# Personalized Merchant Recommender Engine utilizing Credit Card Transaction

LAN MINGYI
National University of Singapore

***Abstract*—With the advanced machine learning techniques, the recommendation in credit card market has moved towards highly personalized marketing. Understanding individual preferences and spending behaviors has grown significantly. Building a personalized system to suggest products or merchants that align with customers' preferences can enhance customer engagement more efficiently. This paper aims to compare and experiment with the popular existing research models, especially matrix factorization and neural collaborative filtering techniques, utilizing an annual credit card transaction data. For the deep learning approach, the evaluation metrics shows obvious improvement by incorporating hidden layers to learn the nonlinear interactions between users and items. The precision score increases 2.64% to 7.02% for top 1 recommendations among selected 10K customers. However, it suffers from the tradeoff between accuracy and computing intensity. Future work can be done for more context-aware models.**

***Index Terms*—Collaborative filtering, recommendation systems, Neural Network, Deep Learning, Matrix Factorization, Non-negative matrix factorization, Implicit Feedback, merchant recommendation; credit card payment data**

## I. INTRODUCTION

As the credit card market continues to grow steadily, banks are adopting new marketing strategies to attract customers. The application of recommendation systems to credit card transactions has yielded noteworthy outcomes. Jannach et al. [1] observed that by analyzing transaction data, financial institutions could identify customer preferences and spending habits. Recommending merchants based on customers' individual interests and preferences can enhance customer engagement, leading to higher satisfaction and loyalty. Additionally, the recommender engine strengthens partnerships with merchants, leading to increased sales and revenue-sharing opportunities. It also enables cross-selling of other banking products and services. Implementing such a system gives banks a competitive advantage and improves customer retention.

In this paper, we are going to experiment with the Matrix Factorization (MF) and Neural Collaborative Filtering (NCF) models as they have gained significant attention due to their effectiveness in capturing user-item interactions and improving recommendation accuracy. Customers were divided into three groups (Been There/Disengaged/Never Been There) according to their visiting frequency during different time periods, and performance was compared for each group.

The paper is structured as follows:

Section 2 begins with a literature review that explores popular approaches used by recommendation system including content-based recommendation, collaborative filtering and hybrid methods. In section 3, we elaborate on the mechanisms of matrix factorization (MF) and neural collaborative filtering (NCF). Then we explore and process the real-world credit card transaction data from a bank in section 4. In section 5, we then construct the merchant recommendation system and evaluate the model performance by comparing the metrics. Finally, we wrap up the paper by concluding the findings and discussing further improvements in section 6 and 7.

## II. LITERATURE REVIEW

Over the years, extensive research has been conducted in the field of recommendation systems, leading to the development of various algorithms and techniques including collaborative filtering, content-based filtering, and hybrid method etc.

Collaborative filtering aims to find other users with similar interests and preferences by studying their past behaviors and then recommend items that those similar users have interaction with to the target user [2]. Content-based filtering, however, will analyze the attributes of items, in this case, the brand, category, location and customer reviews of the merchants, and recommend items with similar attributes [3].

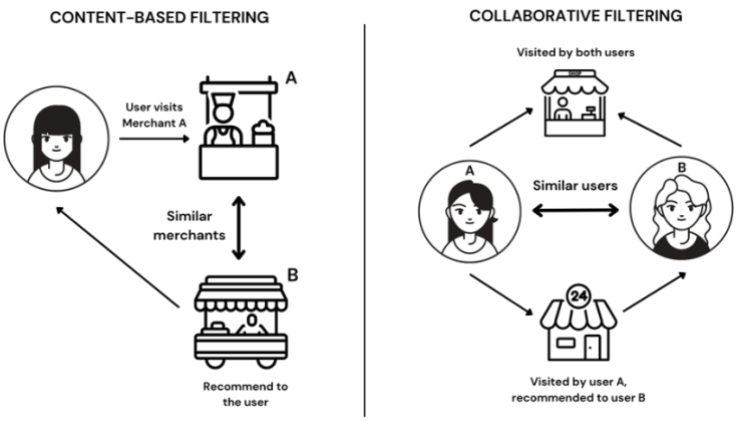



**Fig. 1.** Content-based Filtering vs. Collaborative Filtering

Content-based filtering has advantages over collaborative filtering, but it also has drawbacks. This type of filtering may handle new or specialized objects better because it is independent of the ratings or interactions of other users [4]. By

demonstrating the rationale behind a recommendation made in light of an item's qualities and enabling customers to control or modify their selections, it also makes recommendations more apparent and understandable. It may, however, be limited by the caliber and accessibility of item information, experience overspecialization or a lack of diversity, and neglect to account for social or contextual factors that influence user preferences [5]. In other words, collaborative filtering can capture complex patterns and user preferences even when item attributes are not well-defined, so it is better suited to large or dense datasets. This makes it possible to provide unexpected and diverse recommendations that can introduce users to new items, even though it may suffer from the "cold start" problem (new users/items have limited data) [6]. Contrarily, content-based filtering works better with smaller or sparser datasets and with objects that are simple to characterize and contrast based on their attributes.

