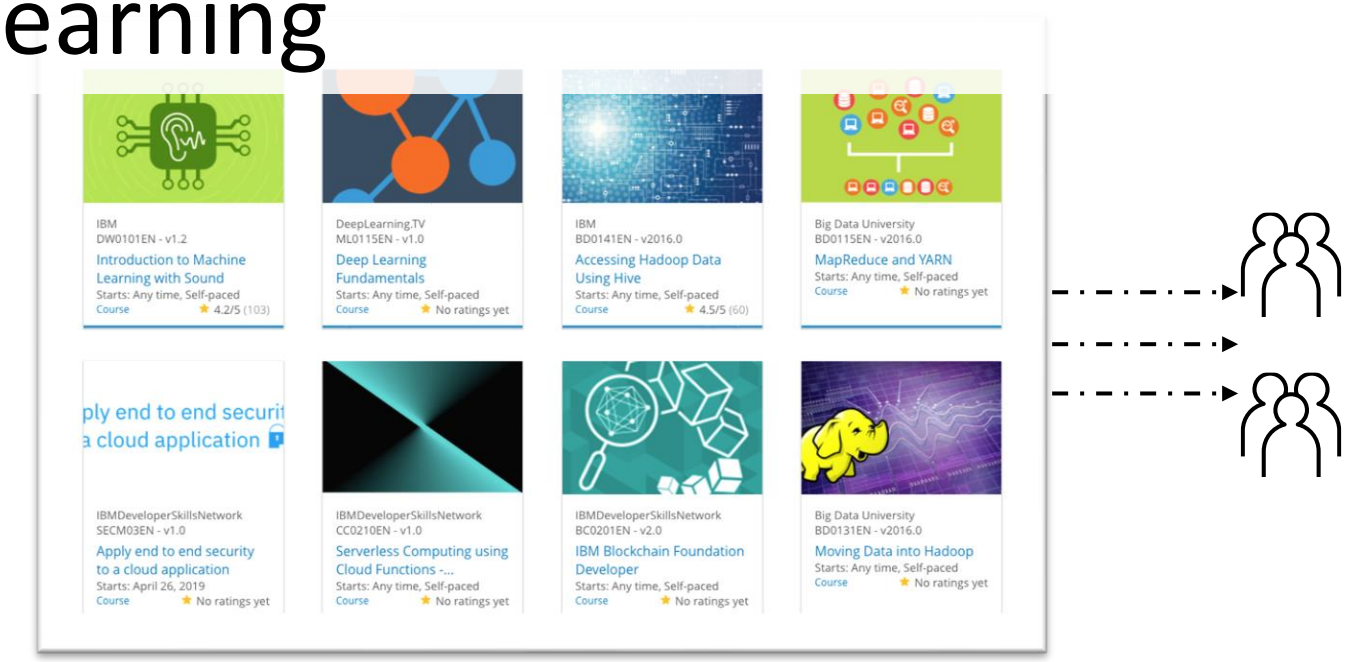
Detección de Fraude en Tarjetas de Crédito utilizando Machine Learning

Magali J. Cazella Mendez
14/10/2024



Outline

- Introduction and Background
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Conclusion
- Appendix

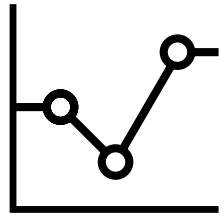
Introducción

Background del Proyecto:

