Mid-Project Report

Optimal Product Ranking for Improving Customer's Decision-Making

Tarun Khanna tarun19339@iiitd.ac.in **Yugansh Sharu** yugansh19347@iiitd.ac.in **Ritik Aggarwal** ritik19320@iiitd.ac.in
Tanishq tanishq18108@iiitd.ac.in **Saurav Ranjan** saurav19387@iiitd.ac.in

Problem Formulation

E-commerce websites are popular for their wide range of products, but sometimes, it can be difficult for shoppers to locate relevant information, even with user-generated content. Basic ranking methods such as price, rating, and relevance can only do so much in creating a personalized shopping experience. To address this issue, a more individualized system could be introduced, taking into account a variety of factors, including the user's recent browsing and shopping history, the product's characteristics, user interaction with the product page, and micro and macro behaviors. By combining these factors, the system could provide more personalized recommendations, ultimately enhancing the customer's overall shopping experience.

To make the system even more effective, it could incorporate advanced technologies such as artificial intelligence and machine learning. These technologies can analyze vast amounts of data in real-time and predict customer behavior, which can be used to make more accurate and personalized product recommendations. Additionally, the system could use natural language processing to better understand the customer's queries and improve the accuracy of search results.

Introduction

Our approach for creating personalized recommendations requires extensive background information on user interactions and items, which must be added to the system whenever new users or items are introduced. These interactions, also known as micro behaviors, play a critical role in our content-based recommender system, allowing for unique and tailored recommendations for each user. To calculate each user-item interaction score, we consider three basic user interactions: View, Cart, and Purchase. This approach is advantageous as it can overcome the "cold start" problem when there are limited user-item interactions. Although we initially faced limited data availability, our content-based filtering approach still provides reasonably accurate recommendations for users on e-commerce platforms.

"Exploiting Personalized Search for Product Recommendation on E-commerce Platforms" by Jinyoung Yoo, et al. (2019)
This paper proposes a product recommendation system that incorporates personalized search results. The authors used data from an online retailer and a search engine to evaluate the system's performance, and the results showed that it outperformed traditional recommendation methods.

"A Hybrid Product Recommendation Model Based on Collaborative Filtering and Deep Learning" by Zhen He, et al. (2020)
This paper proposes a hybrid recommendation model that combines collaborative filtering and deep learning techniques to make personalized product recommendations. The authors used data from an e-commerce website to evaluate the model's performance, and the results showed that it achieved better performance compared to traditional recommendation methods.

Preprocessing

To build a content-based recommender system for an e-commerce web store, we utilized the "event_type" column of the dataset as the target variable. This column has three categories: View, Cart, and Purchase. To quantify the value of each interaction, we assigned user scores based on the type of interaction. A view was assigned a score of 1, a cart a score of 10, and a purchase a score of 50.

We then categorized items into five price categories relative to their item categories. This categorization was done to account for the price sensitivity of customers and how it affects their purchase decision. The five price categories ranged from 1 to 5, with 1 being the least expensive and 5 being the most expensive.

Since multiple interactions can occur per user per item, we calculated the sum of user scores for each unique user-item interaction using a group by operation. This summation allowed us to capture the total level of interest a user has in a particular item.

To ensure consistency and obtain interaction scores between 0 and 1, we applied MinMaxScaler. This scaling is crucial for normalizing the scores and ensuring that they are comparable across different user-item interactions. An interaction score above 0.5 indicates a high purchase probability, while scores below 0.5 suggest otherwise. These scores are generated for the User-Item Matrix used for model building.

In addition to the user scores and price categories, we also created a separate column for the item categories. This column was created to capture the similarity between items based on their categories. This similarity information is critical in content-based recommender systems as it enables the system to recommend items that are similar to what the user has interacted with previously.