TABLE I
COMPARISON BETWEEN CONTENT-BASED AND COLLABORATIVE FILTERING

| Model | Content-based | Collaborative |
|---|---|---|
| **Examples** | If a user likes action movie, another action movie is recommended | If user A and B have similar preferences and behaviors, and user B likes a certain video, then this video is recommended to user A |
| **Pros** | •Doesn't rely on user data<br>•Handle new or niche items better<br>•Transparent and explainable recommendations | •Doesn't require domain knowledge<br>•Provide novel and unexpected recommendations |
| **Cons** | •Requires a lot of domain knowledge, limited by the quality and availability of the item information<br>•Struggle to recommend items outside a user's explicitly indicated interests | •Hard to handle fresh new items with limited interaction history (cold start)<br>•Hard to include features beyond the query |

Model-based collaborative filtering and memory-based collaborative filtering are the two basic subcategories of collaborative filtering. In model-based collaborative filtering, patterns and correlations are learned from user-item interaction data by developing a mathematical model. These models were developed to forecast missing or ambiguous user evaluations on objects using prior user-item interactions as their training data. It is scalable to huge datasets, handles sparse data effectively, and captures complicated and hidden patterns in the data. However, it involves a lot of computation during training, and fine-tuning the model could be difficult. Memory-based collaborative filtering, often referred to as neighborhood-based collaborative filtering, works by comparing persons or objects based on their previous interactions to determine how similar they are.

The field of personalized content delivery and information retrieval has turned to hybrid recommendation systems, which combine several recommendation approaches. With the aim of giving consumers suggestions that are not only more accurate but also inclusive in their variety and contextual relevance, these techniques aim to go above the limitations of solo methods. Hybrid recommendation strategies include weighted combination, context-aware approaches, fusion of predictions, neural network integration, and ensemble approaches etc. For example, weighted combination combines suggestions from many strategies using specified or acquired weights, like using weighted averages of predictions from collaborative filtering and content-based filtering. By doing this, it takes use of each method's unique strengths while minimizing its flaws.

Researchers have also explored the integration of deep neural networks with collaborative filtering (DL-DF approaches) since the development of deep learning. Multi-Layer Perceptrons (MLPs) and Autoencoders were early DL-CF models that showed the power of deep learning for collaborative filtering tasks. In order to increase the accuracy of recommendations, these models incorporate non-linear interactions and latent representations of people and things.

Comparatively to content-based filtering, collaborative filtering and deep learning techniques are better suitable for developing the merchant recommendation system given the size of the dataset employed in this study. On one hand, collaborative filtering can address the scalability and sparsity issues that come with large datasets. On the other hand, deep learning techniques, in particular deep neural networks, have demonstrated a high degree of effectiveness in recommendation systems using substantial datasets. When dealing with substantial transaction data and varied user preferences, they can capture complicated patterns and interactions between users and objects.

## III. MATHODOLOGY

### *A. Matrix Factorization*

Matrix Factorization involves breaking down the user-item interaction matrix into two matrices: one for user-related latent factors and another for item-related latent factors, achieved through matrix decomposition. Consequently, the user-item interactions are represented by their dot product within an n-dimensional latent factor space.

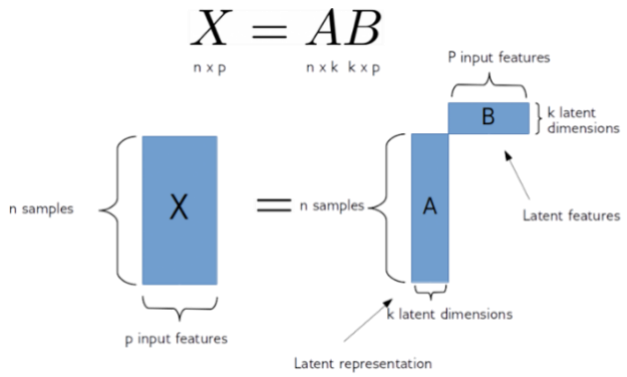



**Fig. 2.** Matrix Factorization Mechanism

The mathematical representation of Matrix Factorization is as follows:

$$\hat{y}_{ui} = f\left(u, i \middle| \boldsymbol{p_u}, \boldsymbol{q_i}\right) = \boldsymbol{p_u^T q_i} = \sum_{k=1}^{K} p_{uk} q_{ik} \qquad (1)$$

where
$\hat{\boldsymbol{y}}_{\boldsymbol{ui}}$**:** prediction score
$\boldsymbol{p_u}$**:** latent vector for user $u$
$\boldsymbol{q_i}$**:** latent vector for item $i$
***K:*** the dimension of latent space

There are five popular techniques in matrix factorization: Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), Probabilistic Matrix Factorization (PMF), Alternating Least Squares (ALS), and Stochastic Gradient Descent (SGD). Singular Value Decomposition (SVD) and Alternating Least Squares (ALS) are two matrix factorization algorithms that have emerged as important tools in recommendation systems. The user-item interaction matrix is decomposed into latent variables via SVD, allowing the system to capture underlying patterns. Meanwhile, ALS adjusts user and item factors repeatedly, demonstrating scalability and efficacy.