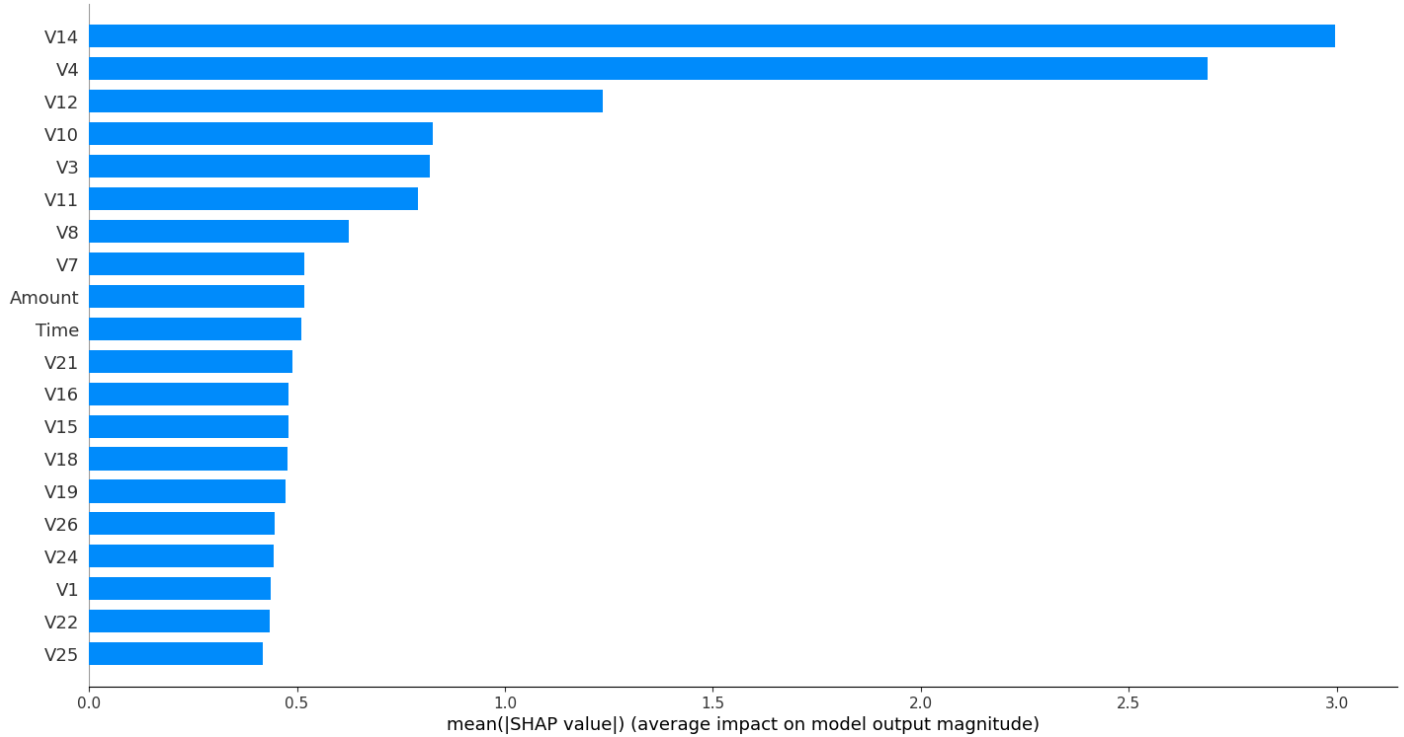
- Este proyecto está basado en el análisis de un **dataset de detección de fraudes en tarjetas de crédito** proveniente de **OpenML** (ID: 1597). El fraude en transacciones con tarjetas de crédito es un problema mundial que afecta tanto a los consumidores como a las instituciones financieras, causando grandes pérdidas económicas. Detectar patrones de fraude en tiempo real es crucial para mitigar estos impactos.
- **Contexto del dataset:**
- **Origen de los datos**: Transacciones realizadas por titulares de tarjetas de crédito europeos en septiembre de 2013.
- **Número de transacciones**: 284,807 transacciones.
- **Transacciones fraudulentas**: 492 transacciones marcadas como fraudulentas (0.172% del total, dataset desbalanceado).
- **Características del dataset**: La mayoría de las variables son transformaciones **PCA** para preservar la confidencialidad de los datos. Las únicas variables sin transformar son **'Time'** y **'Amount'**.

Problem Statement:
El principal desafío de este proyecto es **detectar transacciones fraudulentas** en tiempo real, utilizando un **dataset altamente desbalanceado**. Las técnicas de machine learning serán aplicadas para **distinguir entre transacciones legítimas y fraudulentas**, basándonos en patrones extraídos de los datos.
Retos clave:
•**Dataset desbalanceado**: Solo 0.172% de las transacciones son fraudulentas.
•**Alto volumen de datos**: Más de 284,000 transacciones.
•**Detección de anomalías**: Identificar patrones de fraude a partir de datos anónimos.
3. Hipótesis planteadas:
1.Hipótesis 1: Las transacciones fraudulentas tienden a mostrar patrones específicos en variables como el monto y el tiempo, diferenciándose de las transacciones legítimas.
2.Hipótesis 2: Los algoritmos de **machine learning** como **CNN** y **LSTM** pueden aprender estos patrones ocultos y mejorar significativamente la **detección de fraudes** en comparación con métodos tradicionales de reglas o modelos lineales.
3.Hipótesis 3: El uso de **modelos avanzados de machine learning** como **redes neuronales profundas (DNN)**, **CNN** y **RNN/LSTM** permitirá manejar mejor la naturaleza secuencial de los datos y mejorar la **precisión en la detección de transacciones fraudulentas**.