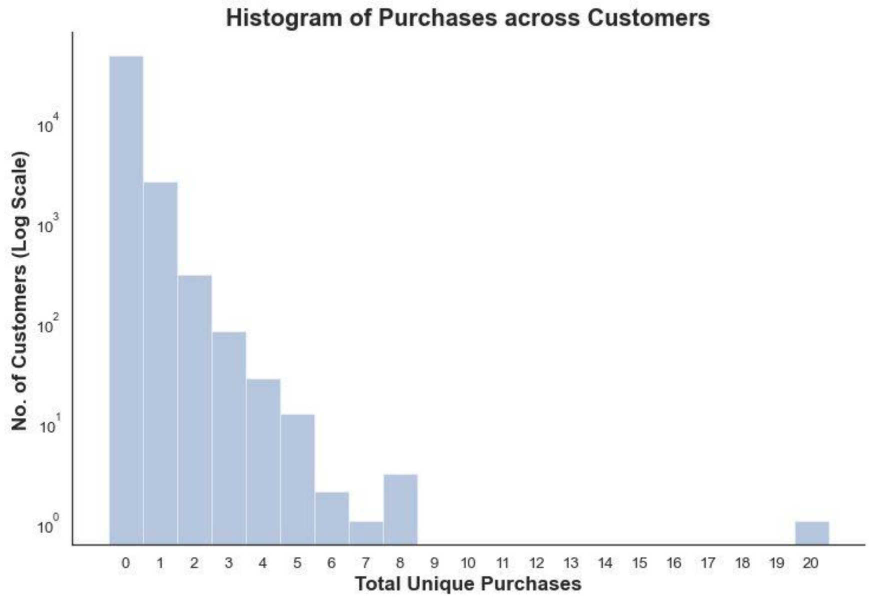
Overall, by using user scores, price categories, and item categories, we were able to create a comprehensive User-Item Matrix that captures the level of interest a user has in a particular item. This matrix can then be used to generate personalized recommendations for each user based on their past interactions and preferences.

Choosing a Model

In the previous model, The most significant step in older model is to visualize the target variable of the recommender system, which includes the categories of View, Cart, and Purchase. The analysis results show that the distribution of the target variable is highly imbalanced, with a significantly low conversion rate of only from view to purchasing.

- **Split the data**:
  To select the appropriate model for the recommender system, we began by dividing the entire dataset into separate training and test datasets using a Simple

Validation approach and split into the ratio of 3:2 of training to test, respectively.



- **Things to Improve upon**
  To improve the older model, several strategies can be considered. Firstly, the class imbalance issue of the target variable should be addressed, as it can result in poor model performance. This can be achieved by oversampling the minority class, undersampling the majority class, or using a combination of both methods. Secondly, incorporating more features beyond the existing ones, such as demographic information or contextual features, could potentially enhance the recommendations. Thirdly, different algorithms, such as matrix factorization or deep learning models, can be explored to compare their performance. Fourthly, a hybrid model that combines multiple algorithms may be more effective than relying solely on one type of recommendation algorithm. Lastly, continuous evaluation and iteration of the model's performance is essential, which could involve testing on new data, conducting A/B testing, and incorporating user feedback to further refine the recommendations.