However, when it comes to a recommendation system based on credit card transactions, Non-Negative Matrix Factorization (NMF) is the clear winner. NMF imposes non-negativity requirements on factors, which aligns well with credit card transaction data, which contains positive monetary values by definition. This property enables NMF to develop interpretable and contextually relevant latent variables capable of modeling spending patterns and preferences.

*B. Neural Collaborative Filtering*

While Matrix Factorization (MF) has proven to be effective for collaborative filtering, its performance can be hindered by the simplistic choice of the interaction function, specifically the inner product. He, Xiangnan, et al. [5] have provided an example to illustrate the possible limitation of MF due to its use of a simple and fixed inner. Since MF maps users and items into the same latent space, their similarity can be measured using an inner product or the cosine of the angle between their latent vectors. And the Jaccard coefficient represents the ground truth for assessing the similarity between two users, which MF aims to recover accurately.

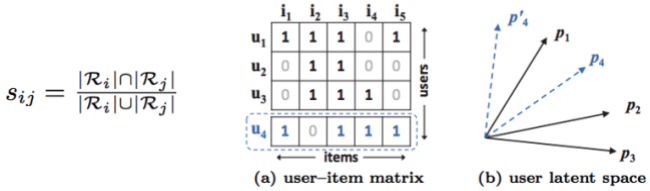



**Fig. 3.** MF: user-item latent space projection

For example, for the first three rows in (a), the geometric relations of $p_1, p_2$, and $p_3$ in the latent space are plotted in Figure 3b, where $S_{23}(0.66) > S_{12}(\ (0.5) > S_{13}(\ (0.4)$. Now, consider a new user, $u_4$, represented by the dashed line in Figure 3a. The similarity scores are $S_{41}(\ (0.6) > S_{43}(\ (0.4) > S_{42}(\ (0.2)$, indicating that $u_4$ is most similar to $u_1$, followed by $u_3$, and least similar to $u_2$. However, if an MF model positions $p_4$, closest to $p_1$ (options shown in Figure 3b with dashed lines), it will mistakenly place $p_4$ closer to $p_2$ than to $p_3$, resulting in a significant ranking loss.

One approach that can tackle this issue is to use a large number of latent factors, K. However, increasing K may negatively impact the model's generalization ability. To address the shortcomings of the traditional MF model, He et al. [7] proposed the NCF architecture, leveraging neural networks to learn intricate user-item interactions and feature representations.

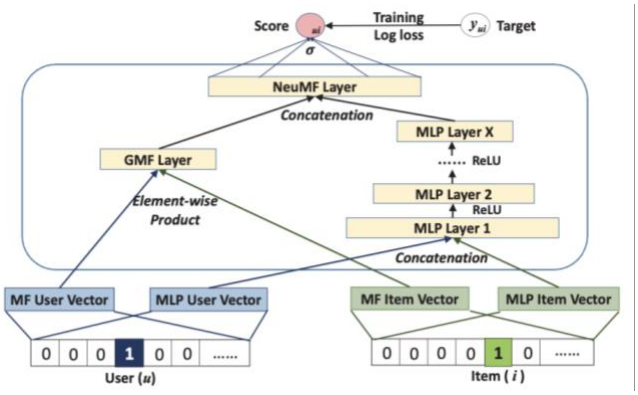



**Fig. 4.** Neural Collaborative Filtering Architecture

The proposed Neural Matrix Factorization model combines the linear aspects of the Generalized Matrix Factorization (GMF) model with the nonlinear characteristics of the Multi-Layer Perceptron (MLP) model. This amalgamation capitalizes on the strengths of each model while mitigating their limitations. GMF and MLP employ distinct embedding layers for users and items. In the architecture, as depicted in Figure x, user and item feature vectors from the input layer are

represented in sparse one-hot encoding. These sparse representations undergo transformation via an embedding layer, generating k-dimensional latent vectors. The latent vectors are then fed through a neural collaborative filtering layer, akin to a general embedding process. Subsequently, the outputs of each model are combined with a weight parameter h in the output layer to yield the final prediction. The architecture is illustrated in the following mathematical expressions:

(1) Concatenating the last hidden layer for GMF and MLP:

$$\emptyset_{u,i}^{GMF} = p_u^{GMF} \odot q_i^{MLP}$$

(2) Obtain $\emptyset^{GMF}$ from MLP:

$$\emptyset_{u,i}^{MLP} = a_{out}\left(W_L^T\left((a_{out}\left(\ldots a_{out}\left(W_2^T\begin{bmatrix}p_u^{GMF}\\ q_i^{MLP}\end{bmatrix} + b_2\right)\ldots\right)\right)\right) + b_L$$

(3) Fuse output from GMF and MLP:

$$\hat{r}_{u,i} = \sigma\left(h^T\begin{bmatrix}\emptyset^{GMF}\\ \emptyset^{MLP}\end{bmatrix}\right)$$

## IV. DATASET

### *A. Dataset Description*

To incorporate seasonality factors, we used a 12-month data covering the entire year of 2022 for training and the following three months in 2023 as the validation dataset. The credit card transaction dataset contains valuable information, including customer unique identifiers, merchant names and categories, credit card types, account types, transaction frequency, and amounts etc.

