FinRec

Fraud-Aware Credit Card Recommendation System

IE7275 – Data Mining in Engineering | Spring 2026

Group 9 | Project 1 – Recommendation Systems

Team Member	Role
Nagashree Bommenahalli Kumaraswamy	Airflow Orchestration, Modeling & Validation
Hemanth Mareedu	Data Pipeline & EDA
Aditya Sai Kumar Kusetti	Model Development & Evaluation

Dataset: Kaggle – Credit Card Transactions Dataset

Source: kaggle.com/datasets/priyamchoksi/credit-card-transactions-dataset

Submission Date: March 1, 2026

Table of Contents

1. Dataset Description & Preprocessing

1.1 Dataset Overview

The Credit Card Transactions Dataset was obtained from Kaggle (priyamchoksi, 2024). It contains real-world credit card transaction records representing customer spending behavior across different merchants, categories, locations, and time periods. The dataset comprises **1,296,675 rows** and **22 columns**, with a file size of approximately 121 MB. This implicit feedback dataset — where transaction frequency and amount serve as preference signals — is well-suited for building a personalized fintech recommendation system.

1.2 Key Attributes

Column	Type	Description
trans_date_trans_time	Datetime	Full timestamp of transaction
cc_num	String	Customer credit card number (unique customer ID)
merchant	String	Merchant name
category	String	Transaction category (13 unique values)
amt	Float	Transaction amount in USD
first / last	String	Cardholder first and last name
gender	String	Cardholder gender (M/F)
city / state / zip	Mixed	Cardholder location details
lat / long	Float	Cardholder geographic coordinates
city_pop	Integer	Population of cardholder's city
job	String	Cardholder occupation
dob	Date	Cardholder date of birth
trans_num	String	Unique transaction identifier
merch_lat / merch_long	Float	Merchant geographic coordinates
is_fraud	Integer	Target label: 1 = Fraud, 0 = Legitimate

1.3 Preprocessing Steps

- **Datetime Parsing:** Extracted hour, day, month, year, day-of-week, and quarter from trans_date_trans_time to enable temporal analysis.
- **Age Derivation:** Parsed date-of-birth (dob) and computed customer age relative to January 1, 2020; clipped to [18, 100] for sanity.
- **Missing Value Handling:** Numerical columns filled with column median; categorical columns filled with column mode. No column had more than 0.1% missing values.

- **Duplicate Removal:** Deduplicated on trans_num (unique transaction ID). Zero duplicates found in the dataset.
- **Outlier Treatment:** Transaction amount (amt) capped at the 99.5th percentile ($844.22) to reduce the influence of extreme fraud-driven outliers.
- **Categorical Encoding:** Applied LabelEncoder to category, state, and gender for model compatibility.
- **Normalization:** MinMaxScaler applied to amt_capped to produce amt_norm in [0, 1]. StandardScaler applied to behavioral customer features for clustering.
- **Customer Identifier:** cc_num cast to string and used as customer_id throughout.

1.4 Train / Validation / Test Split

A **temporal split** was used — the most realistic strategy for transaction data — where the dataset is sorted by transaction timestamp and split as follows:

Split	Proportion	Approx. Rows	Date Range
Train	70%	~907,000	Jan 2019 – ~Oct 2019
Validation	15%	~195,000	~Oct 2019 – ~Mar 2020
Test	15%	~195,000	~Mar 2020 – Dec 2020

Using a temporal split prevents data leakage (future transactions informing past recommendations) and reflects real-world deployment conditions where a model trained on historical data is evaluated on future interactions.

2. Exploratory Data Analysis (EDA)

2.1 Transaction Amount Distribution

The transaction amount distribution is heavily right-skewed. The median transaction is approximately **$47** while the mean is higher due to outlier fraud transactions. After capping at the 99.5th percentile ($844.22), the distribution becomes more manageable for model training. Fraudulent transactions tend to have significantly higher amounts compared to legitimate ones.

2.2 Class Imbalance: Fraud vs. Legitimate

The dataset exhibits significant class imbalance typical of real-world fraud data. Approximately **0.58%** of all transactions are fraudulent. This imbalance is important to consider in evaluation — accuracy alone is misleading. We use ranking-based metrics (Precision@K, NDCG@K) which are more informative for recommendation quality.

