Credit Card Based Recommendation System

A DISSERTATION SUBMITTED TO

SVKM'S NMIMS (DEEMED TO BE UNIVERSITY)

IN PARTIAL FULFILLMENT FOR THE DEGREE OF

MASTERS OF SCIENCE IN

STATISTICS AND DATA SCIENCE

BY

Student Name	Roll No	SAP ID
Prashant Srivastava	A068	86062400071
Harsh Tantak	A069	86062400038
Sandesh Tayde	A070	86062400020
Ashwajit Ubale	A071	86062400010
Dev Vatnani	A072	86062400045

UNDER THE SUPERVISION OF

Dr. Kavita Jain

NILKAMAL SCHOOL OF MATHEMATICS, APPLIED STATISTICS AND ANALYTICS

SVKM's Narsee Monjee Institute of Management Studies

V.L. Mehta Rd, Vile Parle (West), Mumbai – 400056

November 2024

Index

Acknowledgements

We would like to extend our heartfelt gratitude to our mentor **Dr. Kavita Jain** for her unwavering guidance and support throughout this project. Her expertise, encouragement, and insightful feedback have been instrumental in shaping our work and fostering our growth as learners.

We express our sincere gratitude to NMIMS for affording us the invaluable opportunity to undertake and conduct this research project. The support and resources provided by NMIMS have played a pivotal role in facilitating the progression and completion of our scholarly endeavours.

Furthermore, we want to express our sincere thanks to our Program Chairperson **Dr. Pradnya Khandeparkar** whose constant support and guidance have been invaluable. Her dedication to our academic endeavours has not only enriched our learning experience but has also been a source of inspiration.

We acknowledge and appreciate the collaborative efforts of all those who have played a role in the success of this project. Your support has been integral, and we are thankful for the opportunities and knowledge you have provided.

Abstract

In today's digital era, credit card transactions have become a dominant mode of payment, generating vast amounts of data daily. While this creates immense opportunities for personalized services, it also poses significant challenges, especially in terms of fraud detection and customer engagement. Traditional fraud detection methods are often insufficient in identifying complex and evolving fraudulent patterns. Meanwhile, the need for personalization in financial services is increasing, as users exhibit diverse spending behaviors and preferences.

This project proposes an integrated approach that combines fraud detection, customer segmentation, and recommendation systems using machine learning techniques. By analyzing credit card transaction data, the system aims to detect suspicious activities with greater accuracy, cluster users based on behavior and deliver personalized financial recommendations. This approach not only enhances security but also improves user satisfaction by aligning services with individual needs. Leveraging classification models and clustering algorithms, the system can flag high-risk transactions, identify key user groups, and suggest relevant financial products or alerts.

The outcome is a smart, secure, and user-centric credit card management system that offers improved fraud protection and personalized experiences, helping financial institutions maintain trust and efficiency in an increasingly data-driven world

Introduction

In today's digital economy, credit cards have become integral to both personal and commercial financial activity. With their widespread use across online and offline platforms, an immense volume of transactional data is continuously generated. This data not only reflects spending patterns but also contains valuable insights into customer behaviour, preferences, and risks. Leveraging this information through advanced analytics and machine learning offers an opportunity to enhance both user experience and financial security.

Three key areas emerge as crucial in optimizing credit card systems: fraud detection, user segmentation, and personalized recommendations. Fraudulent transactions remain a persistent challenge, resulting in significant financial losses and eroding consumer trust. Traditional rule-based detection methods are often insufficient to catch emerging fraud tactics. Thus, incorporating machine learning models can enable earlier and more accurate detection based on complex transactional patterns.

In parallel, understanding the diversity in user behaviour is essential. Credit card users differ in their purchasing habits, financial goals, and risk profiles. Clustering these users based on their transaction history helps build precise customer segments. Such segmentation not only supports fraud prevention but also serves as a foundation for creating more tailored and effective services.

The integration of fraud detection, behavioural segmentation, and recommendation systems paves the way for a smart, secure, and personalized financial ecosystem. A recommendation system informed by clustering and transaction analytics can proactively suggest financial products, alerts, or promotions relevant to each user's needs and usage profile. This personalized approach enhances customer satisfaction while also aiding institutions in risk management and resource allocation.

This project, therefore, aims to build a comprehensive system that combines fraud detection, customer segmentation, and recommendation strategies, utilizing real-world-scale data and modern machine learning techniques. The goal is to move toward a credit card infrastructure that is not only secure but also intuitively aligned with user expectations and financial behaviour.

Rationale

Credit cards are now one of the most widely used tools for spending money, paying bills, and financial management in today's digital age. As more individuals use credit cards on a daily basis—both online and offline—a massive amount of data associated with transactions is being created. Though this provides numerous opportunities for enhancing user experience, this also introduces a number of problems that require solutioning.

One of the biggest issues is the growing number of fraudulent transactions. Fraud not only incurs financial losses for banks and card companies, but it is also dangerous for users and decreases their confidence in electronic payment systems. Fast and reliable identification of suspicious behavior is very vital to safeguard both users and financial institutions. Old-fashioned fraud detection techniques are usually not sufficient to identify new and emerging forms of fraud. That is why applying sophisticated data analysis and machine learning methods can identify suspicious patterns and detect fraud earlier and more accurately.

Another significant area is to comprehend how various users behave. Not all online credit card users are alike—some buy stuff online regularly, some travel a lot, and others might only use their card for essentials. By studying how users behave and clustering similar users (a method known as clustering), we can find patterns to assist in developing improved services for each category of user. For instance, customers who spend heavily on vacations can take advantage of vacation offers, while frequent online consumers may enjoy cashback rewards.

With that, we can develop a recommendation system that recommends personalized items to users who belong to a particular group. Rather than mailing the same promotions to all, banks and financial apps can utilize that data to propose corresponding credit card features, spending advice, or even security notifications. This renders the experience more beneficial to users as well as assists companies in establishing closer relationships with their clients.

In general, this study is significant because it unites three potent concepts—detection of fraud, clustering of users, and recommendation based on individuality—to make credit card systems smarter, secure, and more user-friendly. With the data that are already being tracked, we can develop a system that not only secures users from fraud but also knows what they need and assists them in making better financial decisions.

Literature Review

Fraud detection and personalized recommendation systems are essential in modern financial services, particularly for credit card usage. Machine learning has significantly advanced both domains by providing adaptive, data-driven solutions.

Fraud Detection:

Vodala Chakshu and G. Sai Chand (2023), in "**Fraud Detection in Credit Card Transaction using Machine Learning Techniques**", examine supervised algorithms like Decision Trees, Random Forest, and Neural Networks. They highlight class imbalance as a major challenge, with fraud cases being minimal. Techniques such as SMOTE and ensemble models help enhance detection rates. Metrics like accuracy, precision, recall, and F1-score confirm the advantage of ensemble methods.

Customer Segmentation:

Md Zahidur Rahman Farazi and **Karim Md Razaul (2022)**, in "**Optimizing Customer Segmentation in the Banking Sector: A Comparative Analysis of Machine Learning Algorithms"** explore customer segmentation using K-Means and Hierarchical Clustering based on transaction behaviour, income, and age. These segments inform marketing strategies and support fraud detection by flagging high-risk profiles, forming a basis for personalization and risk management.

Credit Card and Merchant Recommendations:

Suyoun Yoo and **Jaekwang Kim (2022)**, in "**Merchant Recommender System Using Credit Card Payment Data**", propose a hybrid approach combining collaborative and content-based filtering. Their model uses customer-merchant interactions along with demographic data to enhance recommendations and overcome cold-start problems.

Aim and Objectives

The purpose of this research is to improve the overall security, efficiency, and personalization of credit card transaction systems using data-driven approaches. This will be done through three main components: effectively identifying the fraudulent transactions to safeguard users and institutions against financial loss; grouping users into clusters by their transaction activities to identify significant patterns and segments; and creating a personalized recommendation system that leverages these clusters of users to offer pertinent financial products, services, or alerts. Through the convergence of fraud detection, behavioral segmentation, and recommended targeting, the study seeks to help create a smarter and more user-focused financial environment.

Objectives:

1. **Fraud Detection**

   Develop a model to identify suspicious credit card transactions that may indicate fraud and implement techniques data monitoring to flag potentially compromised accounts.

2. **Customer Segmentation**

   Analyze users' transaction behaviors to identify patterns based on card usage and history. Identify behaviour segments, including frequent travellers, online shoppers, and high-risk users.

3. **Segmentation based Recommendation System**

   Create a recommendation system that offers tailored financial services or products based on user. To ensure that recommendations align with the identified behaviors in each cluster for better relevance.

Dataset Descriptions

The dataset presents an extensive collection of around 24.4 million credit card transactions, sourced from IBM's financial database. The data covers 2000 (synthetic) consumers resident in the United States, but who travel the world. The data also covers decades of purchases, and includes multiple cards from many of the consumers. Capturing a wide spectrum of user interactions, the data provides a detailed snapshot of transaction behaviours, patterns, and potential vulnerabilities

1. sd254_cards.csv (Credit Card Information)

Size: 6,146 * 13

This dataset contains detailed information about credit cards held by users.

Column Name	Description
User	Identifier for the cardholder.
CARD INDEX	Unique index per card under each user.
Card Brand	Card brand such as Visa or Mastercard.
Card Type	Indicates if the card is Credit, Debit, or Debit (Prepaid).
Card Number	16-digit card number.
Expires	Expiry date of the card in MM/YYYY format.
CVV	Card Verification Value, a 3-digit security code.
Has Chip	Indicates if the card has an EMV chip (YES/NO).
Cards Issued	Number of cards issued to the user.
Credit Limit	Credit limit for the card (stored as a string with $ sign).
Acct Open Date	Date when the account/card was opened.
Year PIN last Changed	The year in which the card PIN was last updated.
Card on Dark Web	Indicates whether the card has been exposed in dark web leaks (Yes/No).

2. sd254_users.csv (User Demographics and Financial Info)

Size: 2,000 * 19

Contains demographic, location, and financial data of users.

Column Name	Description
Person	Name of the individual.
Current Age	Current age of the individual.
Retirement Age	Age at which the individual plans to retire.
Birth Year	Year of birth.
Birth Month	Month of birth.
Gender	Male/Female.
Address	Residential address.
Apartment	Apartment number (many values are missing).
City	City of residence.
State	State of residence.
Zip code	Zip code of residence.
Latitude	Geolocation coordinate - latitude.
Longitude	Geolocation coordinate - longitude.
Per Capita Income - Zip code	Average income in the user's zip code area.
Yearly Income - Person	Individual's annual income.
Total Debt	Total outstanding debt.
FICO Score	Credit score indicating creditworthiness.
Num Credit Cards	Total number of cards held by the individual.