TABLE II
DATASET SAMPLE

| Customer ID | Merchant Category | Merchant Name | Credit Card | Account Type | Transaction Time | Transaction Count | Transaction Amount |
|---|---|---|---|---|---|---|---|
| *xxxxxxxx* | *Restaurant* | *KFC* | *xxx* | *xxx* | *2022-01-17* | *1* | *18.5* |
| … | … | … | … | … | … | … | … |

### *B. Data Cleaning & Preprocessing*

(a) Remove unmapped merchants

To streamline the data, we created a mapping of merchant categories according to business rules, including Medical, Restaurant, Supermarket, Gasoline, Business Services, and more. Those merchants that are unable to be mapped into the existing merchant categories will be removed.

(b) Remove transactions with negative transaction amount

We noticed that some transactions have negative transaction amount, which could be caused by the following reasons:

a. **Refund:** When a customer returns a purchased item or cancels a service, the merchant issues a refund, leading to the money being credited back to the customer's account.

b. **Cashbacks & Discounts:** Some credit cards offer cashback rewards to customers for making certain types of purchases, and merchants may provide discounts or promotional offers to customers, resulting in a reduced transaction amount.

c. **Authorization Holds:** When a credit card is used for pre-authorization, such as reserving a hotel room or renting a car, the transaction amount might be negative if the final transaction amount is less than the authorized amount.

Those transactions related to special scenarios as illustrated above have limited ability to reflect customers' spending pattern and might introduce some noise to the data, therefore should be removed.

(c) Remove merchants with too few transactions

Additionally, we filtered out merchants with too few transactions. Merchants with infrequent transaction frequencies often lack substantial transaction history, resulting in limited and unreliable data points for analysis. As a consequence, their representation in the recommendation model may be inadequate to capture meaningful patterns or preferences, and could potentially introduce noise or bias into the system's recommendations. By excluding such merchants, the recommendation system focuses its attention on merchants with more robust and informative transaction histories, allowing it to provide more accurate and relevant recommendations to users. After experimenting with several thresholds, we remove merchants with fewer transactions. This can help eliminate bias datapoints while keep the robustness of the data since the total transaction amount is still 95% of the original amount after removal.

(d) Change all the transaction count to '1'

Though transaction frequency tells how popular the merchant is, in the case of recommendation system, it might cause some bias towards more low-spending merchants. For example, restaurants like 'McDonald's' have a large customer base, and possibly it has become customers' consumption habits. Even if we don't recommend these merchants, customers might still visit them as usual. Plus, the transaction amount of these merchants could be relatively small compared with merchant categories like 'Education', 'Trave', and 'Business Services' etc. Therefore, to avoid this situation, we simply replace all transaction count with '1', indicating that these merchants have been visited.

### *C. Feature Selection*

(a) Transaction frequency

The data was further grouped by customer unique identifier and merchant name, aggregating transaction frequency and amount. Among all the variables, the most useful feature to consider would be 'transaction count'. Though transaction amount can also show the relative importance of the merchant, it will push the recommendation system to recommend more

high-end merchants like luxury stores, Airlines, and Insurance companies etc. Those merchants, despite higher transaction amount per visit, might not have a very large customer base and frequent visits from the same customer. For example, if a female customer has purchased a Louis Vuitton handbag this month, she might be less likely to visit the merchant again in the near future. But if we use transaction amount as the single feature, the model will skew to these higher-end merchants, making it less generalizable for all customers. Transaction count, however, after we change all values to 1, helps force the model to treat all merchants more equally, reducing possible bias and making it a better option in this case

(b) Card Type

After conducting a comprehensive analysis of different card types, distinct consumption behaviors have emerged among various categories of cardholders. These behaviors can be broadly categorized into five groups:

1. **High-End Customers:** This group constitutes a small fraction of the total customer base, approximately 1.03%. Nevertheless, they make a considerable contribution to overall average expenditures, making up around 50% of it. Luxury goods like Louis Vuitton and Hermès, as well as industries like airlines, electronics, and insurance businesses, are preferred by high-end consumers. These preferences are reflected in their high average expenditure.

2. **Female Customers:** Designed exclusively for women, these card types offer specialized perks and advantages to meet their unique preferences and requirements. Rewards, discounts, and privileges for dining, shopping, beauty, and wellness-related spending are among these perks. Notably, the top 10 merchant categories for this demographic include the "Retail-Fashion" and "Personal Care" sectors, highlighting their distinctive buying habits.