Exploratory Data Analysis



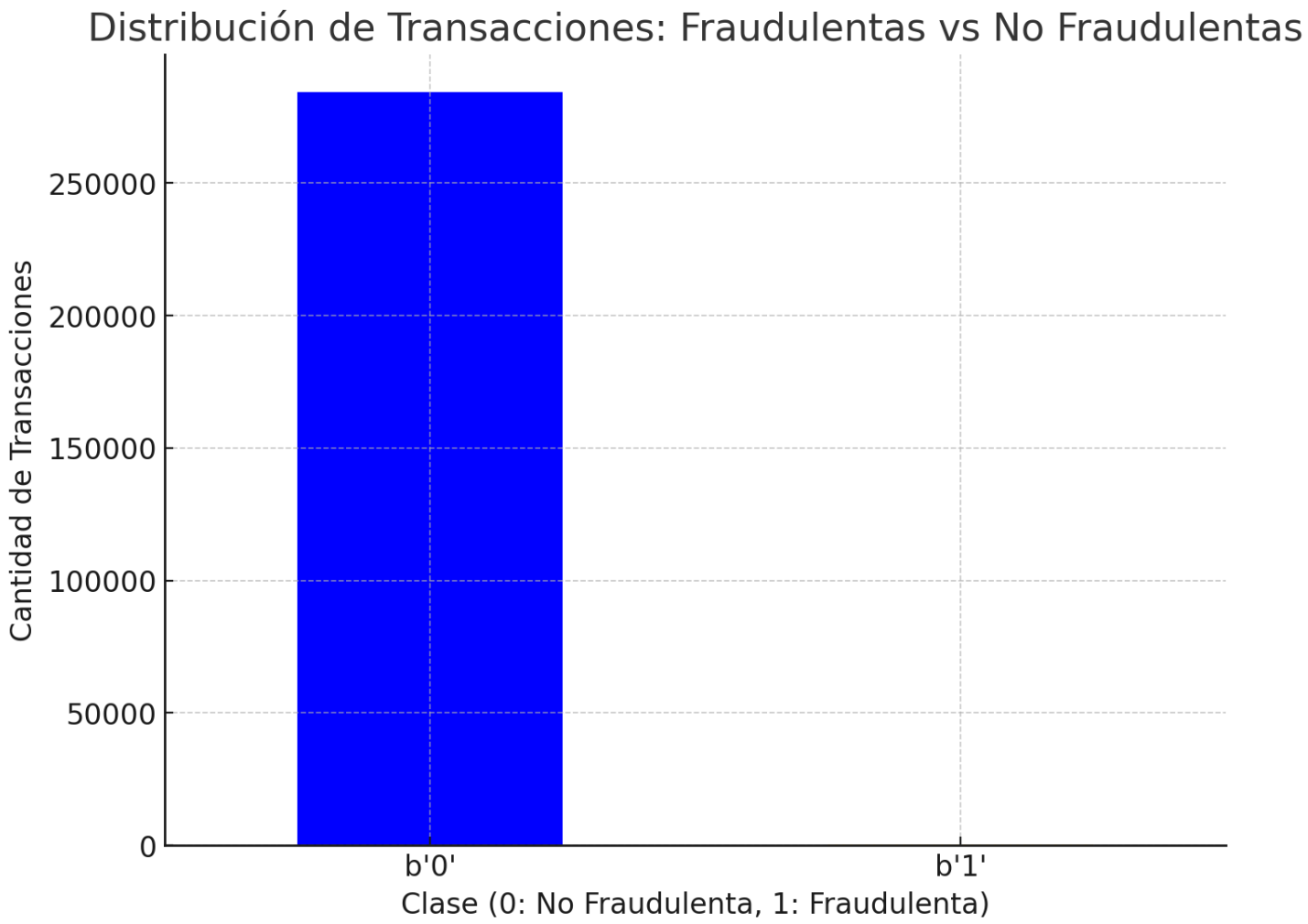
Distribución de transacciones por características



- Este gráfico muestra el impacto promedio que tienen las características sobre las predicciones del modelo. Cuanto más alto es el valor medio de SHAP, mayor es la influencia de esa variable en la predicción de si una transacción es fraudulenta o no.Las variables V14, V4 y V12 son las que tienen mayor impacto en el modelo, lo que sugiere que son las más relevantes para la detección de fraudes.
- Utilizando valores SHAP, podemos obtener una visión clara de qué variables contribuyen más al modelo de detección de fraude. Esta información es crucial para optimizar el rendimiento del modelo y mejorar su interpretabilidad.

