Unsupervised Learning Report: Fraud Detection Using Clustering

By Veronica Magdaleno

Final Project

Main Objective of the Analysis

The primary objective of this analysis was to uncover hidden patterns in transaction behavior using unsupervised learning, with a focus on clustering. The goal was to determine whether naturally occurring clusters in the data could reveal transaction types that are more susceptible to fraud and to generate insights that could support proactive fraud detection strategies.

Given the high class imbalance typical in fraud datasets, clustering offers a valuable way to segment transaction behaviors without requiring labeled fraud data at the outset. This can help fraud teams detect new and emerging fraud patterns before they're labeled.

Dataset Description

This analysis used the Variant II dataset from the FiFAR fraud detection study. The dataset contains 1,000,000 rows of transaction data, with features related to user behavior, financial activity, device usage, and risk metrics. The dataset includes a binary target column `fraud_bool`, which was excluded during model training but used afterward for evaluation and interpretation.

The final dataset included ~50 engineered features after preprocessing, including:

- Numerical features (e.g., income, credit risk score, transaction velocities)
- Categorical variables (e.g., payment type, housing status, device OS)
- Session behavior and device characteristics

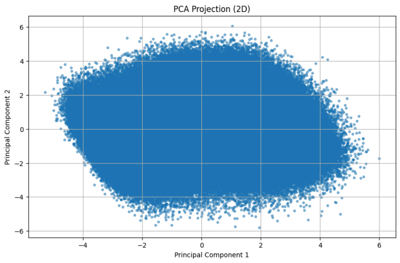
  Sample Columns:

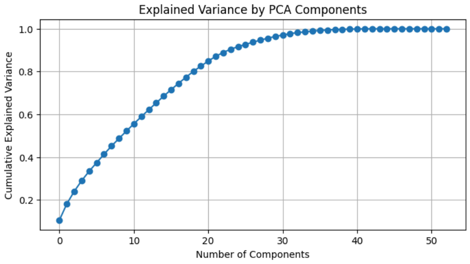
- `income: float64`
- `name_email_similarity: float64`

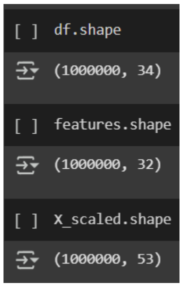
- `prev_address_months_count: int64`
- `current_address_months_count: int64`
- `customer_age: int64`
- `days_since_request: float64`
- `intended_balcon_amount: float64`
- `payment_type: object`
- `zip_count_4w: int64`
- `velocity_6h: float64`
- `velocity_24h: float64`

Data Preparation

- Dropped `fraud_bool` prior to model fitting to maintain unsupervised integrity
- Scaled numerical features using StandardScaler
- Encoded categorical features using one-hot encoding
- Performed dimensionality reduction via PCA (Principal Component Analysis) for visualization and to aid distance-based clustering models
- Cyclical encoding was applied to month data
- Outlier handling was addressed through scaling and variance analysis







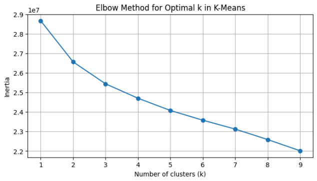
```
[ ] df.shape
(1000000, 34)

[ ] features.shape
(1000000, 32)

[ ] X_scaled.shape
(1000000, 53)
```

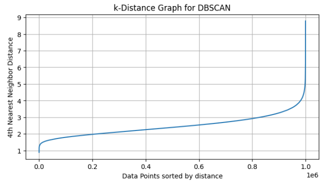
Unsupervised Models Compared

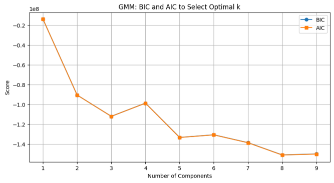
Three clustering algorithms were tested:

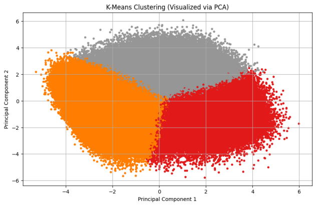
Model	Description

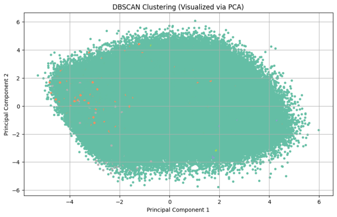
K-Means	Hard clustering; assumes spherical cluster shapes
DBSCAN	Density-based; identifies outliers; doesn’t require pre-setting number of clusters
GMM	Soft clustering; probabilistic cluster membership with flexible shapes