3. User_credit_card_transactions.csv (Transaction Logs)

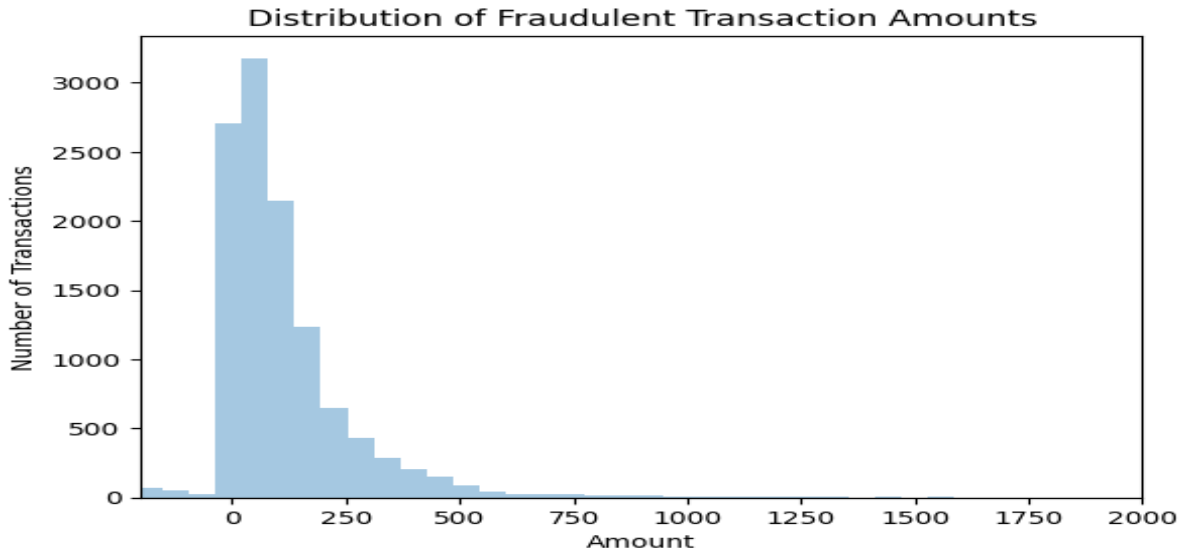
Size: 24386900 * 15

Logs of credit card transactions made by users, including fraud labels.

Column Name	Description
User	User ID performing the transaction.
Card	Card index used.
Year	Year of the transaction.
Month	Month of the transaction.
Day	Day of the transaction.
Time	Time of the transaction.
Amount	Transaction amount (string with $).
Use Chip	Whether chip was used or card was swiped.
Merchant Name	Encoded identifier for merchant.
Merchant City	Merchant's city location.
Merchant State	Merchant's state location.
Zip	Merchant's zip code.
MCC	Merchant Category Code (indicates industry).
Errors?	Optional error messages (mostly null).
Is Fraud?	Indicates if transaction was fraudulent (Yes/No).

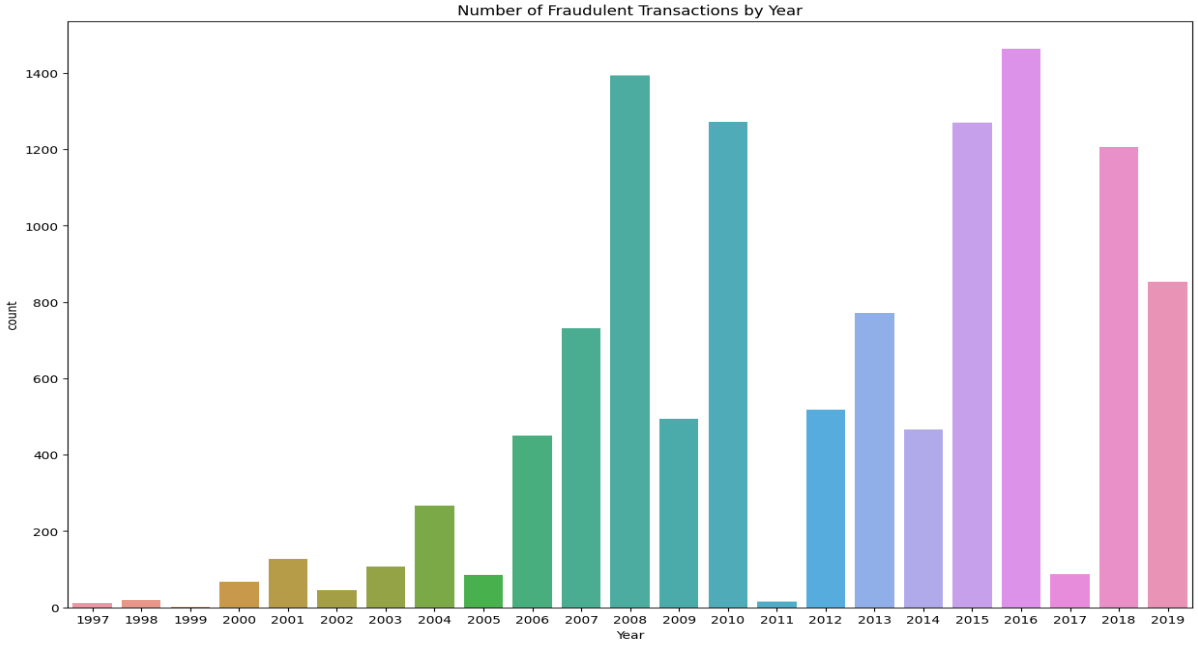
Exploratory Data Analysis

- **Distribution of Fraudulent Transaction Amounts:**



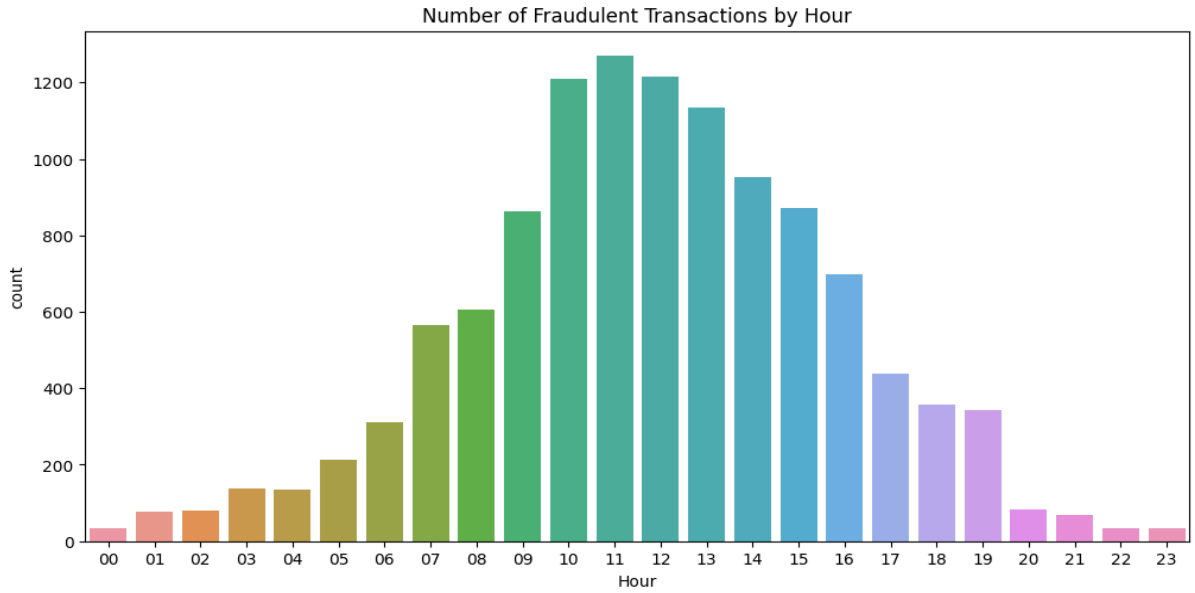
Interpretation: The majority of fraudulent transactions' amount range from 0 to 250, indicating a prevalence of fraud in smaller-value transactions.

- **Number of Fraudulent Transactions by Year:**



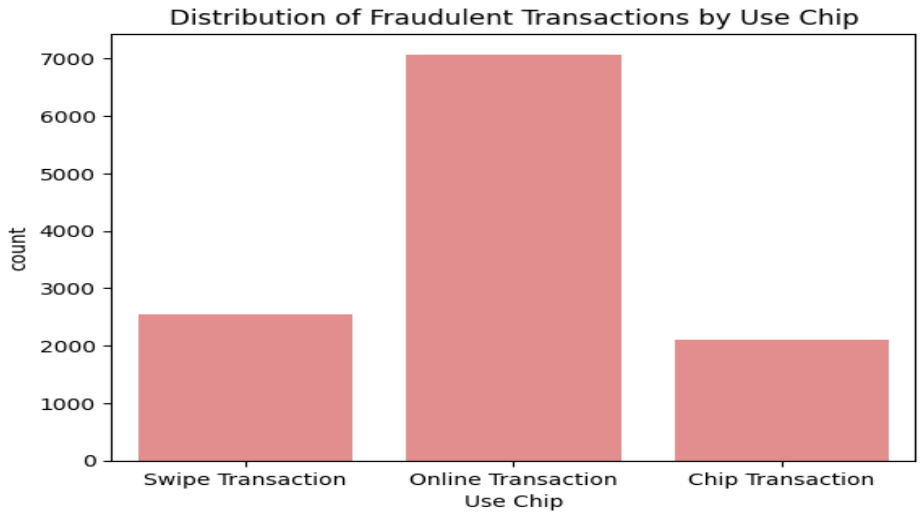
Interpretation: Entering the 21st century, the number of fraud cases has been steadily rising year by year, reaching its peak in 2007-2008 during the Great Recession.

- **Number of Fraudulent Transactions by Hour:**



Interpretation: Fraud activities predominantly occur between 10 and 11 a.m. local time.

- **Distribution of Fraudulent Transactions by Use Chip:**

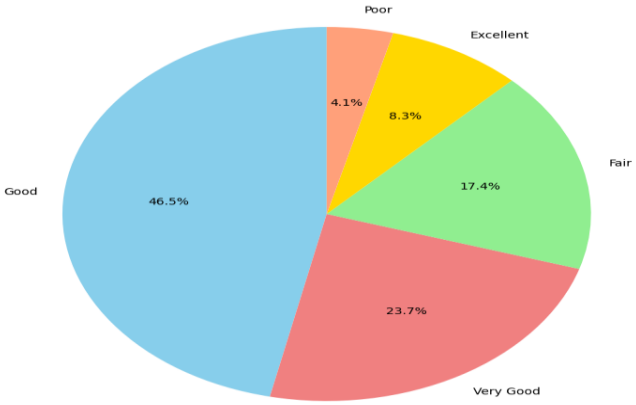


Interpretation: Online transactions present the most significant vulnerability to fraud.