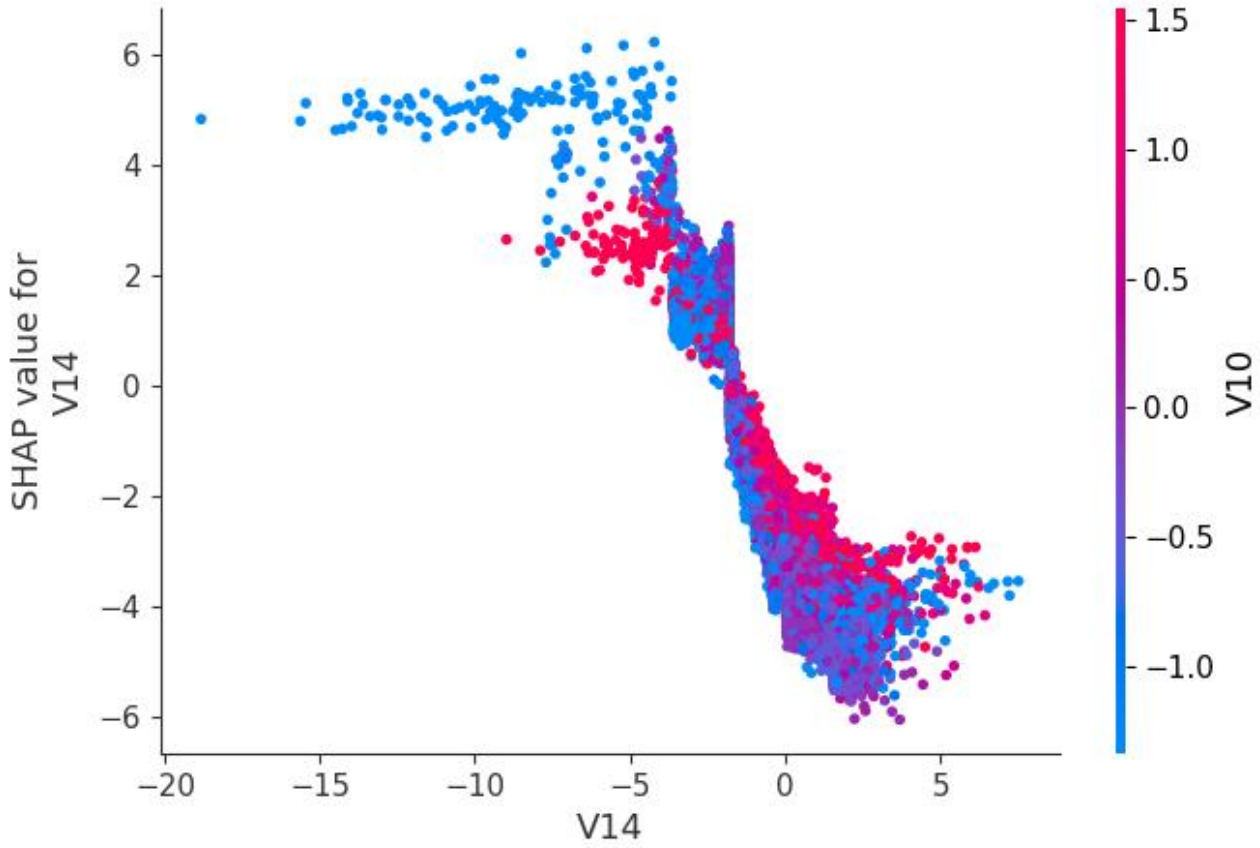
Transaction Fraud Distribution

- Este gráfico muestra la distribución de transacciones fraudulentas frente a las no fraudulentas en el dataset. Como se observa, la cantidad de transacciones no fraudulentas (Clase 0) es mucho mayor en comparación con las fraudulentas (Clase 1), lo que resalta el desbalance en el dataset. Este desbalance puede ser crítico al entrenar modelos de machine learning, ya que es importante implementar estrategias para manejarlo adecuadamente, como la asignación de pesos a las clases o la aplicación de técnicas de balanceo de datos.

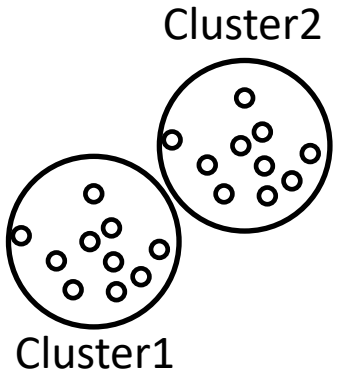


SHAP Dependence Plot para la característica V14

- El gráfico de dependencia SHAP para la característica V14 nos permite ver cómo los valores de esta característica afectan el valor SHAP (impacto en la predicción) para cada instancia. Las áreas más altas del gráfico indican que, para valores bajos de V14, es más probable que la transacción sea detectada como fraudulenta, mientras que los valores más altos de V14 tienden a no ser fraudes. La coloración (basada en V15) añade un contexto adicional sobre cómo otras características interactúan con V14 para influir en la predicción. En este caso, podemos observar que V15 también tiene un rol significativo en cómo V14 afecta el resultado final del modelo. **Interpretación:** Este tipo de análisis nos ayuda a entender cómo el modelo está utilizando las características para tomar decisiones, lo cual es crítico para asegurar la transparencia y la efectividad en la detección de fraudes.