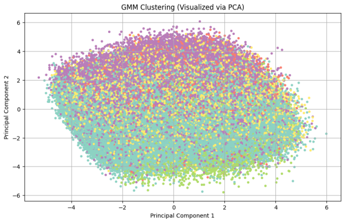












Silhouette Score Results (sample of 10,000 rows)

Model	Score	Interpretation
K-Means	0.0677	Weak but present structure
DBSCAN	-0.0915	Poor clustering; failed to segment well
GMM	0.0830	Best performer; captured overlapping behaviors effectively

Final Recommendation: Gaussian Mixture Model (GMM) GMM slightly outperformed K-Means in silhouette score and revealed more interpretable cluster patterns.

Key Findings and Insights

After fitting GMM with 3 components and reintroducing the `fraud_bool` labels:

Cluster 2 showed the highest fraud rate (2.25%), despite being the smallest group.

- Older users (avg. age 49)

- Lower credit scores
- Higher foreign request rate (6%)
- Shortest session times
- Lower device diversity

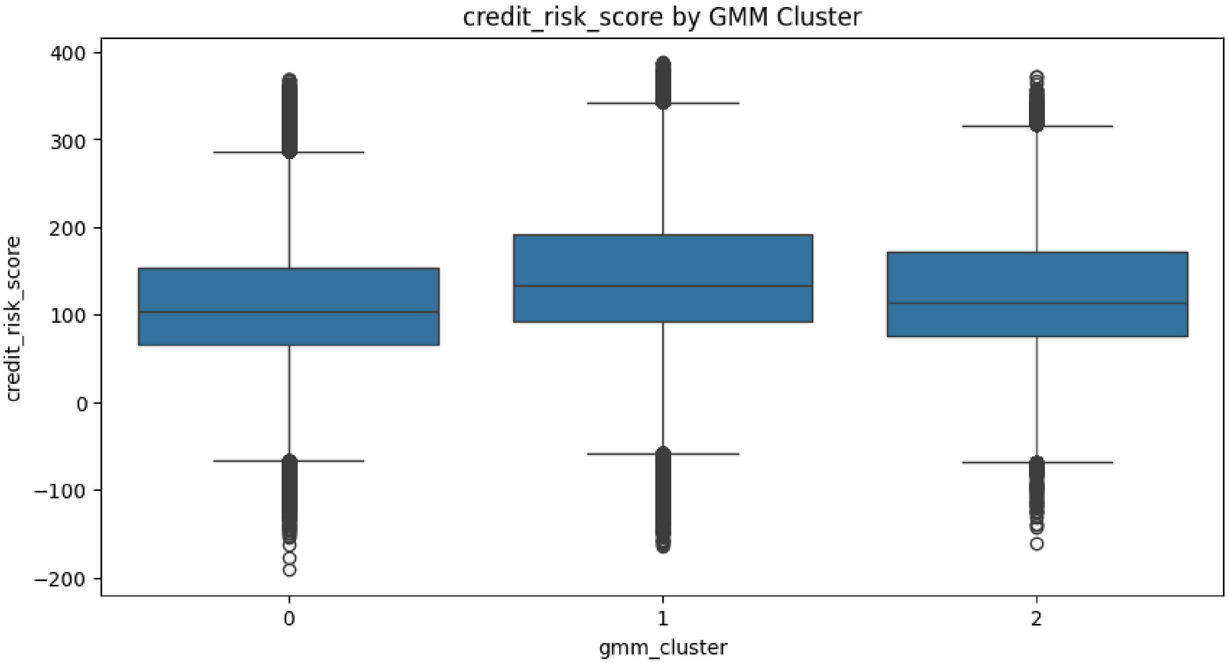
Though it contains only 760 of the 10,000 total fraud cases, Cluster 2 has the highest per-capita fraud concentration.

Cluster 0 had the most total fraud cases but a lower fraud rate (1.4%)

- Likely due to its larger population size

Cluster 1 had the lowest fraud rate (0.99%)

- These users had the highest credit scores, longest bank history, and no foreign requests



Limitations and Next Steps

Limitations:

- Silhouette scores were low overall, which may indicate overlapping or noisy clusters
- `device_fraud_count` had no variance, reducing its utility
- Some behaviors (e.g., user navigation, transaction time-of-day) were not captured

Recommended Next Steps:

- Incorporate additional behavioral signals (clickstream, velocity deltas, device switching)
- Explore semi-supervised learning with limited fraud labels
- Consider isolating rare events through anomaly detection or deep autoencoders