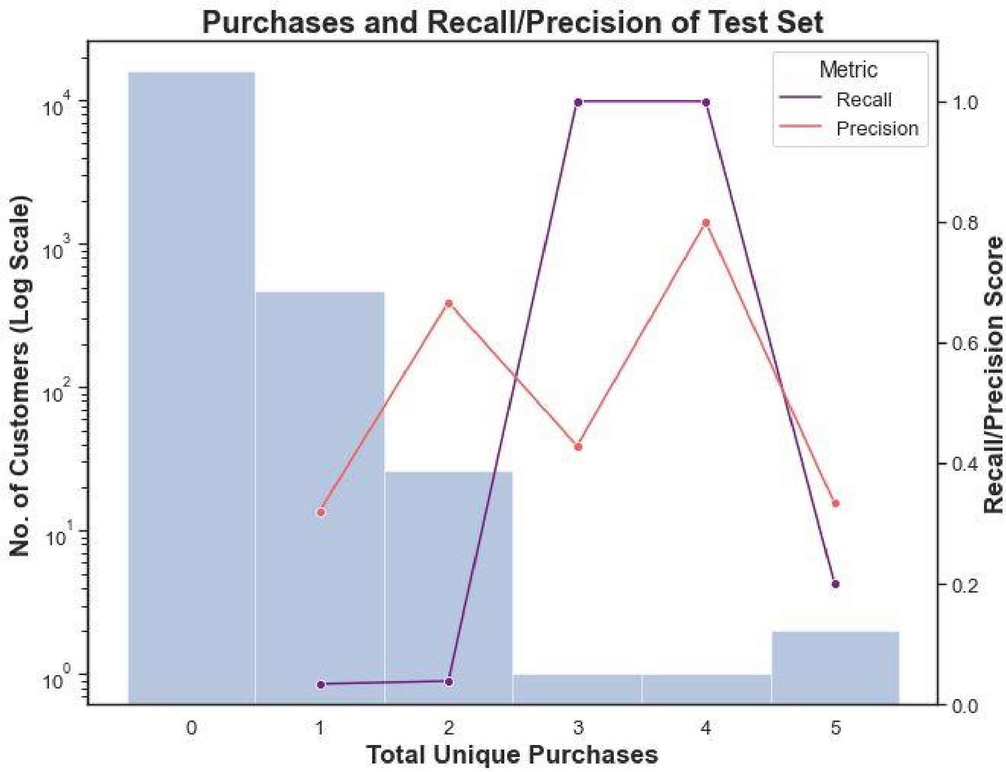
- **Step-by-Step Process of New Model:-**
  - **Data Exploration:** Conduct exploratory data analysis to gain insights into the dataset. Understand the features and their relationships, analyze the distribution of the target variable, and identify any missing values or outliers.
  - **Feature Engineering:** Select relevant features to be used in the model, including user and item features such as age, gender, location, and product category, brand, and price. Transform these features into a format suitable for modeling.
  - **Similarity Matrix Creation:** Create item-to-item similarity matrices based on price, category, and brand. For the price similarity matrix, generate price categories during data preprocessing and calculate the similarity

between each item's price category using a suitable distance metric such as inverse Euclidean. For the category and brand similarity matrices, use cosine similarity to calculate the similarity between items based on their category and brand.
    - **Model Selection:** Select a suitable model for making recommendations, such as collaborative filtering or content-based filtering. Consider using hybrid models that combine both methods for better accuracy.
    - **Model Training:** Train the selected model on the dataset using suitable techniques such as matrix factorization or deep learning. Use cross-validation to avoid overfitting.
    - **Model Evaluation:** Evaluate the model's performance using suitable metrics such as precision, recall, and F1 score. Conduct A/B testing to compare the performance of different versions of the model.
    - **Model Optimization:** Optimize the model based on the evaluation results. Try different algorithms and hyperparameters to improve the model's accuracy and generalization ability.
    - **Deployment:** Deploy the optimized model in a suitable environment such as a web application or mobile app. Continuously monitor and update the model based on user feedback and changing data patterns.

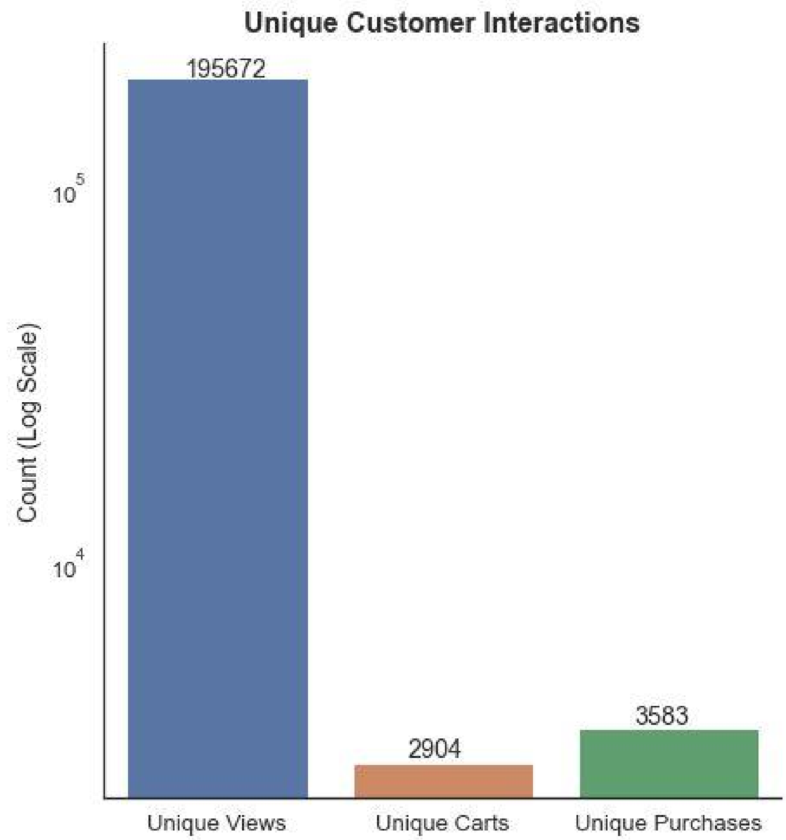
- **Prediction**
  The final matrix is merged with the test dataset. Based on the final similarity score, purchase predictions are calculated. If the similarity score is above 0.5, then the item is considered purchasable.