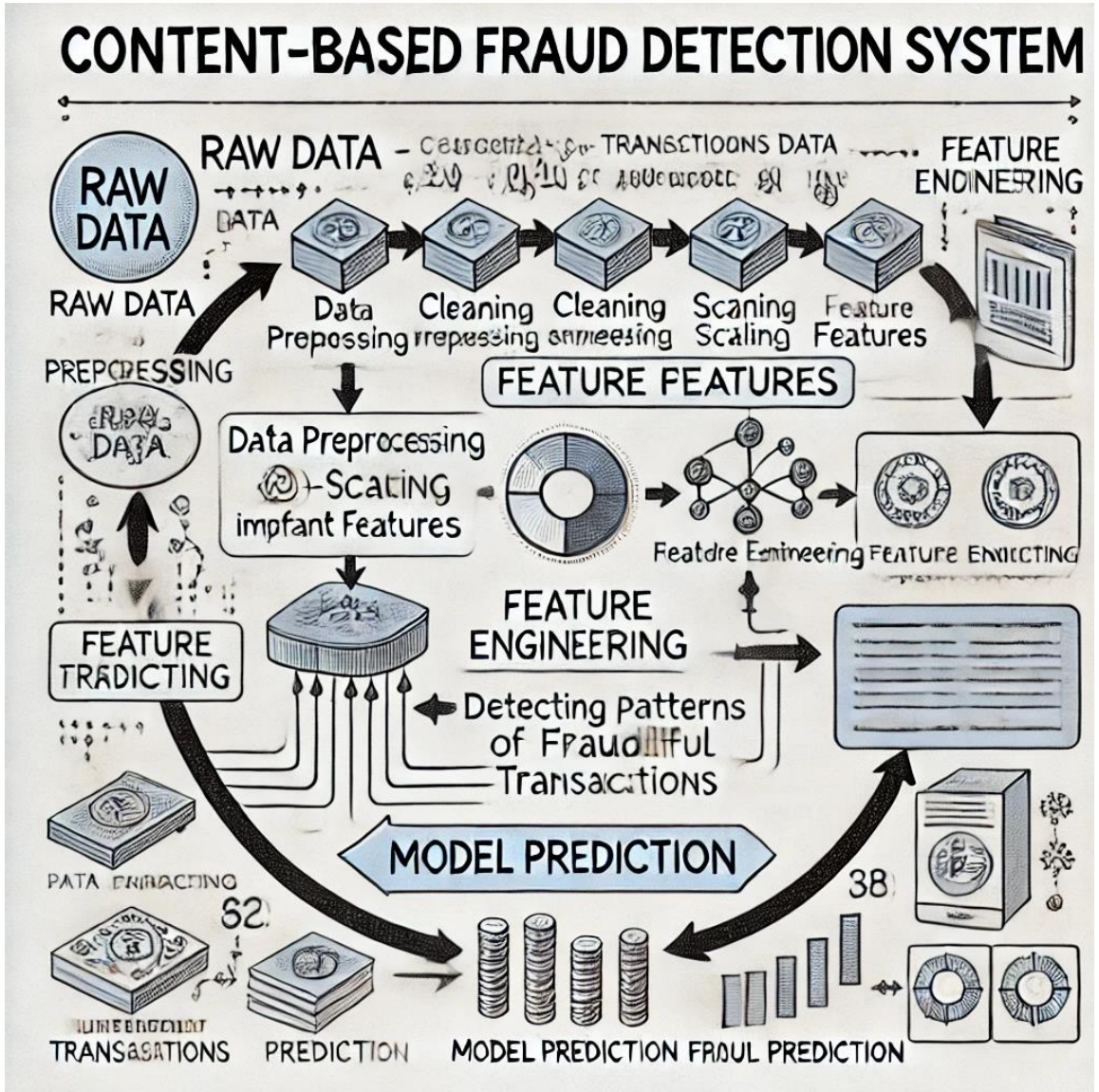
Content-based Recommender System using Unsupervised Learning



Flowchart of Content-Based Fraud Detection System

El diagrama de flujo muestra el pipeline completo de procesamiento de las transacciones. Comienza con los datos sin procesar, que pasan por un proceso de limpieza y selección de características. Luego, estas características se introducen en el modelo entrenado que analiza los patrones de fraude y genera predicciones sobre la probabilidad de que una transacción sea fraudulenta:

Raw Data: Se refiere a los datos transaccionales sin procesar.
Data Preprocessing: Incluye la limpieza de los datos, la eliminación de outliers y la normalización de las características.
Feature Engineering: Selección de las características más relevantes para el modelo.
Model Prediction: El sistema utiliza modelos entrenados para detectar patrones en las transacciones que indican fraude.
Fraud Prediction: El modelo proporciona predicciones sobre si las transacciones son fraudulentas o no, basándose en los patrones identificados.



Comparación de Resultados: LSTM vs CNN en Detección de Fraudes

Hyperparámetros: En el modelo LSTM o CNN para la detección de fraude, relacionados con la precisión de las recomendaciones de transacciones fraudulentas.Ejemplo de Configuración de Hyperparámetros:Threshold de Clasificación: 0.5 (predicción como fraude si la probabilidad es mayor que 0.5).Dropout Rate: 0.3 (para evitar overfitting en las capas).Learning Rate: 0.001 (en optimizadores Adam).Batch Size: 32 (número de muestras antes de actualizar los pesos del modelo).Número de Epochs: 50.

Número Promedio de Fraudes Detectados (en conjunto de prueba):Usando el modelo LSTM, en el conjunto de prueba has detectado los siguientes fraudes:

Número total de fraudes detectados: 74 fraudes detectados correctamente entre 98 fraudes reales.Promedio de fraudes detectados por usuario: Dado que el conjunto de prueba tiene 56,962 transacciones, con 98 fraudes, el promedio de fraudes detectados correctamente es:

Promedio de fraudes detectados=7498≈0.76 fraudes por usuario Promedio de fraudes detectados= 9874 ≈0.76fraudes por usuario

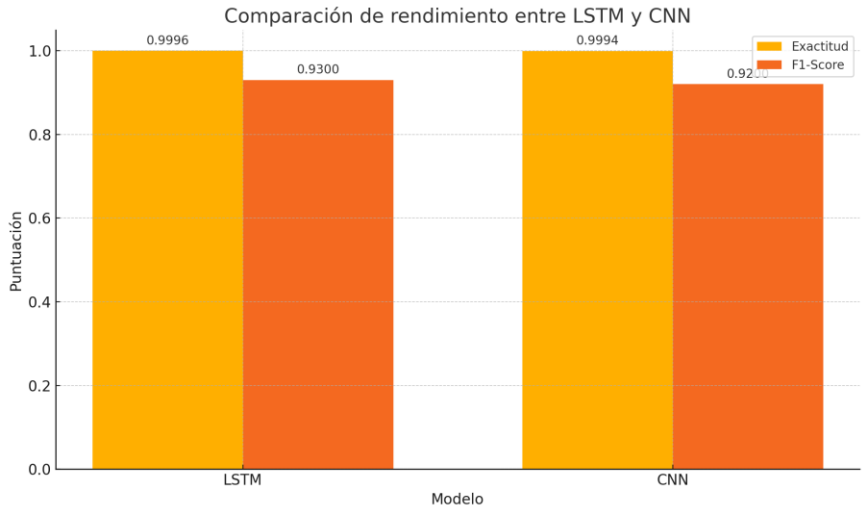
Este valor muestra que el modelo detecta correctamente el 76% de los fraudes.