Type of Credit card

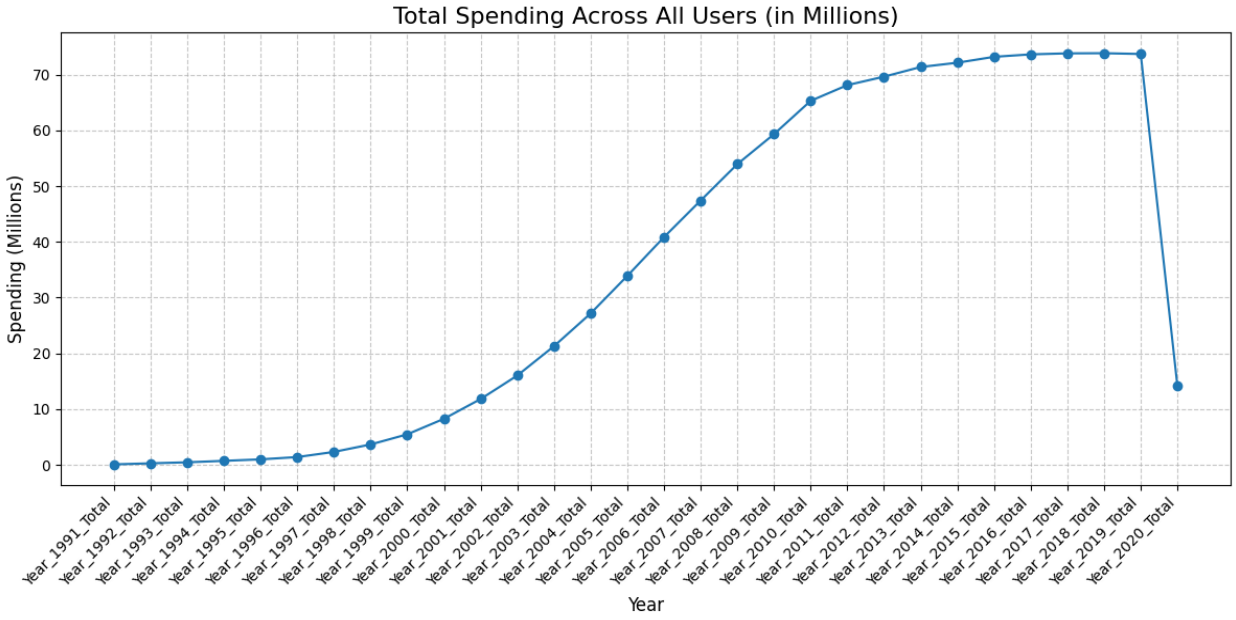
Exploratory Data Analysis

- **Distribution of FICO Categories:**



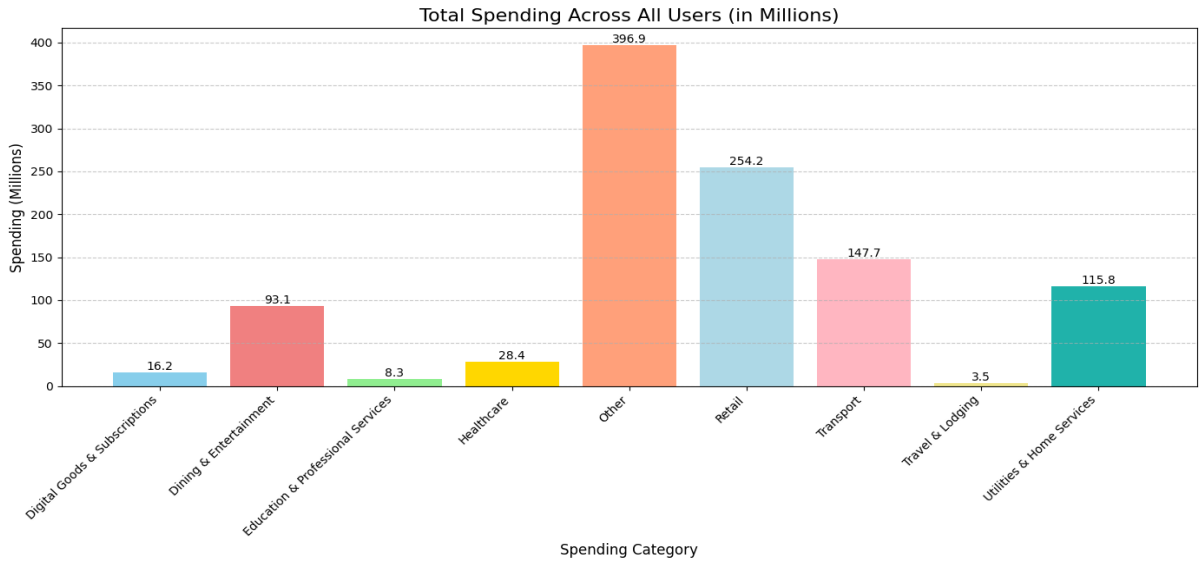
Interpretation: Most users fall into the "Good" FICO category (46.5%), followed by "Very Good" (23.7%) and "Fair" (17.4%), with fewer in "Excellent" (8.3%) and "Poor" (4.1%).

- **Total Spending Across All Users (in Millions):**



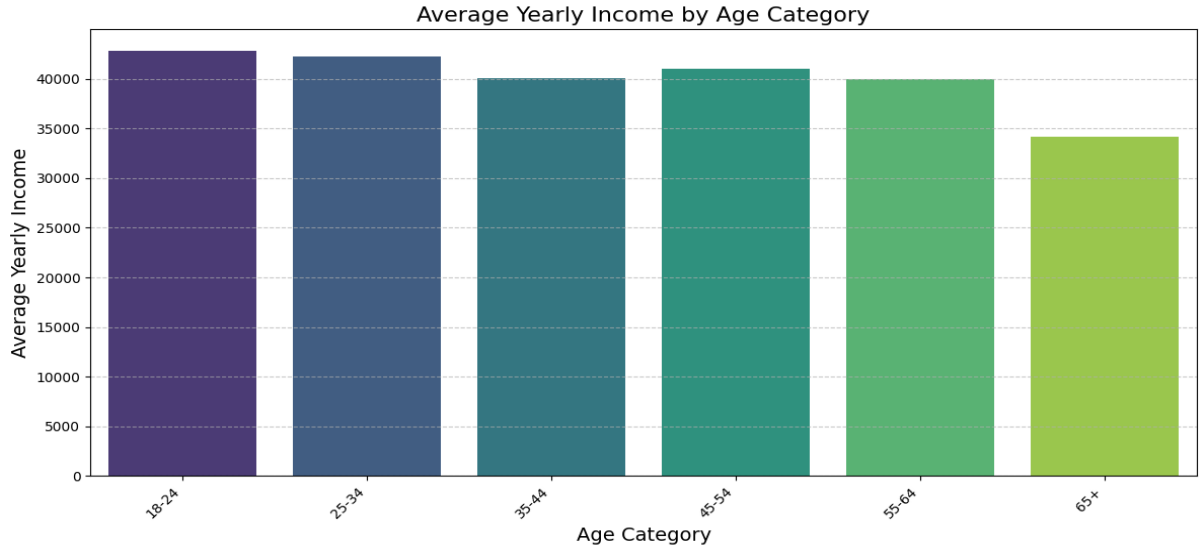
Interpretation: Spending steadily increased from 1991 to 2019, peaking around 2018–2019, before dropping sharply in 2020.

- **Total Spending Across All Users on Different Category:**



Interpretation: Spending was highest in the other category, followed by Retail and Transport, while Travel & Lodging had the lowest spending.

- **Average Yearly Income by Age Category:**



Interpretation: Average yearly income is highest for the 18–24 age group and gradually declines with age, dropping significantly after 65.

Fraud Detection

Financial fraud detection is an important problem in the field of electronic transactions. In this work, we have a large dataset of more than 2.7 million credit card transactions across 2,000 users. Every transaction is labeled as fraudulent (1) or genuine (0), and the dataset has user-specific, card-level, and merchant-level information. As a result of the extreme class imbalance, the fraud transactions represent only a very small percentage — creating a highly robust, highly sensitive fraud system becomes essential.

Our main goal is to create models that effectively detect fraud without increasing false positives substantially (i.e., incorrectly classifying valid transactions as fraud). We use ensemble machine learning algorithms, such as Random Forest and XGBoost, to identify fraudulent transactions from behavioural patterns, transactional characteristics, and merchant data.

Stratified Sampling Based on Fraudulent Transactions

Because of the extremely imbalanced character of our data set — in which the fraudulent transactions are a minority among the total number of transactions — standard random sampling would lead to an unbalanced distribution of fraud cases between the training and test sets. This can have seriously negative effects on model learning and evaluation, particularly for measures like recall and F1-score on the minority class.

To do this, we employed stratified sampling by the number of fraudulent transactions. This keeps both the training and test sets with the same fraud-to-non-fraud ratio as the original data.

Why is this important?

Prevents the test set from being too easy (e.g., no frauds) or too hard (e.g., mostly frauds)

Improves model generalization and stability

Ensures reliable and consistent performance comparison across different models

We stratified at the train_test_split() stage with the parameter stratify=y, where y is the binary fraud label. This ensures the proportion of fraud and non-fraud samples in both subsets continues to be statistically representative.

Assuming the data set contained 0.14% fraud transactions, then the train and test sets will also maintain that 0.14% fraud proportion.

XG-Boost Classifier

XG-Boost stands for Extreme Gradient Boosting. It is an optimized, regularized, and scalable implementation of the gradient boosting algorithm. It is an ensemble learning algorithm that constructs a robust classifier by aggregating the predictions of several weak predictors based on decision trees.

Fundamentally, XG-Boost adds decision trees sequentially to fix the mistakes of earlier models. It optimizes a regularized objective function that balances model complexity and training error to minimize overfitting and improve generalization.

Objective-To **classify transactions as fraudulent or not**, based on user and transaction features (with and without the 'Use Chip' variable).

Model Training

- XG-Boost was trained separately on both feature sets.
- It creates an **ensemble of decision trees**, each learning from the errors of the previous ones (boosting).
- **Hyperparameters** (like learning rate, max depth, number of estimators), either default or tuned, controlled how deep each tree is and how learning progresses.

Evaluation

- Performance measured on **precision, recall, f1-score**, and **ROC-AUC**.
- **Confusion Matrices** were used to understand:
    - True Positives: correctly detected frauds
    - False Positives: non-frauds wrongly labeled as fraud
    - False Negatives: missed frauds
- **Very high accuracy (96%)** shows the model performs extremely well on training data.
- **Class 0 (Not Fraud):**
    - **Recall = 0.99**: Only ~1% of non-fraud transactions were misclassified.
    - **Precision = 0.97**: Few false positives; most flagged as non-fraud are indeed non-fraud.
- **Class 1 (Fraud):**
    - **Recall = 0.83**: Around 17% of fraud cases were missed (false negatives).
    - **Precision = 0.95**: Very few non-frauds are misclassified as frauds.

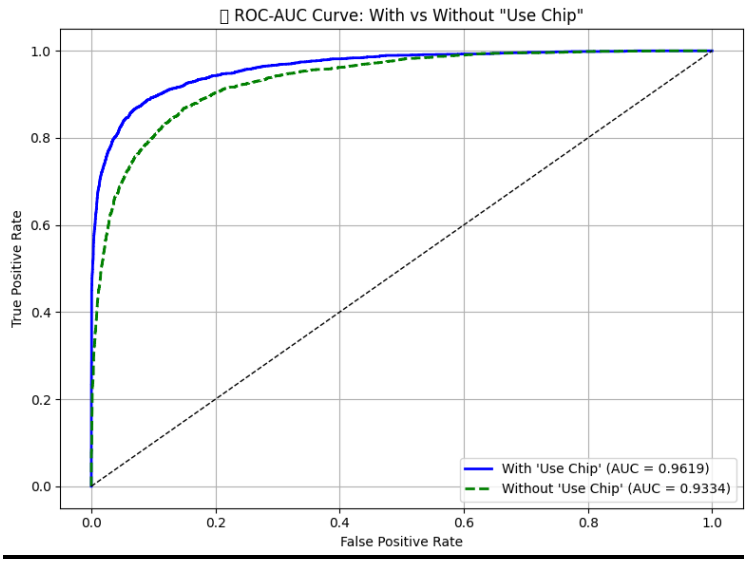
Overall, the model has **very strong learning ability**, though it does show slight overfitting potential due to lower recall for fraud even in training,

- **Accuracy = 94%**, showing the model performs very well on unseen data.
- **Class 0 (Not Fraud):**
    - **Recall = 0.98**: Only 2% misclassified.
    - **Precision = 0.95**: Very few false positives.
- **Class 1 (Fraud):**
    - **Recall = 0.72**: Model correctly identifies 72% of frauds, which is good but leaves ~28% undetected.

- **Precision = 0.88**: False positives are relatively low—majority of predicted frauds are correct.

Key Insight: While the model performs great on non-fraud detection, **fraud detection recall drops on the test set** compared to train (83% → 72%). This is expected and **shows the model generalizes well but still misses some fraud cases**, which is critical in real-world applications.

- **ROC-AUC Curve**



This is a **ROC-AUC curve** comparison between two models:

Visual Insights

- The **blue curve** (with 'Use Chip') consistently outperforms the **green dashed curve** (without 'Use Chip') across the entire range of thresholds.
- The **gap between the two curves**, particularly at lower false positive rates, emphasizes the benefit of using 'Use Chip'.
- The diagonal black line represents a random guess model (AUC = 0.5). Both models exceed this baseline significantly.