- **Training the Model**
  An intriguing trend emerged during the evaluation of Precision and Recall based on the number of unique purchases made.

The Precision is higher than the previous model, This Precision of the model is significantly higher for customers who have previously made at least one purchase.



This indicates that when the model suggests a product to a buying customer, there is a greater likelihood that the customer will make the purchase which was not the case in the previous model.



However, false positives of predicted purchases of customers who did not actually buy a product bring down the overall precision of the model.

Conclusion

A content-based recommender system has been developed to provide accurate product recommendations on an e-commerce website. This system is based on multiple categories such as price, brand, category, and product features. By analyzing the characteristics of the items that users have shown interest in, the system can recommend similar products to them.

To evaluate the performance of the system, precision and recall measures were used. The precision of the model is higher for eventual buyers. This means that the system is better at suggesting products that are likely to be purchased by users who have a history of buying items from the website. The precision of the model refers to the accuracy of the recommendations made by the system. A high precision score indicates that the system is providing relevant and accurate recommendations to users.

On the other hand, the recall of the model increases as the customer makes more purchases. The recall is a measure of the effectiveness of the recommender system to identify all relevant products for a particular user. As a customer purchases more items, the system can use this information to improve the recall of the model. This means that the system becomes better at recommending products that the user is likely to be interested in, based on their previous purchases.

However, there are some limitations to content-based recommenders. For example, they are not effective in capturing user preferences for new or less popular items, as they rely on past user behavior to make recommendations. To overcome these limitations, hybrid recommenders that combine content-based and collaborative filtering techniques can be used. These hybrid systems can provide more accurate and personalized recommendations by leveraging the strengths of both approaches.
Overall, the development of a content-based recommender system based on multiple categories has helped to improve the accuracy of product recommendations on an e-commerce website. By considering various item characteristics, the system can provide more accurate and relevant recommendations to users, which can ultimately lead to increased sales and customer satisfaction.

Future Works

- To improve the accuracy of the content-based recommender system, more micro behaviors can be included as features in the dataset. For example, the time of day or day of the week when a user interacts with an item can be included as a feature. This information can help capture temporal patterns in user behavior, such as when they are more likely to make a purchase. Other features such as the device used, the location of the user, and the referral source can also be included to capture more detailed user preferences.
- Future work could involve using a dataset covering a longer period of time. This can provide a more comprehensive understanding of user behavior and preferences over time. It can also help to capture seasonality and trends in user behavior, which can be leveraged to make more accurate recommendations.
- A hybrid recommender system that combines the content-based approach with collaborative filtering or other techniques can be developed using neural networks. This approach can help to overcome some of the limitations of content-based recommenders, such as the cold start problem, where it is difficult to recommend items to new users with no or limited interaction history. By combining multiple techniques, the hybrid recommender can provide more accurate and personalized recommendations to users. Neural networks can be used to learn complex patterns and relationships between users and items, and can be trained using large datasets to improve the accuracy of the recommendations.

References:

He, Z., Zhu, X., & Yang, Y. (2020). A Hybrid Product Recommendation Model Based on Collaborative Filtering and Deep Learning. IEEE Access, 8, 52662-52672. doi: 10.1109/ACCESS.2020.2986075

Yoo, J., Lee, K., Lee, S., & Kim, S. W. (2019). Exploiting Personalized Search for Product Recommendation on E-commerce Platforms. Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), 3149-3154. doi: 10.1109/BigData47090.2019.9006232