Team Member:
Sonia Kashyap
Pooja Kumari
Sam Thomas Mathew

Final Report: Customer Product Recommendation System

1. Introduction

This project aims to build a customer-focused product recommendation system using data from historical customer interactions. By leveraging various customer behavioral features and advanced machine learning techniques such as Singular Value Decomposition (SVD), we developed a model that predicts products each customer is likely to purchase. Our goal was to ensure personalized, accurate, and highly relevant recommendations.

Introduction to Recommender Systems

Recommender systems are algorithms used to suggest relevant items to users based on their preferences, behavior, or demographic information. These systems are common in industries like e-commerce, streaming services, and social media platforms, aiming to enhance user engagement by providing personalized recommendations. However, one of the key challenges faced by recommender systems is the **cold start problem**—a situation where the system struggles to make recommendations due to limited or no data on new users or products.

---

2. Different Types of Recommender Systems

a) Collaborative Filtering

Collaborative filtering is based on the idea that users who have similar preferences in the past will continue to show similar preferences in the future. It can be divided into:

- **User-based Collaborative Filtering:**
  Recommends items based on the preferences of similar users. If User A likes Items 1, 2, and 3, and User B likes Items 2, 3, and 4, the system will recommend Item 4 to User A, considering User A and User B's preferences are similar.
- **Item-based Collaborative Filtering:**
  Recommends items that are similar to the ones a user has interacted with. For example, if User A liked Item 1, the system recommends other items that are often bought or liked alongside Item 1.

b) Content-Based Filtering

Content-based filtering recommends items based on the attributes of the items themselves, such as genres or features. For instance, if a user watched several science fiction movies, the system will recommend other movies within the science fiction genre.

c) Hybrid Methods

Hybrid recommender systems combine collaborative filtering and content-based filtering to leverage the strengths of both methods. This combination allows the system to make better recommendations, especially in cases where one approach might not perform well due to lack of data.

---

3. Overcoming the Cold Start Problem in Recommender Systems

The **cold start problem** occurs when there is insufficient or no data about new users or products, making it difficult for the recommender system to make meaningful suggestions. This is a significant challenge, especially in large systems where new users and products are frequently added. To overcome this issue, we incorporated key features and techniques aimed at handling scenarios where customer and product interaction data are limited.

Cold Start Analysis

- **67.83% of customers** bought only from the top 10 products, indicating that a large portion of the customer base has a narrow product preference. This makes personalized recommendations harder as many users are confined to a small set of popular products.
- **4.75% of customers** are cold-start users, meaning they have made fewer than 2 or 3 purchases. With limited transaction history, it is challenging to understand their preferences and recommend relevant products.
- **13.07% of products** were identified as cold-start products, having limited sales or interactions, making it hard to generate sufficient data for personalized recommendations.

Feature Engineering to Overcome Cold Start

To mitigate the cold start problem, several feature engineering strategies were applied:

- **Total Quantity & Spend:**
  By tracking the total quantity and total spend for each customer-product pair, we prioritized products that are popular and more likely to be recommended. This helps in suggesting items with high customer demand, even for new users or products with limited data.
- **Frequency of Purchases (Distinct Invoices):**
  By counting the distinct invoices per customer, we captured how often customers interacted with specific products. This feature helped us understand a customer's frequency of purchases, which in turn allowed the system to recommend products aligning with a user's purchasing habits.
- **Recency of Purchase:**
  The recency of purchases was considered to ensure that recommendations were based on products that customers have shown recent interest in. This increases the chances of providing a relevant and timely recommendation.
- **Consistency in Purchase Behavior:**
  The consistency of a customer’s purchase behavior was measured by the ratio of distinct invoices to distinct months. A higher consistency ratio indicated that the customer is a regular buyer. This feature helped in recommending products to customers who show a more stable and predictable purchasing behavior.

Impact of Features on Cold Start

- For **cold-start users** with limited transactions, we recommended products from the top-selling categories or matched them with profiles of customers who exhibited similar purchase patterns. This allowed the system to recommend items based on similarity to other users with a more extensive interaction history.