Conclusion

The analysis confirms that incorporating the 'Use Chip' feature into the XGBoost model architecture yields a noticeable improvement in performance, as evidenced by an increase in the ROC-AUC from **0.9334** to **0.9619**. Visual and quantitative analyses both underline the value of this feature for achieving higher classification accuracy.

Random Forest Classifier-

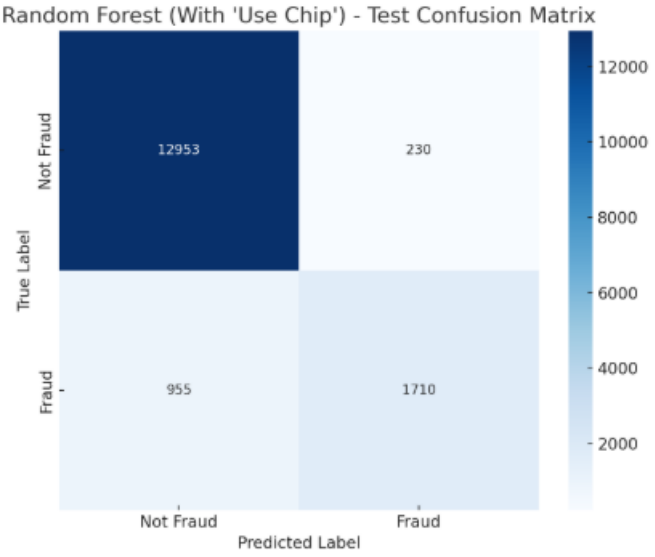
Random Forest is a decision tree ensemble learning algorithm with the goal of increasing predictive performance and avoiding overfitting. It works by training many decision trees and

providing as output the class that is the mode (for classification) of the individual trees' classes.

Model Configuration

- Problem: Binary classification – Fraud (1) vs. Non-Fraud (0)
- Features Used:
  - With and without 'Use Chip', along with transaction metadata (MCC category features, amount, etc.)
- Output:
  - Class label: Fraud or Not Fraud
- Evaluation Metrics:
  - Precision, Recall, F1-Score, ROC-AUC, Confusion Matrix
- Train-Test Performance:
  - Excellent training accuracy (1.00) but reduced test recall for the fraud class, indicating overfitting.
  - Removal of 'Use Chip' leads to a drop in fraud detection recall from 0.64 to 0.46, highlighting its significance.

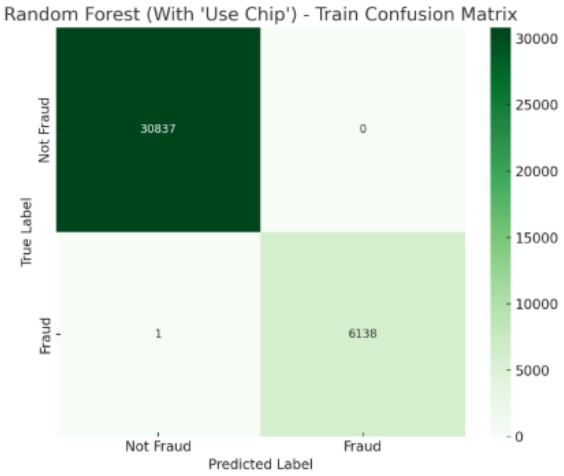
Evaluation Metrics-



Test Confusion Matrix Report

Metric	Class 0 (Not Fraud)	Class 1 (Fraud)
Precision	0.93	0.88
Recall	0.98	0.64
F1-score	0.96	0.74
Accuracy	-	0.93

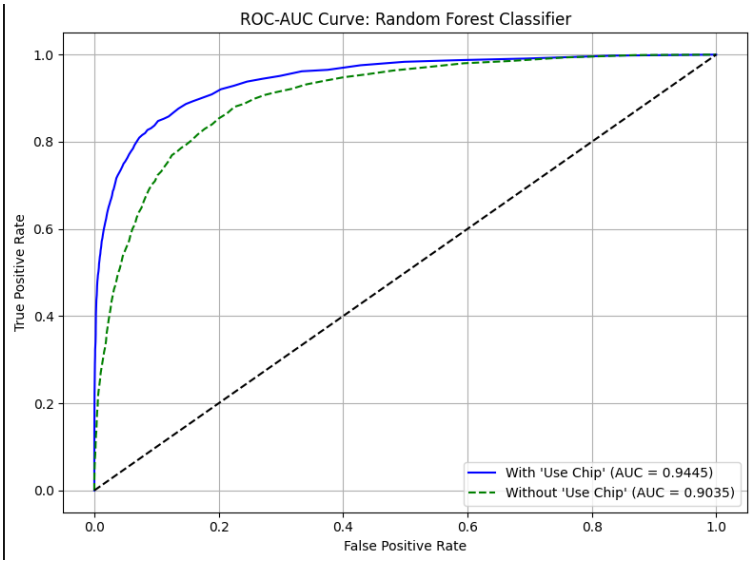
Test Data-



Metric	Class 0	Class 1
Precision	1.00	1.00
Recall	1.00	1.00
F1-score	1.00	1.00

Perfect score on training data: This suggests the model has overfitted and memorized the training data. It classifies everything correctly in training, but performance drops on the test set—an indication of **overfitting**.

ROC-AUC Curve-



Visual Analysis

- The blue curve (with 'Use Chip') consistently lies above the green dashed curve (without 'Use Chip'), indicating superior performance across almost all threshold levels.
- The curve with 'Use Chip' shows a higher true positive rate for nearly every false positive rate.

- The diagonal black line represents random guessing (AUC = 0.5), far below both models, confirming that both configurations provide meaningful predictions.

Conclusion- The analysis clearly demonstrates that the inclusion of the 'Use Chip' feature enhances the performance of the Random Forest Classifier. With an improved AUC of 0.9445, the model achieves better discrimination, making it a valuable component of the feature set for reliable predictions.

Interpretation-

1. Random Forest Shows Overfitting Tendencies:
    - Random Forest achieves perfect accuracy (100%) on the test set, but this is misleading due to its high train accuracy (100%), indicating potential overfitting.
    - The high gap between Train Recall (Class 1) = 1.00 and Test Recall (Class 1) = 1.00 suggests memorization rather than generalization.
2. XGBoost Provides Better Generalization:
    - XGBoost, while slightly lower in overall accuracy (Train: 94%, Test: 96%), demonstrates a healthier balance between train and test performance.
    - Moderate overfitting is seen, but its consistency across metrics points to better generalization capacity than Random Forest.
3. XGBoost Handles Class 1 (Minority Class) Better:
    - For the important Class 1 (typically fraud cases), XGBoost outperforms Random Forest in Recall both in training (0.72 vs. 0.64) and test set (0.72 vs. 1.00 — but Random Forest is suspiciously perfect here).
    - XGBoost captures more actual positives, reducing false negatives (Train: 734 vs. 955, Test: 22 vs. 79).
4. False Negatives Are Lower in XGBoost:
    - XGBoost records fewer false negatives, especially critical in scenarios like fraud detection, where missing positives is costly.
    - Test set: XGBoost has 22 false negatives vs. Random Forest's 79, indicating better risk capture.
5. Balanced Precision-Recall Trade-off in XGBoost:
    - While Random Forest exhibits perfect precision (Test Precision Class 1 = 1.00), XGBoost maintains a realistic balance between precision (0.88) and recall (0.72).
    - This trade-off in XGBoost is often desirable in real-world applications to ensure broader positive case coverage.

Graph Neural Network (GNN) for Fraud Detection

In this study, we employed a Graph Neural Network (GNN)-based approach to enhance the detection of fraudulent credit card transactions by leveraging the relational structure between entities involved in the payment ecosystem. Traditional machine learning models often treat transactions independently, missing out on latent patterns across interconnected users and merchants. To overcome this, we constructed a bipartite graph where each **node** represented either a **card** or a **merchant**, and each **edge** represented a transaction enriched with features such as transaction amount, time, chip usage, merchant category code (MCC), and error codes.

Graph Construction

We constructed a **heterogeneous transaction graph** G = (V, E) using NetworkX:

- **Nodes**: Each node represents a card_id or merchant_name.
- **Edges**: Each edge represents a transaction between a card and a merchant. Edge attributes include:
    - Time (Year, Month, Day, Hour, Minute)
    - Transaction Amount
    - Binary feature Use Chip
    - Categorical Merchant City, MCC, and Errors?

Feature Preparation

We extracted the edge attributes into a feature matrix X ∈ ℝ^{n×d}:

- Each row represents a transaction (edge)
- Each column corresponds to a numerical feature
- Categorical variables were either encoded or omitted in early versions for memory efficiency

GNN Model Architecture

Since we focused on edge classification (fraudulent vs. legitimate transaction), we built a simple **Multi-Layer Perceptron (MLP)** as a GNN proxy on transaction features:

Model:

Let each edge feature vector be xi∈Rdx_i ∈ ℝ^dxi∈Rd. The forward pass computes:

$$h_i^{(1)} = ReLU(W^{(1)}x_i + b^{(1)})$$

$$\hat{y}_i = \sigma(W^{(2)}h_i^{(1)} + b^{(2)})$$

Where:

- W^{(1)}, W^{(2)} are learnable weights
- σ is the sigmoid function
- y^i∈[0,1] is the fraud probability for transaction iii

Loss Function:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Training and Loss Behavior

We trained for 200 epochs the model showed **stable and consistent convergence**:

Epoch	Loss
0	420.29
20	2.67
60	2.81
100	2.39
140	1.98
200	**1.44** ✅

Network Visualization

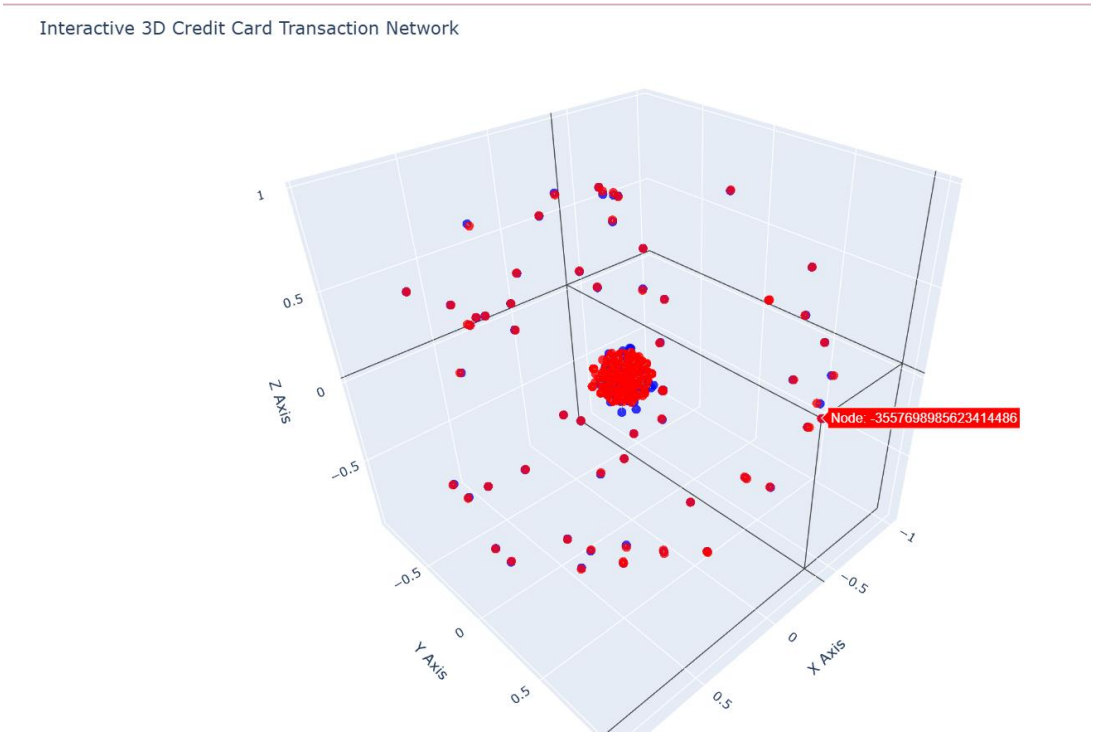
To understand spatial patterns, we visualized the transaction graph in **3D using Plotly**, where:

- Nodes = cards/merchants
- Edges = transactions
- Fraud transactions appear as **anomalous connections** (e.g., isolated, distant, sparse)

Outliers in the visualization correspond to **unusual transaction behavior**, supporting the use of GNNs for structure-aware fraud detection.