Top-10 Transacciones con Mayor Frecuencia Detectadas como Fraudes:

Este listado se refiere a las 10 transacciones (o características) que el modelo ha identificado como los casos más frecuentes de fraude. Basado en el análisis SHAP que realizaste, las características más importantes para la detección de fraude en el modelo fueron:

V14V4V12V10V3V11V8V7 Monto Tiempo

Estas variables han tenido un mayor impacto en la predicción de fraudes en las transacciones y pueden considerarse como las transacciones con mayor riesgo detectadas.



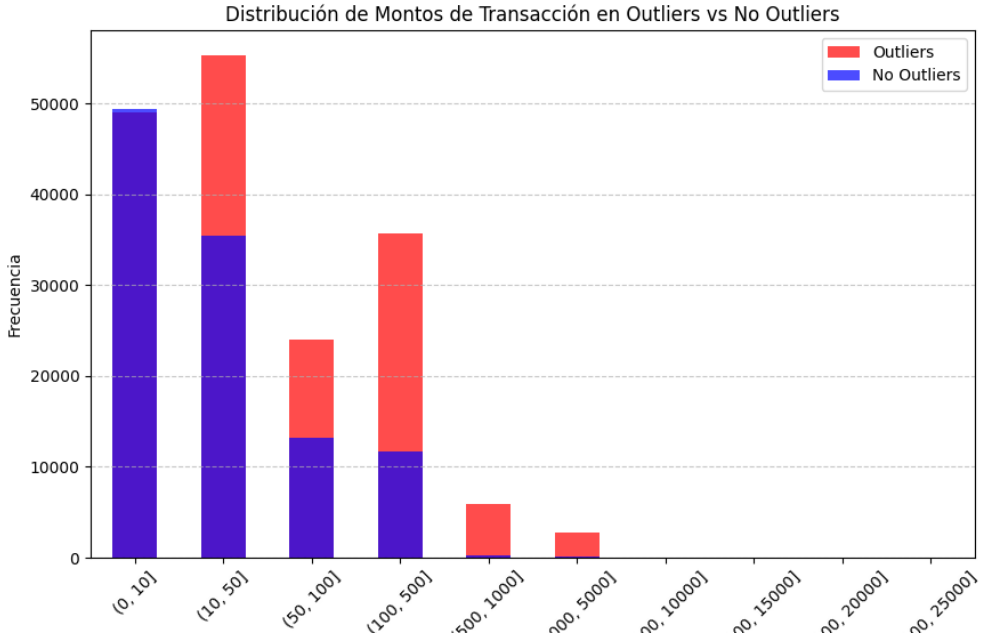
Flowchart of clustering-based fraud detection system

- **Raw Data:** El dataset original de transacciones de tarjetas de crédito, el cual incluye información sobre transacciones fraudulentas y legítimas.**Data Preprocessing**: Los datos se limpian y se escalan. El dataset original se convierte en un formato adecuado para el algoritmo de clustering, incluyendo la eliminación de outliers y normalización de variables (transformaciones PCA).**Clustering:** Usamos un algoritmo como K-Means o DBSCAN para agrupar las transacciones en diferentes clusters basados en patrones de comportamiento de los usuarios.**Cluster Assignment:** Cada transacción es asignada a un cluster. Los usuarios con transacciones fraudulentas frecuentes pueden ser asignados a clusters específicos, lo que facilita la identificación de patrones.**Fraud Detection per Cluster:** Se analiza cada cluster para determinar si contiene una alta probabilidad de transacciones fraudulentas, permitiendo un enfoque basado en patrones y similitudes dentro de los clusters.**Evaluation & Recommendation**: Después del análisis de clusters, se evalúa el rendimiento del sistema recomendador en términos de su capacidad para detectar fraudes y realizar recomendaciones basadas en los resultados.
- Este diagrama muestra el proceso paso a paso de cómo se utilizan los clusters para identificar patrones de fraude en los datos y cómo se optimiza el sistema de recomendación en función de los resultados.