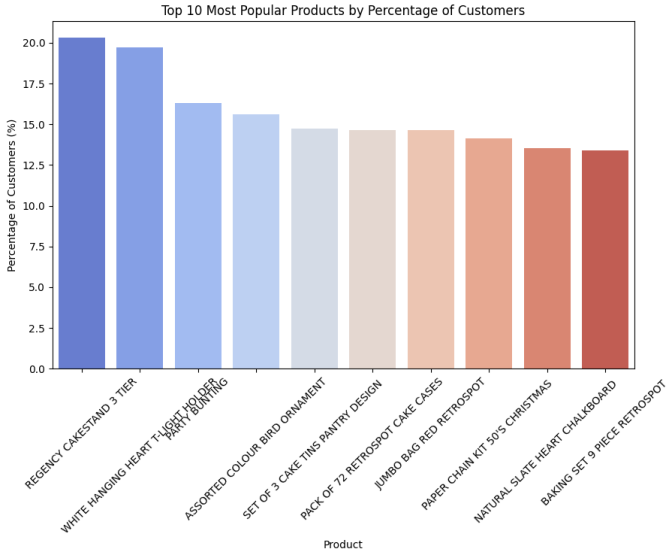
- For **cold-start products** with limited sales, we recommended these products to users who had previously shown an affinity for niche or specific items. The recommendations were based on patterns identified in user behavior towards similar products or attributes.

---

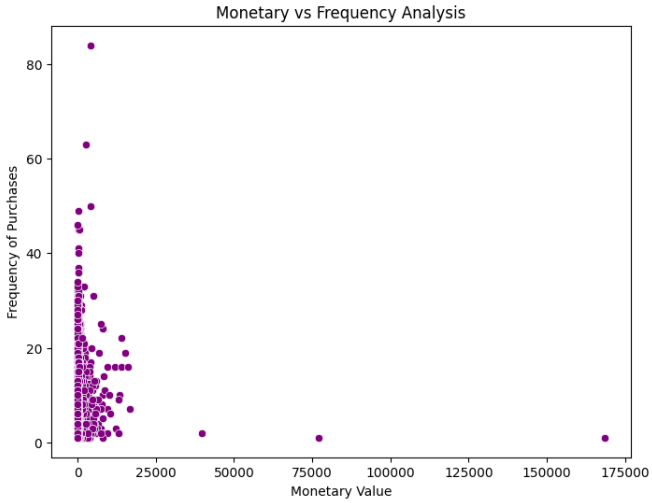
4. Exploratory Data Analysis (EDA)

EDA is a crucial step in understanding the data before model building. In this project, we conducted several EDA techniques to uncover patterns and trends that helped improve the recommendation system:

- **Product Purchase Distribution:**
  We analyzed the frequency of product purchases, identifying popular products and trends in customer preferences. Products with more frequent purchases were prioritized for recommendations.

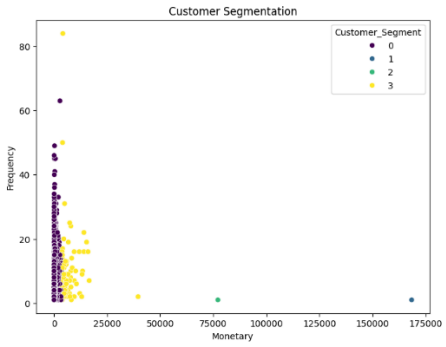


- **Customer Activity Analysis:**
  By assessing customer purchase frequency and total spend, we categorized customers as active or less engaged, tailoring recommendations based on their purchasing behavior.