3. **Partnerships with E-Commerce Platforms:** To offer co-branded credit or debit cards, financial institutions typically work with e-commerce platforms. When making purchases on the partner platform, these cards provide consumers certain advantages including cashback, savings, and special offers. These cardholders primarily utilize their cards for online shopping.

4. **Younger/Lower-End Customers:** These cards feature loose application standards since they are designed for those with lesser incomes or younger populations, such as college students. This group spends its money mostly on everyday necessities, transportation costs, and digital wallet transactions. The frequency of transactions is noticeably higher despite the decreased transaction amounts.

5. **Normal Customers:** The majority of cardholders fall into this group. Their consumption scenarios are more diverse. They engage in a wide range of consuming activities, from regular small-dollar purchases to more expensive consumption like insurance and travel agency services.

The possibility to create individualized recommendation systems catered to each segment is crucial given the various traits displayed by these cardholder groupings. By taking into account each group's particular tastes and needs, we can consider train separate models tailored to different customer groups for more personalized recommendations. That'll leave for future work.

## V. Experiment

### A. Customer Segmentation

From merchant's perspective, we categorized customers into three groups based on their visit frequency during certain time period.

- **Been There** – In the window of last 12 months, customer had done transaction with the merchant in last 3 months.
- **Disengaged** – In the window of last 12 months, the customer had not done transaction in the recent 3 months but had done transaction in the 9 months before that.
- **Never Been There** – Customer had not done any transaction with the merchant in last 12 months.

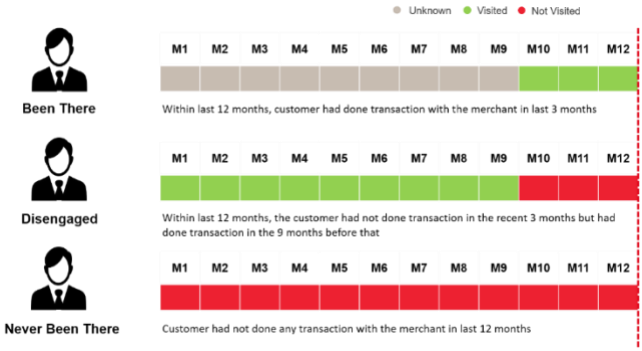



**Fig. 5.** Customer Segmentation

### B. Evaluation Metrics

The metrics we use for evaluation are mainly precision and recall scores:

$$Precision = \frac{\text{Correct predicted merchants}}{\text{All recommended merchants}}$$

$$Recall = \frac{\text{Correct predicted merchants}}{\text{Actual visited merchants}}$$

We evaluate model performance by calculating the precision and recall scores for top 1, top 3 and top 10 recommendations. Take the 'Never Been There' customer group as an example, we first retrieve all the visited merchants for 'Never Been There' customers over the past year. After getting the full recommendation matrix, we set the prediction scores for visited merchants to negative infinity to ensure these merchants won't be recommended. And we remove the visited merchants for the validation dataset as well since we want to know how many of our predictions of unvisited merchants are correct. Then we sort the remaining merchants by scores to get

the top K recommendations and calculate the metrics. An example is illustrated below.

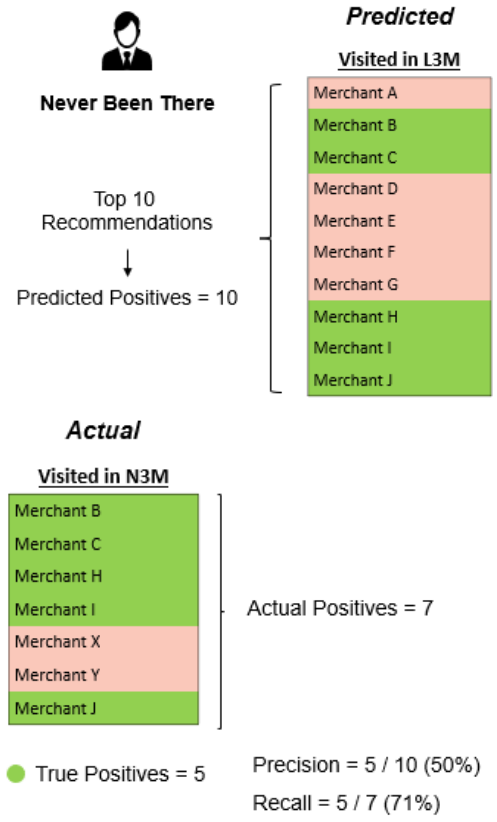



**Fig. 6.** Evaluation score calculation example

*C. Matrix Factorization*

In this study, two model structures were explored: traditional Matrix Factorization using Non-negative Matrix and Neural Collaborative Filtering (NCF) model.