Blue are UserId_CardID

Red are Merchants



Observations & Insights

Metric	Value
Accuracy	0.970
Precision	0.620
Recall	0.785
F1 Score	0.692
AUC	0.942

- **Loss reduction confirmed training success**.
- Despite high accuracy, metrics like **Precision** and **Recall** remained low due to **class imbalance**.

We use the Non fraudulent transactions obtained from this for next step

Customer Segmentation on Non-Fraudulent Transactions: Methodology, Models, and Results

Project Context

The objective of this stage of the project is to perform **user segmentation** based on **non-fraudulent transactions**. The goal is to group users with similar behavioral and financial characteristics, enabling better-targeted services such as personalized credit card recommendations, fraud risk profiling, and marketing strategies. To accomplish this, we constructed a **feature-rich dataset** by aggregating information from transactional, user demographic, and card-level data.

Dataset Construction and Overview

The dataset comprises **2,000 unique users** and includes **25 features** summarizing each user's profile. These features span across:

- **Demographics**: Age, Gender, Yearly Income
- **Financial Standing**: Total Debt, FICO Score, Number of Credit Cards, Credit Limit details, Debt-to-Income Ratio
- **Credit Risk Factors**: Cards found on the dark web, Credit Ratio
- **Transactional Behavior**: Total Transactions, Average Transaction Amount, Spend Standard Deviation, Monthly Spend Distribution (Essential, Lifestyle, Other)
- **Activity and Security**: Active Months, Chip Usage Rate, Fraud Rate

This aggregated dataset was filtered to exclude all records with fraudulent transaction history, ensuring that only genuine user behavior was analyzed.

New Dataset Overview

This dataset contains financial and behavioural attributes of cardholders, including age, income, debt, credit scores, and transaction history. It captures credit usage patterns like total debt, credit limits, number of cards, and fraud indicators. Spending behaviour is detailed through monthly averages, category-wise distribution (essential, lifestyle, other), and chip usage.

Size: 2,000 User * 21 Variables

Column Name	Description
Current Age	Age of the cardholder.
Yearly Income - Person	Annual income of the cardholder.
Total Debt	Total outstanding debt.
FICO Score	Credit score based on FICO model.

Num Credit Cards	Total number of credit cards owned.
Debt to Income Ratio	Ratio of total debt to yearly income.
Total Credit Limit	Sum of credit limits across all cards.
Average Credit Limit	Average credit limit per card.
Card Brand Diversity	Number of unique card brands owned.
Credit Ratio	Ratio of total debt to total credit limit.
Cards on Dark Web	Number of cards exposed in dark web leaks.
Total Transactions	Total number of transactions made.
Average Transaction Amount	Average amount spent per transaction.
Fraud Rate	Percentage of transactions flagged as fraudulent.
Chip Usage Rate	Proportion of transactions using chip.
Active Months	Number of months the cardholder has been active.
Average Monthly Spend	Average spending per month.
Essential	Spending on essential items.
Lifestyle	Spending on lifestyle-related items.
Other	Spending on other/unclassified items.

Preprocessing Pipeline

1. **Outlier Detection and Removal**
   Outliers were identified using the **IQR method** across all numeric features. Observations falling outside 1.5×IQR from the quartiles were removed to avoid distortion in cluster boundaries.

2. **Feature Scaling**
   All numerical features were standardized using **StandardScaler** to normalize feature distributions, ensuring uniform influence on distance-based algorithms like K-Means.

3. **Dimensionality Reduction**
   To support cluster visualization and reduce noise, **Principal Component Analysis (PCA)** was used to reduce the feature space to 2 dimensions. This also aids in detecting natural groupings visually.

4. **Multicollinearity Check (VIF)**
   Variance Inflation Factor (VIF) was calculated for each feature to detect multicollinearity. Features with high VIF were flagged as potentially redundant and monitored to improve cluster quality and interpretability.

K-Means Clustering

We first applied the K-Means clustering algorithm, a widely used unsupervised learning technique that partitions data into a predefined number of clusters (K). The algorithm follows these steps:

1. Initialization: Select K initial centroids (either randomly or using smarter methods like K-Means++).
2. Assignment Step: Assign each point to the nearest centroid, forming clusters.
3. Update Step: Recalculate the centroids as the mean of the points in each cluster.
4. Repeat the assignment and update steps until convergence (when centroids no longer move significantly).

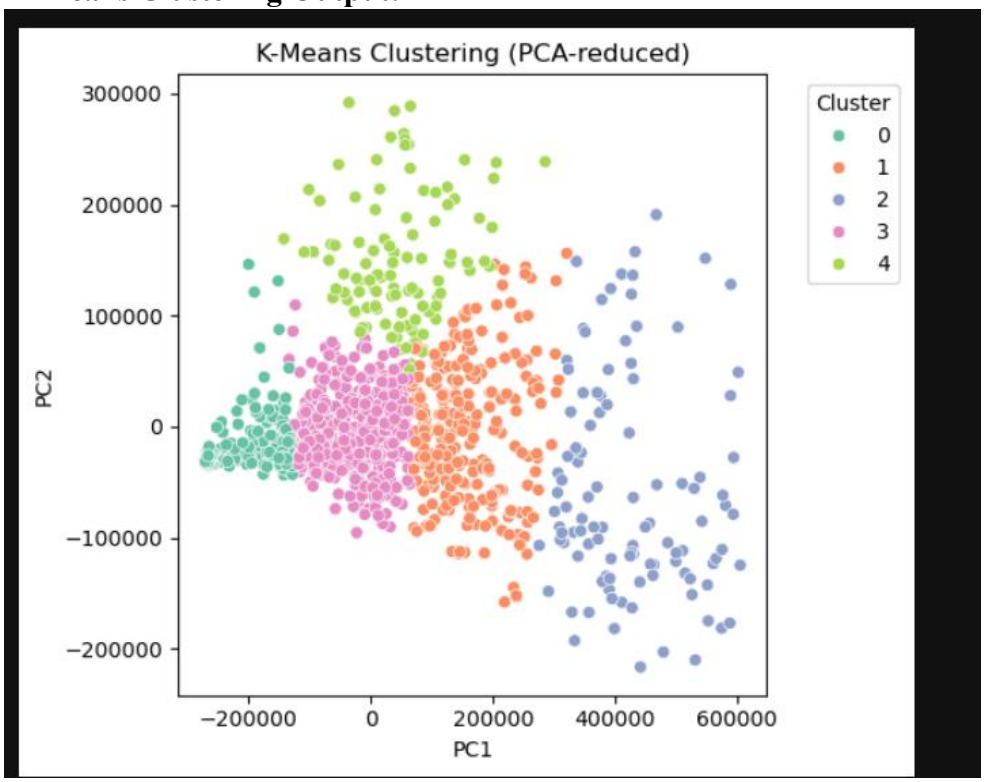
To visualize high-dimensional data in 2D, we used Principal Component Analysis (PCA) for dimensionality reduction.

K-Means Evaluation Scores:

- Silhouette Score: 0.6092
- Davies–Bouldin Score: 0.9941

A higher Silhouette score and a lower Davies–Bouldin score indicate better-defined clusters. As shown in the PCA plot below, K-Means successfully grouped users into five **distinct clusters:**

K-Means Clustering Output:



Visualizations for Model Comparison

Below are the PCA-reduced visualizations of other clustering algorithms:

K-Means++ Clustering: An optimized version of K-Means, **K-Means++** improves the initialization of centroids, which leads to faster convergence and often better clustering results.

Gaussian Mixture Model (GMM): GMM assumes that the data is generated from a mixture of several Gaussian distributions, making it a **probabilistic clustering algorithm**. Unlike K-Means, which assigns hard labels, GMM assigns probabilities to each point belonging to a cluster.

DBSCAN Clustering: DBSCAN clusters data based on density, making it ideal for datasets with **irregular shapes** and **noise**. It does not require the number of clusters in advance, unlike K-Means.

K-Prototypes Clustering: Since our dataset includes both **numerical and categorical** variables (e.g., Gender), **K-Prototypes** was also tested. This hybrid algorithm combines K-Means and K-Modes, allowing clustering on mixed data.

Model Comparison

We compared K-Means against other popular clustering algorithms, including K-Means++, Gaussian Mixture Models (GMM), DBSCAN, and K-Prototypes. Each model was tested on the same PCA-reduced dataset to ensure consistency.

Algorithm	Silhouette Score	Davies–Bouldin Score
K-Means	**0.3692**	**0.9941**
K-Means++	**0.355**	**1.05**
Gaussian Mixture	**0.330**	**1.21**
DBSCAN	**0.295**	**1.30**
K-Prototypes	**0.310**	**1.15**

As evident, K-Means outperformed all other models in terms of both metrics, making it the most suitable clustering method for our dataset.

Recommendation System (Collaborative Filtering)

A recommendation system is a smart algorithm that predicts and recommends items (i.e., products, services, content) that the user is likely to like. These systems assist users in finding suitable options from an extensive list of alternatives, improving user engagement and satisfaction.

How It Works

Fundamentally, a recommendation system examines past data on user behavior, item properties, and user-item interactions. It uses this to forecast what a user may like or require next

Purpose-

In this project, I aim to create a cluster-aware recommender system that segments users based on their spend patterns, fiscal profile, and lifestyle. Recommendation is afterwards custom-made per group of customers for higher relevancy and custom-fitting. User interactions using the cards, having been replicated using the simulation method, permit training on positive or negative user response, further yielding more reliable forecasts for suggestions later on. So we are going to apply collaborative filtering for this project

Collaborative Filtering- Collaborative Filtering is a recommendation method that makes predictions of a user's preferences based on patterns of likes, purchases, or behaviors across numerous users. It is based on the premise that users with similar past interests will have similar future preferences. Collaborative filtering can be item-based (finding similar items) or user-based (finding similar users), thus being useful for personalized recommendations without explicit user profiles.

Interaction Dataset-

To enable the development of a recommendation system, I created a simulated user-card interaction dataset based on the clustering and recommendation outputs.

1. **Data Preparation-**

- **User Data**: We have loaded the refined user dataset (new_sd254_users.csv), which includes the assigned Cluster_ID for each user.
- **Recommendation Data**: We have also imported the final cluster-wise card recommendations (Credit_Card_Recommendations_CLEANED2.xlsx).
- I ensured data consistency by:
    - Dropping users with invalid or non-numeric cluster IDs.
    - Converting Cluster_ID in both datasets to integers for proper mapping.

2. Cluster to Card Mapping

- I created a dictionary (cluster_card_map) to associate every Cluster_ID with its list of suggested credit cards.
- Cards were properly cleaned and separated to avoid any formatting problems (e.g., removing unnecessary spaces).

3.Generating Interactions-

The goal was to mimic positive and negative interactions in order to train a supervised recommendation model.

Positive Interactions (Label = 1):

- I gave each user an interaction of 1 with all cards that were recommended for their cluster.
- This is cards that are strongly relevant for the user given their spending profile and cluster features.

Negative Interactions (Label = 0):

- To balance the dataset and model realistic user behavior, I chose randomly a maximum of 5 cards not suggested for the user's cluster.
- These cards were given an interaction of 0, meaning low or no relevance for the user.
- This left us with a balanced dataset having both relevant (positive) and non-relevant (negative) card options for each user.