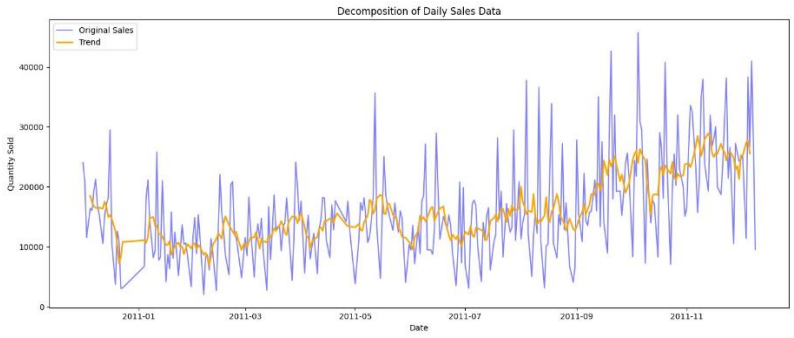


- **Temporal Trends:**
  We examined seasonal trends to understand customer behavior over time, ensuring that the system could adapt to temporal fluctuations in demand.
- **Handling Missing Data:**
  We addressed any missing data through appropriate imputation methods to ensure that the model could process the data without biases or inaccuracies.
- **Product Popularity**: Top-selling items capture 13-20% of customer demand.

- **Customer Segmentation**:
  - **Cluster 0**: Frequent, low spenders.
  - **Cluster 1**: VIP customers with high spending but low purchase frequency.
  - **Cluster 2**: Moderate spenders with steady purchasing habits.
  - **Cluster 3**: High-frequency, low-to-moderate spending customers.



- **Sales Trends**:
  - **Steady growth** in sales over time, with some volatility due to promotions.
  - **Minimal correlation (-0.045) between price and popularity**, indicating customers prioritize factors beyond cost.
  - Steady Growth: Sales are consistently rising over time.
  - Highly Volatile: Sales show sharp fluctuations.
  - Surging Peaks: Sudden spikes hint at promotions or special events.
  - Slow Start: Early sales were relatively low.
  - Seasonal Patterns: Recurring trends suggest cyclical demand.



---

5. Conclusion and Future Work

In summary, overcoming the cold start problem is vital for building effective and personalized recommender systems. By incorporating feature engineering techniques such as tracking total quantity and spend, analyzing purchase frequency, considering recency of purchases, and evaluating consistency, we were able to handle both cold-start users and cold-start products effectively.

The system now provides meaningful recommendations even when interaction data is limited, ensuring that new users and products can still be part of the recommendation process. Future work may focus on refining the hybrid methods, incorporating deep learning techniques, or developing more sophisticated models for better handling of the cold start problem and improving recommendation quality across diverse scenarios.

6. Data Preparation and Cleaning

The dataset used for this analysis contains transactional data from an e-commerce platform. The following preprocessing steps were carried out to ensure the dataset was clean and suitable for modeling:

- **Removal of Null Descriptions and Negative Quantities**: Products with missing descriptions or with negative quantities were excluded from the dataset. This step was critical because products with no description provide no valuable information for the recommendation model, and negative quantities could indicate data entry errors or returns, which aren't useful for predictive modeling.
- **Handling Missing Data**: We ensured that any missing values in important features (such as product descriptions or quantities) were addressed to avoid skewing results. Any missing data were either removed or imputed based on the context.

---

7. Feature Engineering

Several customer features were engineered to capture essential patterns in purchasing behavior. These features allowed the recommendation system to assess customer preferences more accurately.

7.1 CustomerID

Each customer was identified by a unique ID, enabling personalized product recommendations based on their individual purchasing behavior.

7.2 Description

Product descriptions played an important role in identifying which items were of interest to a customer. By analyzing customers' historical product interactions, the model could identify trends in the types of products they purchase (e.g., similar categories or characteristics).

7.3 Monetary

Monetary captures the total spending of each customer, which was used to determine high-value customers. Customers with higher total expenditure were likely to have more consistent preferences, making them important for personalized recommendations.

7.4 Frequency

Frequency represents how many times a customer has made a purchase. Frequent buyers are more likely to engage with products repeatedly, so understanding their frequency is key in identifying potential products for recommendation.

7.5 AvgBasketSize

The average value of items purchased per transaction was calculated for each customer. This feature helps identify whether a customer tends to buy high-value items or smaller, more frequent products. This allows the system to suggest products of similar value.