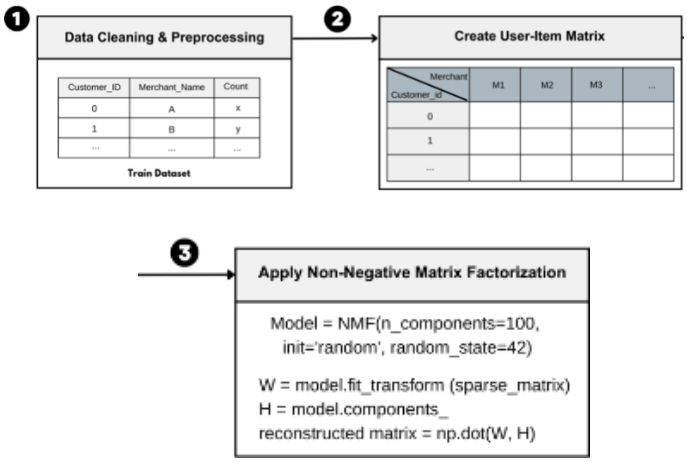



**Fig. 7.** Model fitting: Matrix Factorization

First transform the training data into sparse matrix and then fit into the matrix factorization model with number of components equals to 100. According to the elbow plot, the reconstruction error decreases drastically for small n components, but the curve gets flatter when the number of components becomes larger. However, fitting model with number of components larger than 100 takes a significant amount of time. Thus, 100 seems to be an appropriate option considering the trade-off between reconstruction error and computation power.

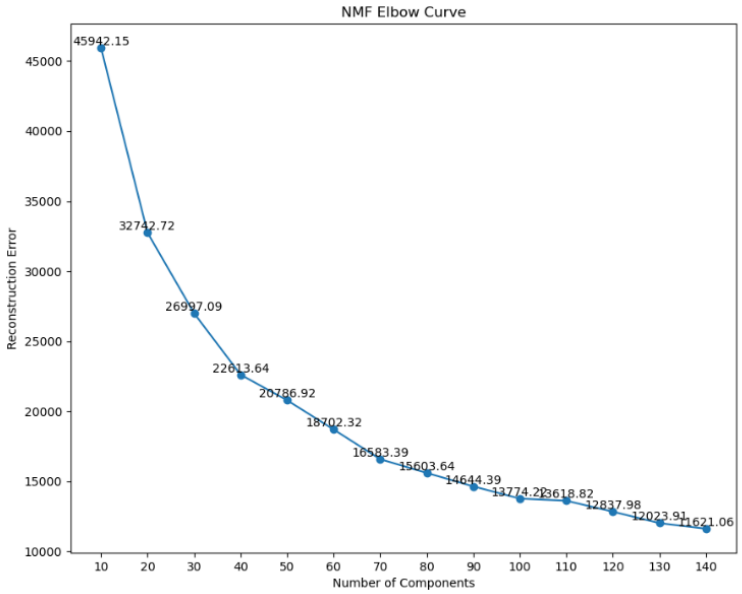



**Fig. 8.** NMF Elbow Curve

The final recommendation is in the matrix format where row represents customers and column indicates merchants. We can look at the recommendation from both merchant's and customer's perspectives. Vertically, we can extract *top K* customers for each merchant while horizontally *top K* merchants can be selected for each customer. The example of the recommendation result is as follows:

TABLE III
MF RECOMMENDATIO MATRIX EXAMPLE

| | **M1** | **M2** | **M3** | **M4** |
|---|---|---|---|---|
| **A** | 0.7 | 0.3 | 0.1 | 1 |
| **B** | 1 | 0.6 | 0.1 | 1.2 |
| **C** | 0.4 | 0.8 | 0.1 | 0.5 |
| **D** | 0..6 | 0.5 | 0.8 | 0.2 |
| **E** | 0.5 | 0.8 | 0.3 | 0.1 |
| **…** | … | … | … | …[1] |

The performance of the algorithm will be evaluated in the 3 months of performance window with the actual transactions made by the eligible customers on the MRE participating merchants. The validation result for the three groups of customers is as follows:

[1] *Note: All figures cited in this paper are fictional for illustrative purposes as real figures are blocked for confidentiality reasons.*

TABLE IV
MF VALIDATION RESULT

| | | Precision | Recall |
|---|---|---|---|
| Been There | Top 1 | 88.61% | 19.95% |
| | Top 3 | 70.48% | 35.34% |
| Disengaged | Top 1 | 21.57% | 3.65% |
| | Top 3 | 18.13% | 6.99% |
| Never Been There | Top 1 | 5.47% | 0.51% |
| | Top 3 | 4.35% | 1.11% |