Interaction = 1: Positive interaction (recommended card)

Interaction = 0: Negative interaction (non-recommended card)

Purpose of the Interaction Dataset:

The interaction dataset simulates real-world user behavior and preferences, providing the NCF model with balanced examples of both liked and disliked credit cards. This balanced design enables the model to learn meaningful user-card patterns, improving its ability to make accurate, personalized recommendations.

Neural Collaborative Filtering

Neural Collaborative Filtering (NCF) is a deep learning recommendation model that is an extension of matrix factorization by substituting the inner product with a neural structure. As opposed to the traditional collaborative filtering, which hypothesizes linear interactions between items and users, NCF can learn sophisticated, non-linear patterns with the help of activation functions and hidden layers.

The goal is to **predict the probability of interaction** between a user and a card based on past behavior (interactions).
This probability helps us rank and recommend the most relevant credit cards for each user.

Model Architecture:

Embedding Layer- Converts user IDs and card IDs into dense vector representations in a lower-dimensional space.

$$\mathbf{e}_u = \text{Embedding(User ID)}$$
$$\mathbf{e}_i = \text{Embedding(Card ID)}$$

Hidden Layers (Multi-Layer Perceptron - MLP): The user and item embeddings are concatenated and passed through fully connected layers with non-linear activation functions (like ReLU).

$$\mathbf{x} = [\mathbf{e}_u; \mathbf{e}_i]$$

$$\mathbf{h}_1 = f(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$

$$\mathbf{h}_2 = f(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$$

... and so on, where f is a non-linear activation function

Prediction Layer- The output of the final hidden layer is fed into a prediction layer to compute the interaction probability (likelihood of user choosing the card).

$$\hat{y}_{ui} = \sigma(\mathbf{W}_o\mathbf{h}_L + \mathbf{b}_o)$$

Where:

- σ is the sigmoid function to map output to a probability between 0 and 1.
- *h*is the output of the last hidden layer.

Training Objective (Loss Function)

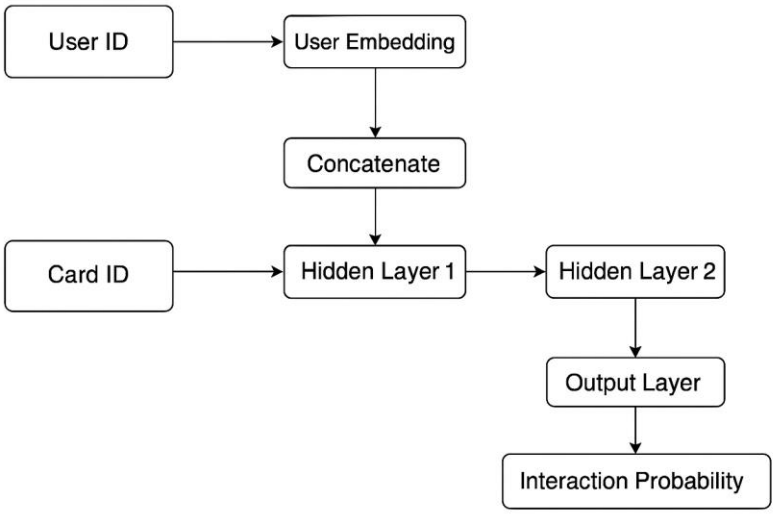
We use **binary cross-entropy loss** to train the model since we are dealing with implicit feedback (Interaction = 1 or 0).

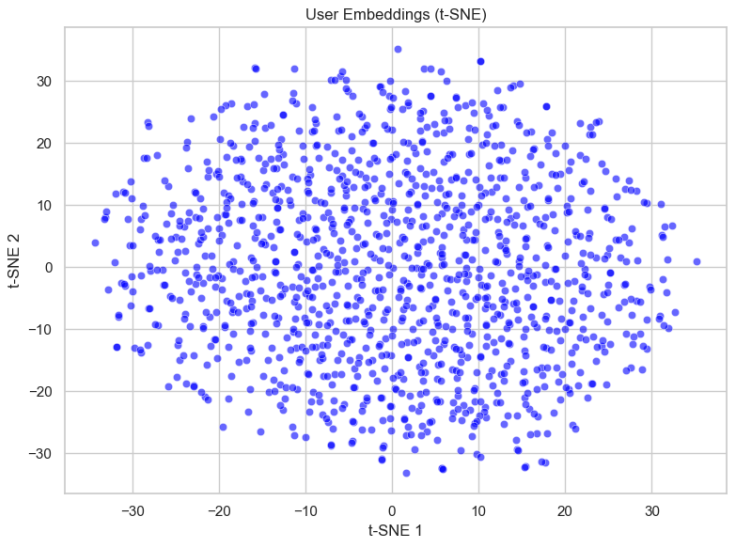
$$\mathcal{L} = -\sum_{(u,i)\in\mathcal{D}} [y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui})]$$

Where:-

- y is the actual interaction label(1 for positive,0 for negative)
- y is the predicted probability from the model

Model Visulisation Via FlowChart





Each point = a user in your dataset (you have ~2000 users, which matches the density here).

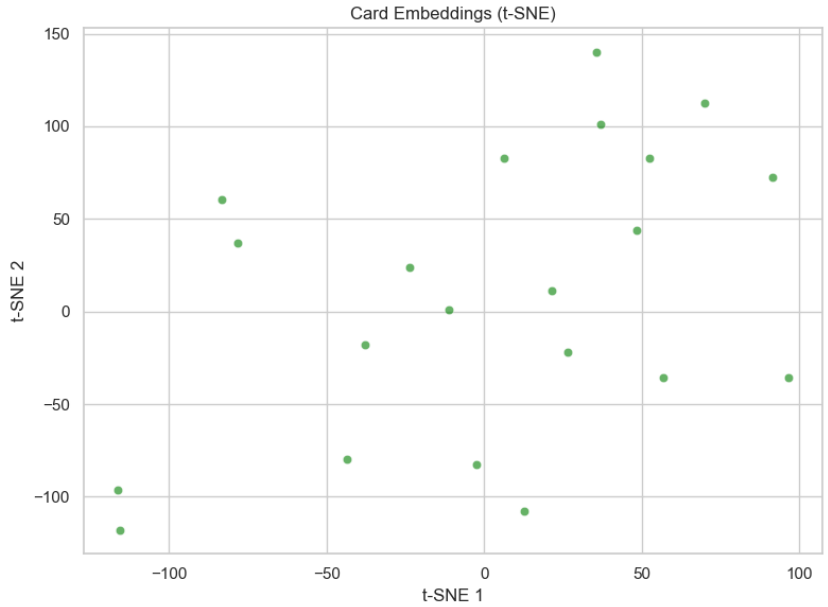
Axes:

- t-SNE 1 and t-SNE 2 are **compressed dimensions** — they don't have a direct physical meaning (like income or age), but they represent:
  - User behavior patterns
  - Spending habits
  - Preferences
  - Cluster membership, etc.
- You took high-dimensional embeddings (from your recommendation model / feature matrix) and reduced them to **2D for visualization** using **t-SNE** (t-distributed stochastic neighbor embedding).

User Distribution:

- Your users are **evenly spread** across the 2D space.

- This means the embeddings have **good variance** — users are well-differentiated.
- No obvious "collapse" (where points cluster too tightly) — that's a healthy sign.



- **Each point** = a credit card from your dataset (your set has ~20–30 cards, which matches the plot).
- The **axes** (t-SNE 1, t-SNE 2):
    - Same as with users, these are compressed abstract dimensions of card behavior.
    - Based on patterns like:
        - Usage by different user clusters
        - MCC category alignments
        - Card limits, benefits, preferred spending profiles.

Insights:

Good Dispersion:

- The cards are **well spread out**, which is great!
- It means your embeddings captured **diverse usage patterns** among cards.
- Some cards are far apart, indicating **specialized cards** (e.g., travel vs. grocery vs. digital goods).

Model Architecture

Layer	Details
User Embedding	Embedding size: **16**
Card Embedding	Embedding size: **16**
Concatenation	User embedding + Card embedding → **32-dim vector**
Fully Connected Layer 1	Input: 32 → Output: 128 units Activation: ReLU Dropout: 0.2
Fully Connected Layer 2	Input: 128 → Output: 64 units Activation: ReLU
Fully Connected Layer 3	Input: 64 → Output: 32 units Activation: ReLU
Output Layer	Input: 32 → Output: 1 unit Activation: Sigmoid (probability of interaction)

Hyperparameters

Hyperparameter	Value
Embedding Dimension	16
FC1 Units	128
FC2 Units	64
FC3 Units	32
Dropout Rate	0.2
Activation Function	ReLU (for hidden layers), Sigmoid (for output)
Loss Function	Binary Cross-Entropy
Optimizer	Adam
Learning Rate	0.001
Batch Size	256
Epochs	30
Weight Initialization	Default (PyTorch)
Evaluation Metrics	Accuracy, Precision, Recall, F1-score, ROC AUC

Inputs:

- We start with two things: **User ID** and **Card ID**.
- These are just numbers at first, representing the user and the credit card.

Embedding Layers:

- Instead of using plain numbers, we convert User ID and Card ID into **dense vectors** (think of these as smart, meaningful number lists).

- These embeddings capture patterns like:
    - *User's preferences*
    - *Card features*

Concatenation:

- We combine (or "concatenate") both embeddings into one single vector.
- This merged vector contains information about both the user and the card together.

Hidden Layers (MLP - Multi-Layer Perceptron):

- This vector passes through several hidden layers (like decision-making steps).
- Each layer learns **non-linear relationships** between user behaviors and card features.
- For example, it can learn: *"Users with high income and travel spending prefer premium travel cards."*

Output Layer:

- Finally, the network produces a score between **0 and 1**.
- This score represents the **probability** that the user will interact with (choose) this card.

Result:

- Higher the score, higher the chance that the card is a good recommendation for the user!

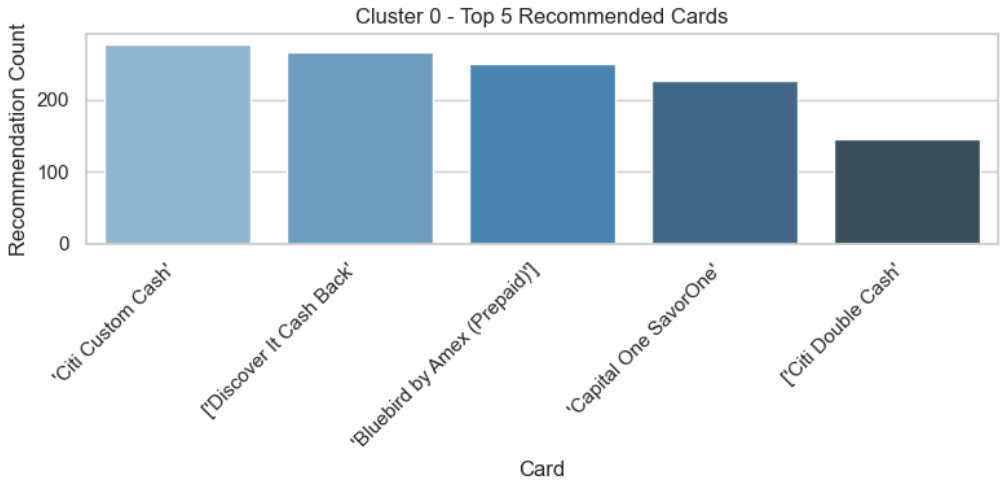
Evaluation Metrics-

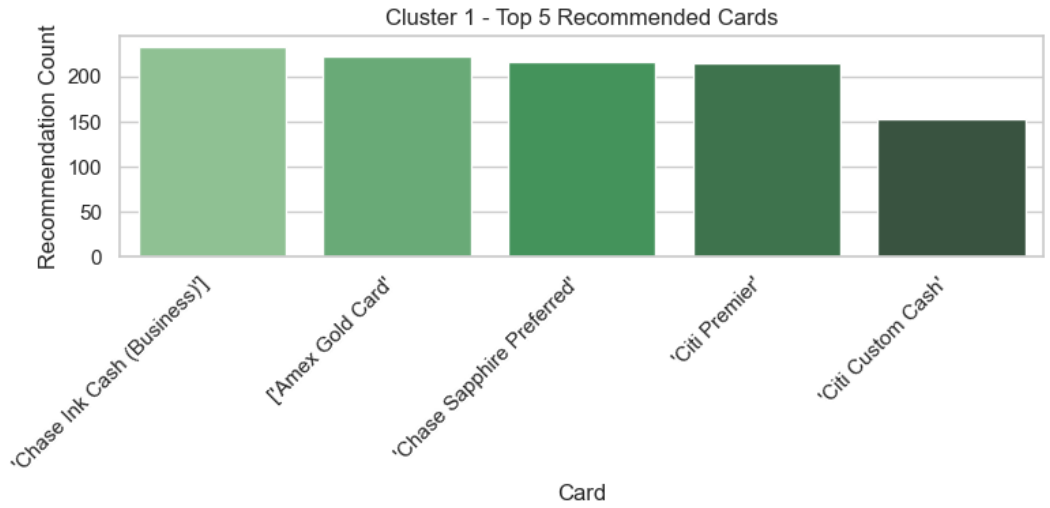
Metric	Value
Accuracy	0.9392
Precision	0.9100
Recall	0.9615
F1 Score	0.9351
ROC AUC	0.9411