7.6 PriceRange

PriceRange categorizes products into various price tiers, helping the system recommend items within a customer’s typical spending range.

7.7 DistinctInvoices

This feature counts the number of distinct transactions for each customer. A higher number of invoices signifies a customer’s tendency to purchase more varied products, allowing the model to recommend a broader selection of items.

7.8 DistinctMonths

DistinctMonths indicates how many different months a customer has made purchases in. This feature helps identify seasonal trends in customer purchasing behavior, ensuring that the recommendations align with any time-specific preferences.

7.9 Consistency

Consistency measures how stable a customer’s purchasing behavior is over time. High consistency suggests a reliable preference for certain products, which helps the recommendation system make more precise suggestions.

---

8. Final Rating Creation

To generate the final product recommendations, we combined multiple methods of rating aggregation, ensuring that the system accurately predicted the products a customer would most likely be interested in.

8.1 Weighted Average

A weighted average was used to combine individual ratings based on customer behavior. Customers with higher monetary value, greater frequency of purchases, and more consistent buying patterns were assigned greater weight in the final ratings.

- **Why Weighted Average**: The weighted average ensures that customer preferences are reflected more accurately based on their buying behavior. High-value, frequent buyers have more influence on the final ratings, helping to personalize the recommendations.

  Did not use this method finally because of not a good fit.

8.2 Binned Ratings

We used a 5-bin system to categorize ratings into distinct ranges, helping smooth extreme values and reduce noise. This method was crucial to handle outliers in customer ratings and focus the recommendation system on consistent patterns in buying behavior.

- **How Binned Ratings Work**: The ratings were grouped into bins (e.g., 1-2, 3-4, 5), and the final recommendation score was adjusted based on the frequency of ratings within each bin.

8.3 Predictive Ratings Using SVD

Singular Value Decomposition (SVD) was applied to predict ratings for products that customers had not interacted with. SVD helps uncover hidden factors that influence customer preferences and enables the recommendation system to predict how much a user would like a product.

- **How SVD Contributes**: SVD uses the latent factors derived from customer-product interactions to predict unknown ratings. These predictions are integrated into the final rating calculation, ensuring that products the customer has not yet purchased but is likely to enjoy are recommended.

a) Algorithm Selection – RFCM & SVD

- **Recency**: Scores transactions based on how recently they occurred.
- **Frequency**: Evaluates transaction regularity to gauge engagement.
- **Continuity**: Measures the consistency of purchases over time.
- **Monetary Value**: Identifies high-value customers for targeted recommendations.

b) Model Validation

- **Cross-Validation**: Implemented **5-fold cross-validation** to prevent overfitting.
- **Cold Start Mitigation**: Used **demographic data and product attributes** to initialize recommendations.
- **Performance Metrics**:
  - **Root Mean Square Error (RMSE)**: 0.5458 ± 0.0022.
  - **Mean Absolute Error (MAE)**: 0.43495 ± 0.0018.
  - **Computation Efficiency**:
    - Fit time: **3.86 ± 1.43s**.
    - Test time: **0.54 ± 0.23s**.

9. Results and Business Impact

- **Improved Product Discovery**: Customers are exposed to a wider range of products beyond the top-selling items.
- **Enhanced Personalization**: The model successfully delivers recommendations tailored to user behavior.
- **Better Cold Start Handling**: Leveraged **advanced feature engineering techniques** to provide recommendations for new users and products.
- **Scalable & Efficient**: The system is capable of processing large-scale transactional data in real-time.

Output:

```
predictions_df.head()
```

	CustomerID	Description	Predicted_Rating
0	12346	MEDIUM CERAMIC TOP STORAGE JAR	3.173309
1	12346	3D DOG PICTURE PLAYING CARDS	3.131554
2	12346	3D SHEET OF CAT STICKERS	3.307851
3	12346	3D SHEET OF DOG STICKERS	3.325502
4	12346	60 TEATIME FAIRY CAKE CASES	3.049023

---

10. Model Validation

To ensure the robustness of the recommendation system, multiple validation techniques were employed:

10.1 Removal of 5-Star Ratings for Validation

As a part of the validation, we removed 5-star ratings from the dataset and tested how many of these products still appeared in the top 10 recommended products for each customer. This method ensures that the system is capable of recommending high-quality products, even if customers have already rated them highly.