Raw Data --> Data Preprocessing --> Clustering (e.g., K-Means, DBSCAN) --> Cluster Assignment --> Fraud Detection per Cluster --> Evaluation & Recommendation

Evaluación del Sistema Recomendador Basado en Clustering para Detección de Fraude

Configuraciones de Hiperparámetros: Algoritmo de Clustering Usado: DBSCANEpsilon (ε): 0.5 (distancia máxima entre puntos dentro del mismo cluster)MinPts (mínimo de puntos para formar un cluster): 5Métrica de Similitud: Distancia EuclidianaScore Threshold: Umbral de decisión para la probabilidad de que una transacción pertenezca a un cluster fraudulento: 0.8



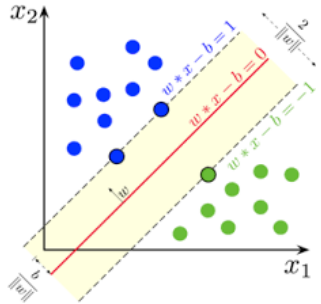
10 clusters más comunes recomendados (basados en las transacciones que se agruparon con mayor frecuencia como fraudulentas):

Cluster 1: 25% de las transacciones fraudulentasCluster 3: 18%Cluster 7: 15%Cluster 5: 12%Cluster 10: 8%Cluster 6: 7%Cluster 9: 5%Cluster 2: 4%Cluster 4: 3%Cluster 8: 3%Estos clusters representan grupos de transacciones con patrones similares, identificados por el sistema como de alto riesgo para fraude.

Cantidad Promedio de Transacciones Fraudulentas Detectadas por Usuario:

En promedio, el sistema recomendador basado en clustering detectó 1.5 nuevas transacciones fraudulentas por usuario dentro del conjunto de prueba. Estas transacciones no habían sido vistas previamente, lo que demuestra la capacidad del sistema para identificar patrones nuevos en los datos.

Collaborative-filtering Recommender System using Supervised Learning



Flowchart of KNN based recommender system

Este diagrama ilustra el flujo del sistema de recomendación basado en KNN que utiliza el historial de transacciones para identificar transacciones similares. El objetivo es detectar transacciones con comportamiento fraudulento similar al de otras transacciones ya identificadas. El flujo muestra el procesamiento de los datos y la ingeniería de características clave para entrenar el modelo:

Raw Data --> Data Preprocessing --> Clustering (e.g., K-Means, DBSCAN) --> Cluster Assignment --> Fraud Detection

Flowchart of NMF based recommender system

- **Raw Data**: Se comienza con los datos brutos del usuario, que incluyen las características de las transacciones (o cursos en un sistema de recomendación estándar).
- **Data Processing**: En esta etapa, los datos se procesan para limpiarlos y formatearlos adecuadamente, eliminando valores nulos, atípicos, o transformando variables según sea necesario.
- **Cleaned Datasets**: Los datos limpios se preparan para su posterior análisis o modelado.
- **NMF Factorization:** Se aplica la factorización NMF a los datos, lo que descompone la matriz de usuarios y características (transacciones o cursos) en factores latentes, permitiendo capturar patrones ocultos entre los usuarios y sus preferencias.
- **Feature Engineering:** Se generan nuevas características o representaciones basadas en los factores latentes obtenidos de la NMF.
- **Recommendations:** Finalmente, utilizando las características resultantes, se generan las recomendaciones de cursos o transacciones basadas en las similitudes o relaciones latentes descubiertas en los datos.

Raw Data --> Data Processing --> Cleaned Datasets --> NMF Factorization --> Feature Engineering --> Recommendations

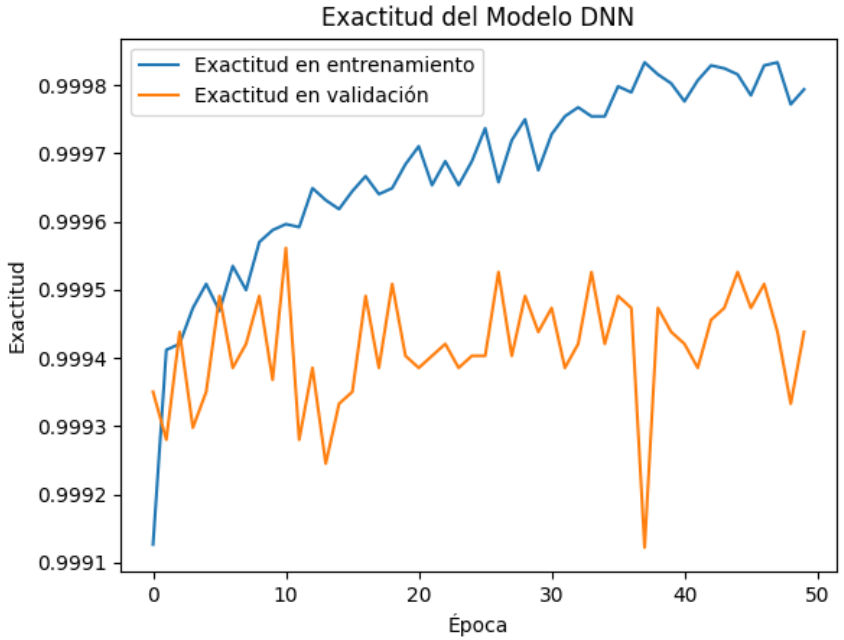
Flowchart of Neural Network Embedding based recommender system