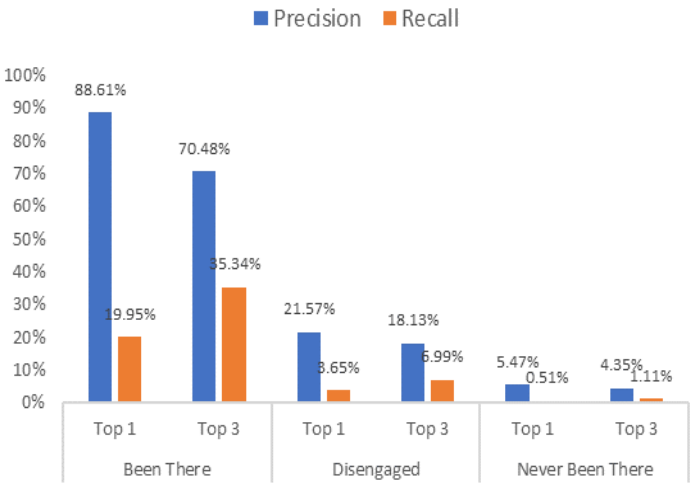



**Fig. 9.** Precision score comparison

*D. Neural Collaborative Filtering*

To train the model, we utilize tensorflow with the following hyperparameters:

TABLE V
HYPERPARAMETER FOR MCF

| Parameters | Values |
|---|---|
| n_factors | 4 |
| layer_sizes | [16, 8, 4] |
| n_epochs | 100 |
| batch_size | 256 |

As the neural collaborative filtering is extremely computationally intensive, we select a customer base of 10K *'Never Been There'* customers to examine the model performance. The following table shows the comparison of validation result between NCF and MF.

TABLE VI
VALIDATION RESULT COMPARISON

| ***Matrix Factorization*** | | | |
|---|---|---|---|
| **Top K** | **Top 1** | **Top 3** | **Top 10** |
| **Match** | 264 | 950 | 2787 |
| **Actual** | 6588 | 6588 | 6588 |
| **Predicted** | 10000 | 30000 | 100000 |
| **Precision** | 2.64% | 3.17% | 2.78% |
| **Recall** | 4.01% | 14.42% | 43.30% |
| **F1 score** | 3.18% | 5.19% | 5.23% |
| ***Neural Collaborative Filtering*** | | | |
| **Top K** | **Top 1** | **Top 3** | **Top 10** |
| **Match** | 702 | 1680 | 3692 |
| **Actual** | 6588 | 6588 | 6588 |
| **Predicted** | 10000 | 30000 | 100000 |
| **Precision** | 7.02% | 5.60% | 3.69% |
| **Recall** | 10.66% | 25.50% | 56.04% |
| **F1 score** | 8.46% | 9.18% | 6.93% |

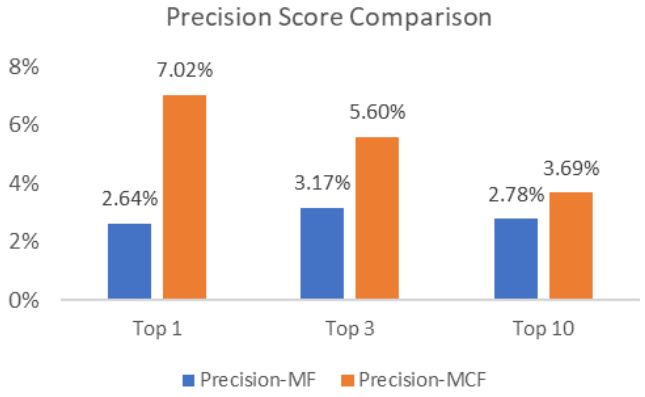



**Fig. 10.** Precision score comparison

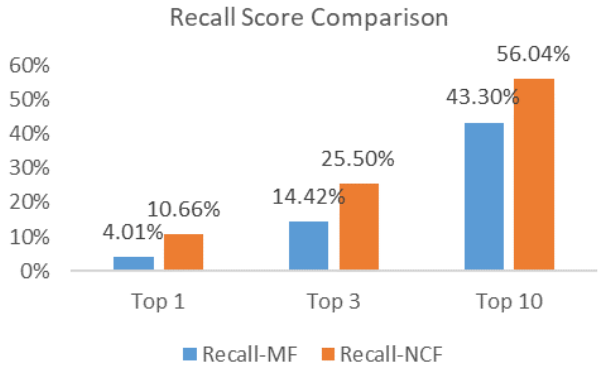



**Fig. 11.** Recall score comparison

We can see apparent improvement for the NCF model. Prominently, the precision score for the top 1 recommendation increases from 2.64% to 7.02%.

## VI. Discussion

In this study, we created a recommendation system based on credit card transactions to help clients find the right retailers. However, a few factors demand further thought for system improvement and refinement. Our dependence on the recall measure, which could not provide an accurate assessment of our system's success, is one major flaw. Particularly in situations where customers contact with several retailers, great accuracy might mask a recall that appears to be low. To give a more in-depth examination, future work can investigate a variety of evaluation criteria, including Mean Average Precision (MAP), Precision at K, and Normalized Discounted Cumulative Gain (NDCG). Additionally, this examination is an offline test. Ideally, we should evaluate the performance using the real-world feedback data after model deployment stage as show in the workflow below. However, the model is still under development stage, making our current evaluation less accurate.