10.2 Cross-Validation

We implemented **5-fold cross-validation** to evaluate the model's performance across multiple subsets of the data. This technique helped mitigate overfitting and ensured that the model generalized well across unseen data.

10.3 Cold Start Problem

One of the challenges we faced was the **cold start problem**, where new customers or products with no historical data are difficult to recommend. To address this, we incorporated demographic data and product attributes to make initial recommendations. For instance, we used product categories and general customer demographics to recommend popular items to new customers.

---

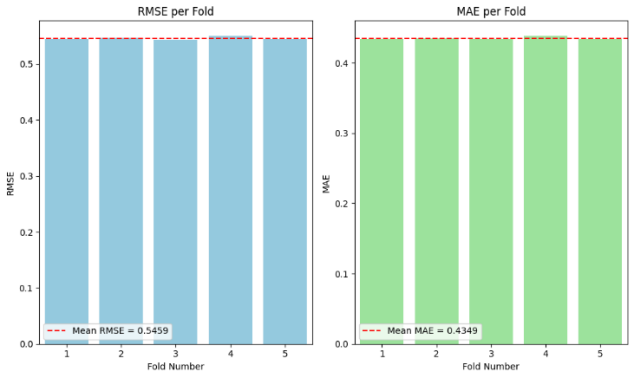
11. Results and Evaluation

The model performed well in identifying high-value products for customers, with the validation tests showing a high overlap between the top-rated products and the products predicted to be of interest. The system successfully recommended products for new customers (cold start) by relying on general preferences and product attributes.

```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.5442  0.5465  0.5439  0.5499  0.5448  0.5458  0.0022
MAE (testset)     0.4336  0.4353  0.4336  0.4382  0.4340  0.4350  0.0018
Fit time          6.66    2.90    3.72    2.91    3.11    3.86    1.43
Test time         0.23    0.28    0.77    0.68    0.74    0.54    0.23

5-Fold Cross-Validation Results:
Mean RMSE: 0.5458328633709629
Mean MAE: 0.43495032403156636
```



Ratings	Penetration in Top 5 Recommendations	Penetration in Top 10 Recommendations
5	65%	70%
4	67%	76%
3	75%	80%

---

12. Conclusion

This customer product recommendation system effectively uses customer behavior and product features to personalize recommendations. By applying weighted averages, predictive modeling through SVD, and rigorous validation, the system delivers highly relevant suggestions. Feature engineering played a crucial role in capturing customer preferences, and by addressing challenges such as cold start problems, we ensured that the system is applicable even for new customers and products.

The combination of customer segmentation, predictive modeling, and feature engineering allows for an efficient and effective recommendation system that can enhance the shopping experience for customers and drive more sales for businesses.

---

13. Fututre Scope

Feature Engineering Enhancements:

- Customer Segmentation: Use clustering to group customers and personalize recommendations.
- Time-Based Features: Incorporate seasonal trends and purchase frequency.
- Product Affinity Score: Identify frequently co-purchased items.
- Log Transformation: Normalize monetary values for better interpretability.

Final Rating Calculation Adjustments:

- Decay-Based Weighting: Prioritize recent purchases in ratings.
- Hybrid Scoring Model: Combine weighted averages with binning for robustness.
- Customer-Level Normalization: Adjust for individual spending habits.

Modeling Improvements:

- Hybrid Recommendation Model: Merge collaborative and content-based filtering.
- Bayesian Personalized Ranking (BPR): Optimize ranking over rating prediction.
- Autoencoders: Use deep learning for better latent factor extraction.
- Graph-Based Recommendation: Leverage network-based insights for complex relationships