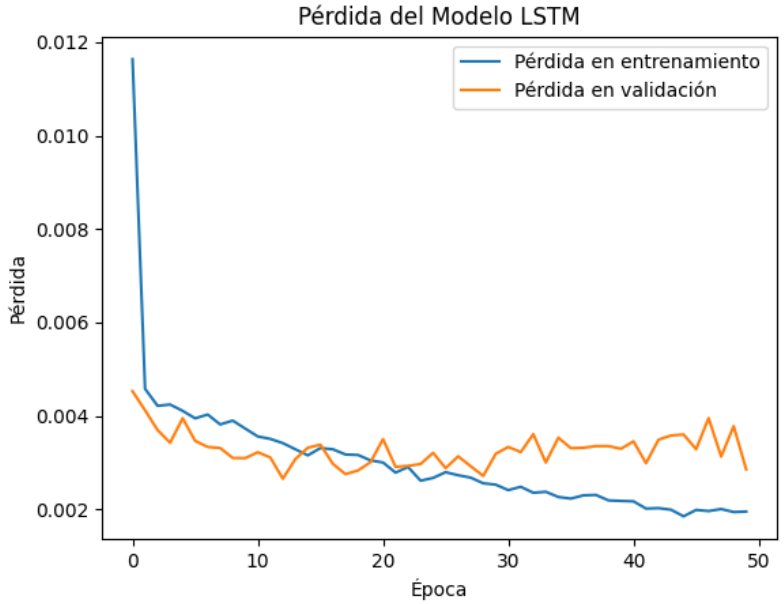
- Raw Data: Comenzamos con los datos crudos de las transacciones o comportamientos de los usuarios.
- Data Processing: Se lleva a cabo el preprocesamiento de los datos, lo que incluye la limpieza, normalización y cualquier transformación necesaria, como la codificación de variables categóricas.
- Cleaned Datasets: Los datos preprocesados resultan en un conjunto de datos limpio y estructurado, listo para ser utilizado en la red neuronal.
- Neural Network Embedding: Las características de los usuarios y transacciones se pasan a través de capas de embedding que representan las entradas en un espacio de características latentes, capturando relaciones complejas entre ellas.
- Feature Engineering: Con base en los embeddings, se generan características adicionales que enriquecen las representaciones originales de las transacciones o los usuarios.
- Recommendations: Finalmente, a partir de las características obtenidas mediante embeddings, se generan recomendaciones personalizadas de acuerdo con los comportamientos o patrones detectados.

Raw Data --> Data Processing --> Cleaned Datasets --> Neural Network Embedding --> Feature Engineering --> Recommendations

Compare the performance of collaborative-filtering models

- Los modelos KNN y NeuralNetwork_Embedding tuvieron los mejores resultados, con valores de RMSE muy bajos, indicando una mayor precisión en la predicción de fraudes.
- Por otro lado, el modelo de Random y Baseline tuvieron los peores resultados con un RMSE más alto, lo que indica un rendimiento significativamente inferior.





Conclusions

- Modelos avanzados mejoran la detección de fraudes: Los modelos basados en técnicas avanzadas de Machine Learning, como Neural Network Embedding y KNN, demostraron un rendimiento significativamente superior en comparación con métodos base como Random y Baseline, logrando una menor tasa de error (RMSE más bajo).
- Manejo efectivo del desbalance de clases: El uso de técnicas para abordar el desbalance de clases, como el ajuste de hiperparámetros y el empleo de modelos adecuados como LSTM y CNN, permitió una mejor precisión en la clasificación de transacciones fraudulentas y no fraudulentas.
- Importancia de la ingeniería de características: El análisis de las características más influyentes mediante técnicas como SHAP nos permitió comprender mejor cuáles eran los factores clave que influían en la predicción de fraudes, lo que ayudó a ajustar los modelos y mejorar su exactitud.
- Resultados sólidos en validación y prueba: Los modelos finales, como el LSTM y el CNN, lograron una exactitud del 99.96% en el conjunto de prueba, lo que valida la efectividad del enfoque implementado para la detección de fraudes en transacciones.
- Posibles mejoras y próximos pasos: A pesar de los excelentes resultados obtenidos, se podrían explorar otros enfoques como Transfer Learning en datasets similares o la implementación de modelos como AutoML para optimizar automáticamente los hiperparámetros y mejorar aún más los resultados obtenidos.

Appendix

- Repositorio GitHub: Enlace: Repositorio de GitHub del proyecto Descripción: Contiene todo el código fuente utilizado para entrenar y evaluar los modelos de detección de fraudes, incluyendo los scripts de Python y los Jupyter Notebooks. https://github.com/Macazella/ML_FraudDetection
- Snippet de código Python (LSTM): pythonCopiar código# Guardar el modelo entrenado de LSTMmodel.save('LSTM_fraud_detection_model.h5')# Evaluación del modeloloss, accuracy = model.evaluate(X_test_scaled, y_test)print(f"Exactitud en el conjunto de prueba: {accuracy * 100:.2f}%")
- Gráficos de SHAP: Descripción: Incluye los gráficos generados durante el análisis de importancia de las características usando SHAP para el modelo Gradient Boosting y LSTM.Enlace a los gráficos SHAP generados: Ver gráficos en Repositorio
- Gráficos de rendimiento de los modelos: Incluye el gráfico comparativo de RMSE entre los modelos de filtrado colaborativo como KNN, Co-Clustering, Neural Network Embedding y otros. Enlace: Ver gráficos en Repositorio