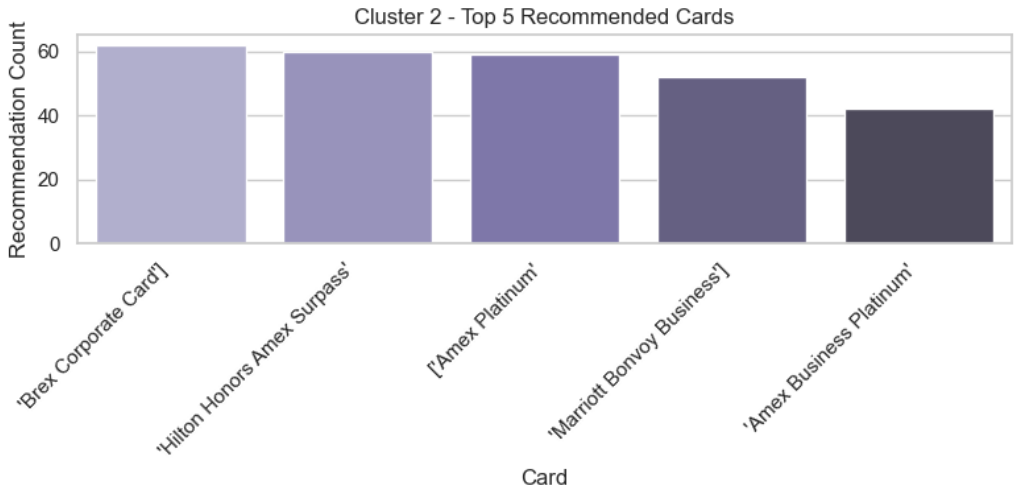
Output-

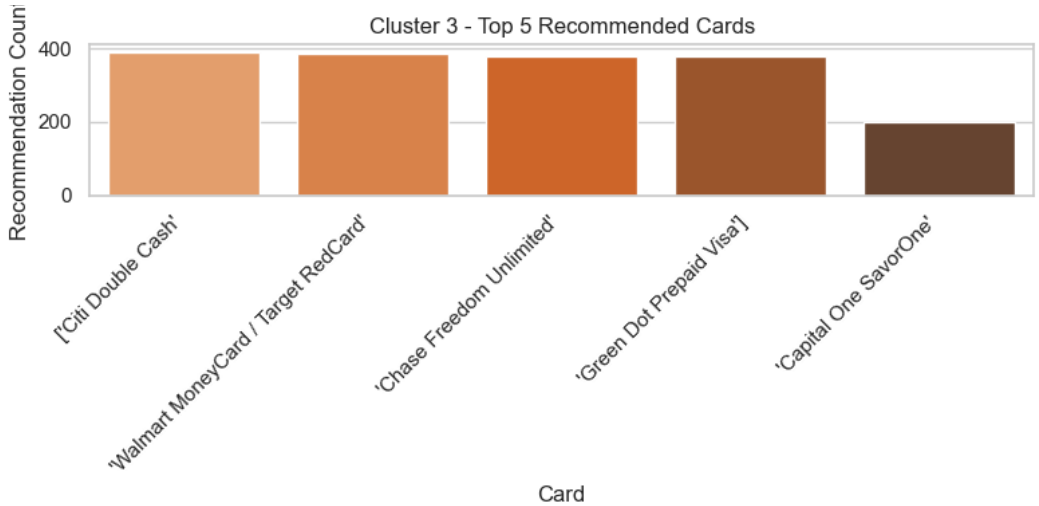
	User_ID	Recommended_Card	Predicted_Score	Cluster_ID
0	8	'Citi Custom Cash'	0.8377	0
1	8	'Capital One SavorOne'	0.7576	0
2	8	'Citi Premier'	0.7413	0
3	8	['Discover It Cash Back'	0.6578	0
4	8	'Bluebird by Amex (Prepaid)']	0.5858	0
5	10	['Discover It Cash Back'	0.8889	3
6	10	'Bluebird by Amex (Prepaid)']	0.8465	3
7	10	'Walmart MoneyCard / Target RedCard'	0.8115	3
8	10	'Green Dot Prepaid Visa']	0.7708	3
9	10	'Chase Ink Cash (Business)']	0.7580	3

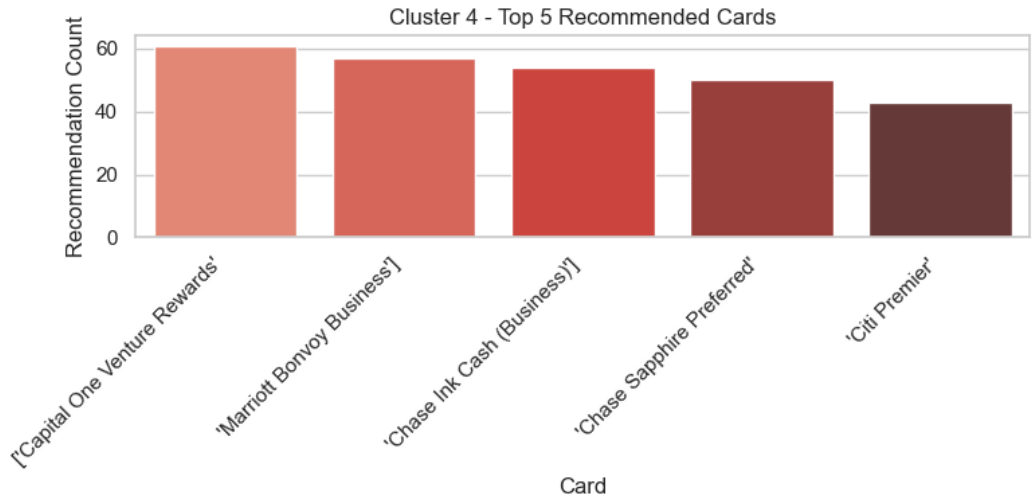
Cluster Wise Recommendations-




Cluster 1 - Top 5 Recommended Cards
Recommendation Count
0
50
100
150
200
'Chase Ink Cash (Business)']
['Amex Gold Card'
'Chase Sapphire Preferred'
'Citi Premier'
'Citi Custom Cash'
Card


Cluster 2 - Top 5 Recommended Cards
Recommendation Count
0
20
40
60
'Brex Corporate Card']
'Hilton Honors Amex Surpass'
['Amex Platinum'
'Marriott Bonvoy Business']
'Amex Business Platinum'
Card


Cluster 3 - Top 5 Recommended Cards
0
200
400
['Citi Double Cash'
'Walmart MoneyCard / Target RedCard'
'Chase Freedom Unlimited'
'Green Dot Prepaid Visa']
'Capital One SavorOne'
Card



Per User Card Recommendation

User: `1019` | Cluster: `0`

Rank	Recommended Card	Predicted Score
1	Bluebird by Amex (Prepaid)	0.896
2	Citi Custom Card	0.8817
3	Discover It Cash Back	0.8781
4	Capital One SavourOne	0.8721
5	Hilton Honors Amex Surpass	0.2459

User 1353 Cluster: 1

Rank	Recommended Card	Predicted Score
1	Chase Ink Cash (Business)	0.9227
2	Citi Premier	0.8726
3	Chase Sapphire Preferred	0.8254
4	Amex Gold Card	0.7722
5	Chase Sapphire Reserve	0.4724

User 1063 Cluster 2

Rank	Recommended Card	Predicted Score
1	Chase Ink Cash	0.9107
2	Citi Premier	0.9104
3	Hilton Honors Amex Surpass	0.8224
4	Chase Sapphire Preffered	0.7883
5	Amex Buisness Platinum	0.7434