Class	Count	Percentage
Legitimate (is_fraud = 0)	~1,289,169	99.42%
Fraudulent (is_fraud = 1)	~7,506	0.58%
Total	~1,296,675	100%

2.3 Category-Level Analysis

The 13 transaction categories show distinct spending patterns. The table below summarizes key statistics per category (approximate values based on dataset analysis):

Category	Tx Count	Avg Amount ($)	Fraud Rate (%)
grocery_pos	High	~35	~0.4%
gas_transport	High	~50	~0.5%
food_dining	High	~25	~0.3%
shopping_pos	Medium	~90	~0.6%
misc_net	Medium	~70	~1.8%
shopping_net	Medium	~120	~1.5%
entertainment	Medium	~60	~0.7%
home	Low	~200	~0.4%
personal_care	Low	~40	~0.3%
health_fitness	Low	~80	~0.4%
kids_pets	Low	~45	~0.5%
travel	Low	~350	~2.1%
misc_pos	Low	~30	~0.4%

Key insight: travel and misc_net have the highest fraud rates despite lower transaction volumes. This motivates the fraud-penalty component in our Hybrid model.

2.4 User-Item Interaction Analysis

The user-category interaction matrix reveals important characteristics of the dataset:

- The dataset contains approximately **1,000 unique customers** and **13 categories**.
- Interaction sparsity is relatively low (most customers transact in multiple categories), which is favorable for collaborative filtering.
- Transaction counts per customer follow a power-law distribution — a small number of customers generate a disproportionate share of transactions.
- On average, each customer engages with **8–10 categories** out of 13, indicating broad spending behavior.

2.5 Temporal Spending Patterns

Analysis of transaction timestamps reveals actionable temporal patterns:

- **Hourly patterns:** Transaction volume peaks between 10am–2pm. Fraudulent transactions are slightly more concentrated in late-night hours (12am–4am).
- **Day-of-week:** Weekend spending is higher, particularly Saturday. Fraud rates are marginally elevated on weekends.
- **Monthly trends:** Spending shows seasonal variation with peaks in November–December (holiday shopping) and summer months.
- **Quarterly patterns:** Q4 dominates in shopping_net and shopping_pos categories; travel peaks in Q2 and Q3.

2.6 Geographic Analysis

Transaction volume is highest in populous states including California (CA), Texas (TX), Florida (FL), and New York (NY), consistent with their large population sizes. Geographic distance between customer and merchant coordinates was computed as an additional feature, with larger distances correlating weakly with higher fraud probability.

3. Feature Engineering

3.1 User-Category Interaction Matrix

The primary input to matrix factorization models is a **user-category interaction matrix** of shape (n_customers x 13). Each entry represents the normalized total spend of a customer in a given category. Row normalization ensures that high-spending customers do not dominate over moderate spenders.

3.2 Customer Behavioral Features

Twelve behavioral features were engineered per customer from training data for use in the clustering model:

Feature	Description
total_spend	Sum of all transaction amounts
avg_spend	Mean transaction amount
std_spend	Standard deviation of amounts
tx_count	Total number of transactions
unique_categories	Number of distinct categories visited
unique_merchants	Number of distinct merchants visited
max_single_spend	Maximum single transaction amount
fraud_tx_count	Number of flagged fraud transactions
fraud_rate	Proportion of fraudulent transactions
avg_hour	Average hour of day for transactions
weekend_ratio	Fraction of transactions on weekends
night_ratio	Fraction of transactions between 10pm–6am

3.3 Recency Weighting

For the Hybrid model, transaction recency was encoded using an exponential decay function: **w = exp(-days_since / 180)**, applying a 6-month half-life. More recent transactions receive higher weights, reflecting the intuition that recent behavior is a stronger signal of current preferences than old transactions.

4. Model Design & Rationale

4.1 Problem Framing

We frame the recommendation problem as: given a customer's transaction history, recommend the Top-K spending categories (or merchant offers) most relevant to that customer. Since we have no explicit ratings, we use **implicit feedback** — transaction frequency and spend amount serve as preference signals. All models generate a ranked list of K=5 category recommendations per customer.

4.2 Baseline Models

Baseline 1 — Global Popularity-Based

Recommends the globally most-popular categories (by transaction count) to every customer, regardless of their individual history. This is the simplest possible recommendation strategy and serves as a lower-bound benchmark. It has zero personalization but excellent cold-start handling and requires no training.

Baseline 2 — Frequency-Based (Personalized)

For each customer, recommends their most-frequently transacted categories from training data. Empty slots are filled with globally popular categories. This simple personalized baseline captures individual habits without any latent factor modeling. It represents a strong practical baseline for transaction data.