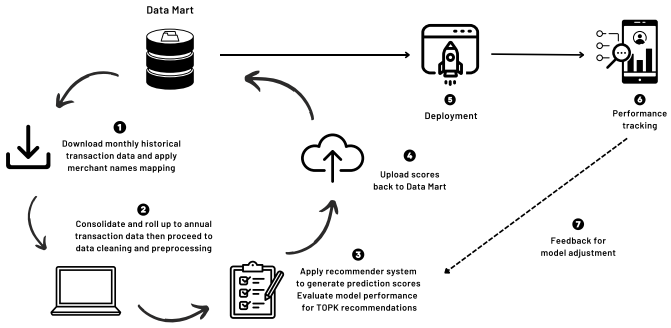



**Fig. 12.** Project Flow

In cases with sizable client bases and lengthy transaction data, the neural collaborative filtering model's high computing requirements, while advantageous for accuracy, present problems. Investigating effective model designs, employing parallel computing techniques, or utilizing distributed computing frameworks can all help to overcome this restriction. Our current dependence on a single attribute, "transaction count," to model interactions, is a related problem. Integration of additional customer- and merchant-related information, such as credit card kinds, demographics, and merchant qualities, such as locations and revenue, may assist to improve suggestion personalization and produce more accurate and individualized recommendations.

Taking into account user characteristics like credit card kinds, demographics, and transaction histories can potentially improve the system's capacity to collect unique preferences. Additionally, the importance of include merchant-related features to improve recommendation quality has been explored in depth by the works of Liu, B. et al. (2013) [8] and Li, X. et al. (2015) [9]. It has been shown that characteristics including retailer locations, yearly income, and popularity help to provide a more thorough insight of merchant profiles and client interactions.

By building on existing strategies, we may get a more comprehensive understanding of client preferences. To sum up, even if our recommendation system has potential, overcoming these drawbacks and seeking new opportunities can improve its functionality, individualization, and suitability for use in actual credit card transaction scenarios.

## VII. Conclusion & Future Work

The goal of this study was to create a strong merchant recommendation system by exploring the constantly changing environment of credit card transactions. Important insights have been gained from a thorough examination of transaction data involving millions of consumers and billions of transactions, which help to further our understanding of customer preferences and habits.

Matrix factorization and neural collaborative filtering techniques were extensively used in this work to build the recommendation system. These approaches were chosen because they were good at uncovering hidden correlations and patterns in the enormous dataset of credit card transactions.

The article also discussed the difficulties of dealing with large amounts of credit card transaction data. The complexities of working with enormous datasets, the relevance of data pretreatment, and the value of choosing the right assessment metrics were all covered. Transaction data's nature, where customers interact with a variety of retailers, frequently leads to a lower recall rate while obtaining high accuracy. The evaluation of model performance is hampered by the absence of accurate feedback information following model deployment and marketing campaigns. Along with improving prediction accuracy, the use of neural collaborative filtering also highlighted the method's high computing requirements. This is a problem, especially when working with large client bases and volumes of transactional data.

Investigating ways to reduce the processing requirements of neural collaborative filtering, whether through parallelization or distributed computing techniques, is necessary to overcome this restriction. Given these factors, this work makes a contribution to the field of recommendation systems by elucidating how matrix factorization and neural collaborative filtering are used in the context of credit card transactions. This study addresses the shortcomings while laying the groundwork for future investigation, optimization, and improvement of customized merchant recommendation systems targeted to various client categories.

## References

[1] Jannach, D., Ludewig, M., & Lerche, L. (2021). Financial Product Recommendation: The Role of Transaction Data. Journal of Retailing and Consumer Services, 59, 102358.
[2] Resnick, P., & Varian, H. R. (1997). Recommender Systems. Communications of the ACM, 40(3), 56-58.
[3] Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. In The Adaptive Web, 325-341.
[4] Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.

[5] Hurley, N., & Zhang, M. (2011). Novelty and diversity in top-n recommendation--analysis and evaluation. ACM Transactions on Internet Technology (TOIT), 10(4), 1-30.
[6] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems, 22(1), 5-53.
[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural Collaborative Filtering," in Proceedings of the 26th International Conference on World Wide Web, 2017.
[8] Liu, B., & Xiong, H. (2013, May). Point-of-interest recommendation in location based social networks with topic and location awareness. In *Proceedings of the 2013 SIAM international conference on data mining* (pp. 396-404). Society for Industrial and Applied Mathematics.
[9] Li, X., Xu, G., Chen, E., & Li, L. (2015, November). Learning user preferences across multiple aspects for merchant recommendation. In *2015 IEEE International Conference on Data Mining* (pp. 865-870). IEEE.
[10] Rendle, S., Krichene, W., Zhang, L., & Anderson, J. (2020, September). Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems* (pp. 240-248).

[i]

[i] *Note: All figures cited are fictional for illustrative purposes as real figures are blocked for confidentiality reasons.*