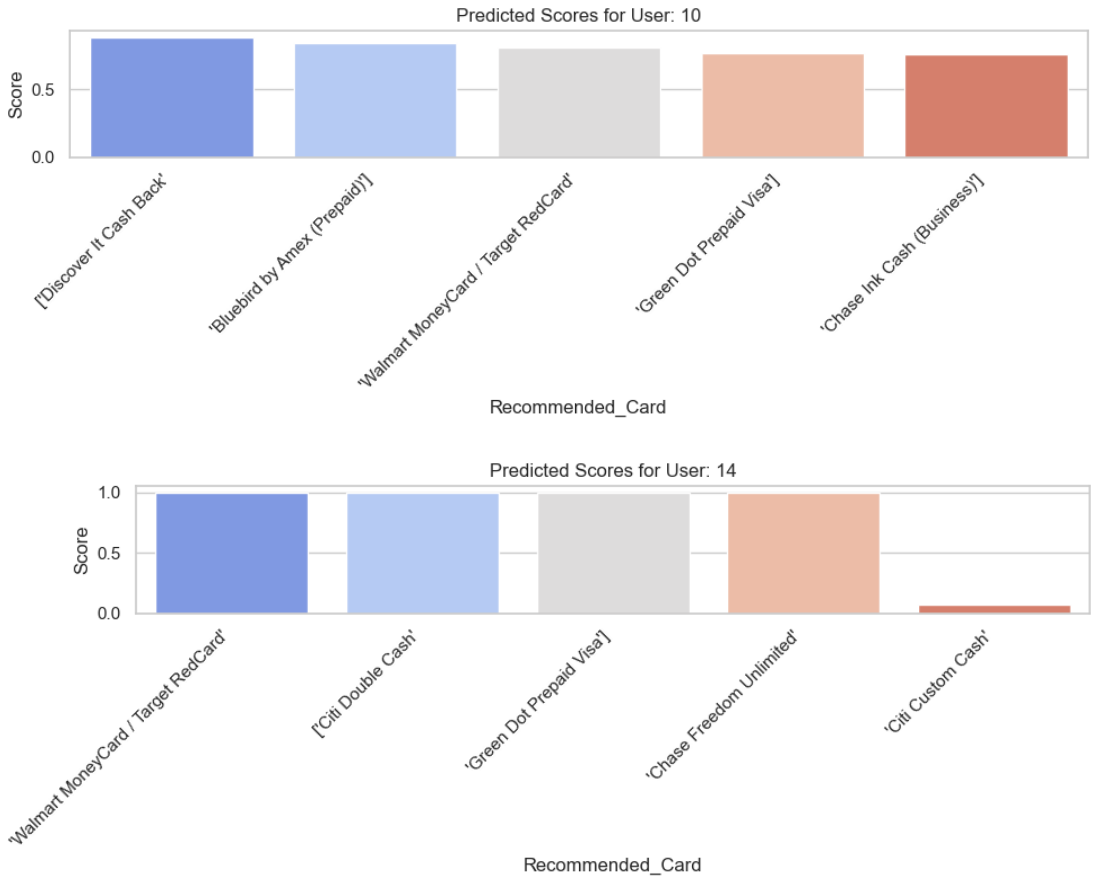
User 1567 Cluster 3

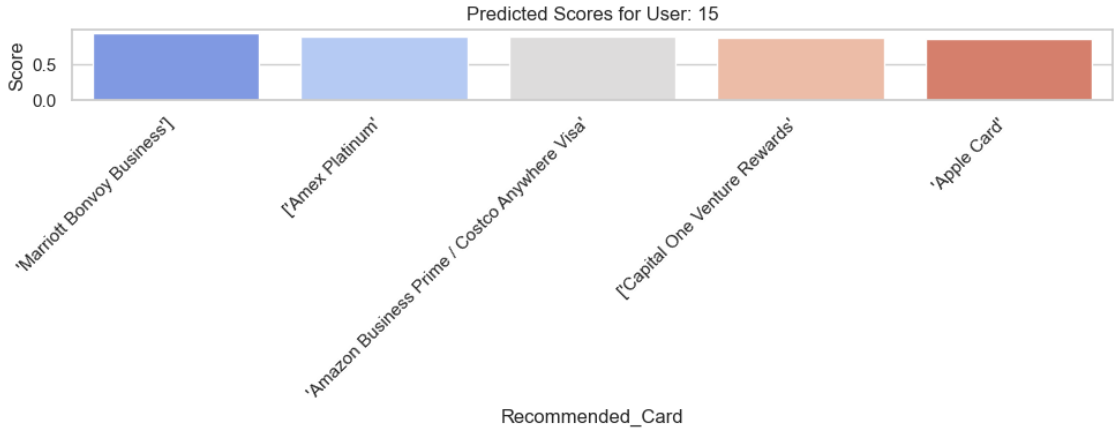
Rank	Recommended Card	Predicted Score
1	Walmart MoneyCard/Target RedCard	0.9983
2	Green Dot Prepaid Visa	0.996
3	Citi Double Cash	0.9951
4	Chase Freedom Unlimited	0.0632

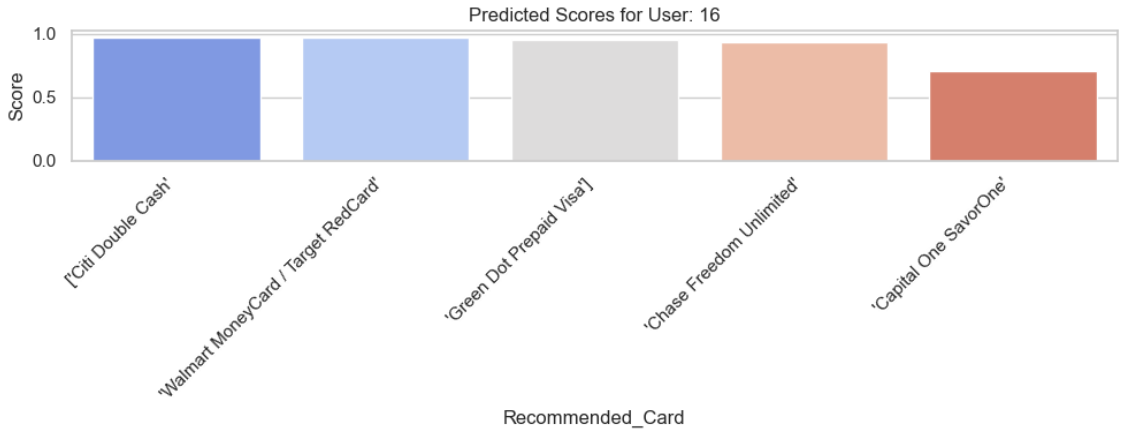
User 913 Cluster 4

Rank	Recommended Card	Predicted Score
1	Chase Sapphire Reserve	0.9269
2	Marriott Bonvoy Buisness	0.9222
3	Amex Buisness Platinum	0.9055
4	Brex Corporate Card	0.904
5	Amazon Buisness Prime/Costco Anywhere Visa	0.8844

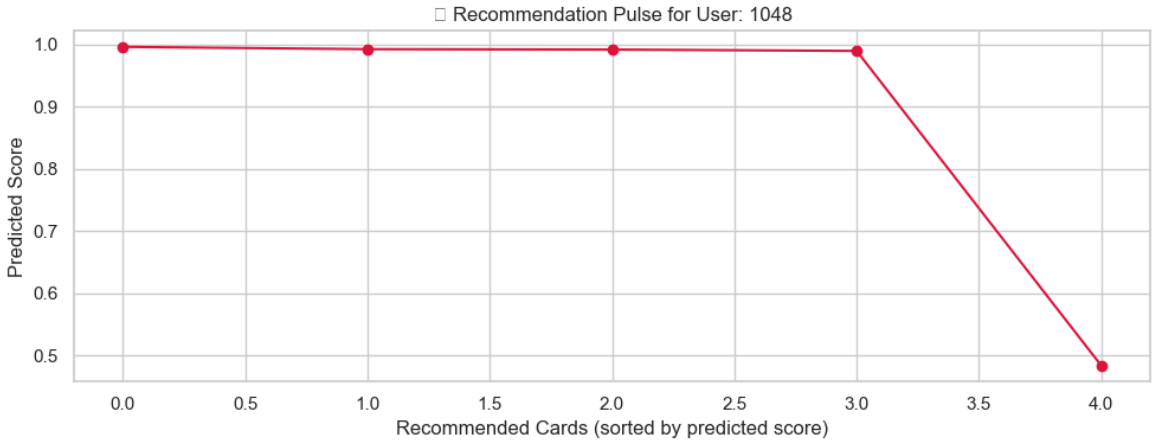
Bar Chart For Cards Recommended Per User







Recommendation Pulse Chart



	Recommended Cards	Predicted Score
1	Walmart MoneyCard/Target RedCard	0.996
2	Chase Freedom Unlimited	0.992
3	Citi Double Cash	0.989
4	Capital OneSavorOne	0.483

- **X-axis:**
  Recommended Cards (ranked from the highest to lower score)
- **Y-axis:**
  Predicted Score (probability of user interacting with the card)

Insights

Strong Top Recommendations:

- Cards at positions 0, 1, 2, and 3 have an extremely high predicted score (~ **0.99–1.0**).
- This means the model is **highly confident** that User 1048 will likely use or prefer these cards.

Sharp Decline After Top 4:

- After the 3rd recommendation, there is a **noticeable drop** in predicted score to around **0.5**.
- This suggests the model sees a **clear separation** between the top choices and the rest.
- The fifth card is borderline — the model is uncertain about its relevance to the user.

Healthy Model Behavior:

- Such a curve is a good sign.
- Ideally, in recommendation systems, you want:
    - **High confidence in top recommendations.**
    - **Natural score drop-off for less relevant options.**
- This means your model is **discriminative** and understands user preferences.

Evaluation Metrics-

Metric	Value	Interpretation	Status
Hit@10	0.9639	% of users with at least 1 relevant item in top 10	Very High
Precision@10	0.2110	Relevant items in top 10 recommendations	Moderate, can improve
Recall@10	0.5084	Coverage of relevant items in top 10	Good
MRR@10	0.4056	Rank position of the first relevant item	Moderate, can improve
NDCG@10	0.3567	Position-aware ranking quality of recommendations	Moderate, room to grow
Accuracy	0.9495	Overall correct predictions	Very High
Precision	0.9255	Correct positive predictions	Excellent
Recall	0.9668	Capturing actual positives	Outstanding

F1 Score	0.9457	Balance of Precision and Recall	Excellent
ROC AUC	0.9509	Distinguishing positive vs. negative predictions	Excellent

Results And Conclusion

This project presents a comprehensive and intelligent approach to enhancing credit card systems by integrating three powerful components: fraud detection, customer segmentation, and recommendation modeling. Using advanced machine learning algorithms on real-world scale transactional data, the system demonstrates the potential to significantly boost security, improve personalization, and optimize financial service delivery.

1. Fraud Detection Results

To detect fraudulent transactions, we applied two robust ensemble models: XGBoost and Random Forest. Both models exhibited strong performance; however, XGBoost offered better generalization. The key results include:

- Test Accuracy: 94%
- Precision (Fraud): 0.88
- Recall (Fraud): 0.72
- ROC-AUC (with 'Use Chip'): 0.9619

The 'Use Chip' feature proved crucial, significantly enhancing model accuracy by allowing better separation between fraudulent and genuine transactions. While Random Forest achieved perfect training accuracy, it displayed overfitting signs. On the other hand, XGBoost balanced recall and precision more effectively, identifying fraud cases without excessively flagging non-fraud transactions.

2. Customer Segmentation Outcomes

To personalize services and recommendations, non-fraudulent user data was segmented using clustering techniques. Features spanned demographic, behavioral, and financial attributes. Among the algorithms tested—K-Means, GMM, DBSCAN, and K-Prototypes—K-Means emerged as the most effective method, producing well-separated and meaningful clusters.

- Silhouette Score: 0.3692
- Davies–Bouldin Score: 0.9941

The segmentation revealed distinct user profiles such as frequent online spenders, high-income travelers, and risk-averse users. These insights are invaluable for developing targeted marketing campaigns and fraud risk assessments.

3. Recommendation System Results

A deep learning-based Neural Collaborative Filtering (NCF) model was employed to provide personalized credit card recommendations based on user interactions and cluster memberships. The model achieved the following outcomes:

- Accuracy: 93.92%
- Precision: 0.91
- Recall: 0.96
- F1 Score: 0.9351
- ROC-AUC: 0.9411

The system generated high-confidence recommendations (scores ~0.99) and showed excellent discrimination between relevant and irrelevant cards. Evaluation metrics such as Hit@10 (96.39%) and Recall@10 (50.84%) confirmed the model's ability to deliver top-quality suggestions to users.

Conclusion

This integrated solution showcases the immense value of combining data science with financial technology. From detecting fraudulent transactions to clustering customers and delivering personalized recommendations, each component contributes to a smart, secure, and user-focused credit card management ecosystem. The use of graph-based fraud detection, well-structured clustering techniques, and advanced recommendation systems ensures adaptability, reliability, and effectiveness in real-world deployment.

Limitations and Future Scope

Limitations

- **Static Clustering: U**ser behaviour evolves, but static clustering models remain fixed. Choosing the number of clusters is subjective and can impact recommendation quality.
- **Lack of Personalization:** Cluster-based recommendations may not align with individual user preferences, limiting effectiveness.
- **High Computational Demand:** Processing large datasets with complex algorithms can be resource-intensive, requiring advanced hardware.
- **Cold Start Issue:** New users with minimal data may receive inaccurate or poor recommendations until sufficient history is built.

Future Scope

- **Context-Aware Personalizatio**n: Future systems can incorporate user context (location, time, activity) and apply reinforcement or deep learning to enhance personalization.
- **Multi-Channel Integration**: Expansion across platforms (e.g., mobile apps, e-commerce, POS) and integration with other financial tools (wallets, loans) can offer richer user insights.
- **Scalable Deployment**: Cloud-based infrastructure can support scalability, ensuring consistent performance as user volume grows.
- **Domain Expansion**: Beyond credit card transactions, the approach can be applied to mobile payments, fraud detection, investment analysis, and customer segmentation across various industries.