4.3 Advanced Models

Model 1 — SVD Matrix Factorization

Applies Truncated SVD (scipy.sparse.linalg.svds) to the normalized user-category interaction matrix with k=15 latent components. The reconstructed matrix yields predicted preference scores for all user-category pairs, including unobserved ones. SVD is a well-established collaborative filtering technique that captures latent user preferences and category characteristics. Cold-start users fall back to global popularity.

Model 2 — Non-Negative Matrix Factorization (NMF)

Applies NMF (sklearn.decomposition.NMF) with k=10 components and nndsvd initialization. NMF enforces non-negativity on both factor matrices, producing more interpretable part-based representations. The reconstructed score matrix is used directly for ranking. NMF tends to produce sparser, more focused recommendations than SVD.

Model 3 — User-Based Collaborative Filtering

Uses SVD latent factors as compressed user representations and computes cosine similarity between users in the latent space. For each target user, the top 10 most similar neighbors are identified, and their spending preferences are aggregated (weighted by similarity score) to generate category recommendations. This model is limited to the first 500 users for computational efficiency.

Model 4 — K-Means Clustering (Segment-Based)

Applies K-Means clustering (k=5 segments, selected via elbow method) on 12 standardized behavioral features. Each cluster represents a customer segment with similar spending behavior. Category recommendations are generated at the segment level — the top-K categories for each segment are recommended to all customers in that segment. This approach is highly scalable and interpretable.

Model 5 — Hybrid Fraud-Aware Recommender (Proposed)

Our novel contribution combines three signals with a fraud safety layer:

- **SVD Score (alpha=0.5):** Normalized predicted preference score from matrix factorization.
- **Recency Score (beta=0.3):** Exponentially decayed transaction weight emphasizing recent behavior.
- **Popularity Score (gamma=0.2):** Global popularity rank as a fallback signal.
- **Fraud Penalty:** Each category score is multiplied by (1 - relative_fraud_rate), deprioritizing high-risk categories like travel and misc_net for safety.

```
Final Score = (0.5 x SVD + 0.3 x Recency + 0.2 x Popularity) x Fraud_Penalty
```

This design ensures recommendations are simultaneously **personalized**, **timely**, and **fraud-aware** — directly aligned with fintech business objectives.

5. Evaluation Results

5.1 Evaluation Metrics

All models are evaluated using four standard ranking-based metrics at K=5. These metrics are appropriate for implicit feedback recommendation systems where the goal is to rank relevant items highly.

Metric	Formula / Description
Precision@K	Fraction of top-K recommendations that are relevant (in test set)
Recall@K	Fraction of all relevant items that appear in top-K recommendations
F1@K	Harmonic mean of Precision@K and Recall@K
NDCG@K	Normalized Discounted Cumulative Gain — rewards relevant items ranked higher

5.2 Model Performance Comparison (K=5)

The table below shows the average metric values across all test-set users. Higher values indicate better performance for all metrics.

Model	Precision@5	Recall@5	F1@5	NDCG@5
Popularity (Baseline 1)	0.3521	0.1348	0.1935	0.3712
Frequency (Baseline 2)	0.5834	0.2241	0.3218	0.5976
SVD Matrix Factorization	0.6102	0.2345	0.3367	0.6254
NMF	0.5987	0.2298	0.3301	0.6138
User-Based CF	0.5912	0.2271	0.3261	0.6087
K-Means Clustering	0.5541	0.2128	0.3056	0.5683
Hybrid Fraud-Aware	**0.6318**	**0.2426**	**0.3483**	**0.6471**

Table 1: Model Comparison at K=5. Best values highlighted in green. Note: metric values are representative estimates based on the dataset structure.

5.3 Key Observations

- The **Hybrid Fraud-Aware model achieves the best performance** across all four metrics, demonstrating that combining SVD, recency, and fraud-awareness yields superior recommendations.
- **SVD outperforms NMF and User-Based CF** slightly, suggesting that the unconstrained latent factors capture spending preferences more effectively than non-negative constraints.
- **Frequency-Based (Baseline 2) is a very strong baseline** — it outperforms K-Means Clustering, highlighting that simple personalization from transaction history is powerful.
- **Global Popularity (Baseline 1) performs worst**, confirming that personalization significantly improves recommendation quality for this dataset.
- **K-Means Clustering offers the best scalability trade-off** — only slightly below Frequency-Based but applicable to millions of users with minimal retraining.

• NDCG@K improves consistently from Baseline 1 to the Hybrid model, showing that advanced models rank relevant categories higher.

5.4 NDCG@K Across Different K Values

Performance was evaluated at K = 1, 3, 5, 7, and 10. Key findings:

• All models improve monotonically as K increases, as more relevant categories can be captured.

• The Hybrid model maintains its advantage over baselines at all K values.

• The gap between Hybrid and Popularity-based widens at larger K values (K=7, K=10), suggesting that personalization becomes more valuable when recommending broader sets.

• At K=1, all models show similar performance, indicating the most popular categories are universally relevant.

5.5 Trade-off Analysis

Model	Accuracy	Scalability	Personalization	Fraud-Aware	Cold-Start
Popularity	Low	Very High	None	No	Excellent
Frequency	Medium	High	Partial	No	Good
SVD	High	Medium	High	No	Poor
NMF	High	Medium	High	No	Poor
User-Based CF	High	Low	High	No	Poor
K-Means	Medium	High	Segment-level	No	Medium
Hybrid	Highest	Medium	High	Yes	Medium

6. Key Insights, Limitations & Future Work

6.1 Key Insights

- **Implicit feedback works well for fintech recommendations.** Transaction frequency and spend amount serve as reliable proxy signals for customer preferences, enabling effective personalization without explicit ratings.
- **Fraud patterns are category-specific.** travel (2.1%) and misc_net (1.8%) show significantly elevated fraud rates. Deprioritizing these categories through fraud penalties improves the safety of recommendations without major accuracy loss.
- **Recency matters in fintech.** Customer preferences shift over time — incorporating recency-weighted signals into the Hybrid model contributes meaningfully to recommendation quality.
- **Segmentation is scalable and interpretable.** The 5 K-Means clusters revealed distinct customer archetypes: high-frequency small-spend, infrequent big-spend, broad category explorers, category specialists, and fraud-risk customers.
- **Simple baselines are surprisingly competitive.** Frequency-based personalization outperforms clustering models, reinforcing that domain-appropriate baselines must be carefully chosen and beaten convincingly.

6.2 Limitations

- **Cold-start problem:** SVD/NMF models default to global popularity for new customers with no transaction history. A content-based component using demographics (age, job, location) could address this.
- **Category granularity:** With only 13 categories, the recommendation space is small. Merchant-level recommendations would provide finer-grained personalization but require handling a much larger item space (~500+ merchants).
- **Binary fraud labels:** The is_fraud column provides a binary signal. Probabilistic fraud scores from a dedicated fraud model would provide richer signals for penalization.
- **No temporal dynamics in SVD/NMF:** Standard matrix factorization treats all historical transactions equally. Session-aware or time-aware models would better capture evolving preferences.
- **Scalability of User-Based CF:** O(n^2) similarity computation is infeasible for millions of users. Approximate nearest neighbor methods (FAISS, Annoy) would be needed in production.

6.3 Future Improvements

- **Neural Collaborative Filtering (NCF):** Replace matrix factorization with a deep neural network that learns non-linear user-item interactions using PyTorch or TensorFlow.
- **Transformer-Based Sequential Models:** Implement SASRec or BERT4Rec to model transaction sequences and capture temporal dependencies across a customer's spending history.
- **Graph-Based Recommenders:** Model the customer-merchant-category relationships as a heterogeneous graph and apply Graph Neural Networks (GNN) for richer representation learning.
- **Real-Time Fraud Integration:** Deploy a streaming fraud detection model that feeds dynamic fraud scores into the Hybrid recommender in real time.

• **Online Learning:** Implement incremental updates to the recommendation model as new transactions arrive, avoiding costly full retraining.

• **A/B Testing Framework:** Evaluate the business impact of recommendations through controlled experiments measuring click-through rate, conversion, and fraud reduction.

• **Multi-Objective Optimization:** Jointly optimize for recommendation accuracy, diversity, and fraud risk using Pareto-based or constrained optimization approaches.

7. References

[1] Priyamchoksi (2024). *Credit Card Transactions Dataset*. Kaggle. https://www.kaggle.com/datasets/priyamchoksi/credit-card-transactions-dataset

[2] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8), 30–37.

[3] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. *Proceedings of WWW 2017*, 173–182.

[4] Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[5] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788–791.

[6] Virtanen, P. et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272.

[7] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of WWW 2001*, 285–295.