Multi-objective Recommender System for E-Commerce using Singular Value Decomposition (SVD) Matrix Factorization Technique.

MSc Research Project
Master's in Data Analytics

Shubham Sunil Rajeshirke
Student ID: X21231036

School of Computing
National College of Ireland

Supervisor: Arjun Chikkankod

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shubham Sunil Rajeshirke ……. ………………………………………………………………………………………………
Student ID:	X21231036 ……………………………………………………………………………………………….……
Programme:	Master's in Data Analytics ……………………………………………………… **Year:** 2022-2023 …………………………..
Module:	MSc Research Project ………………………………………………………………………………………….………
Supervisor:	Arjun Chikkankod ………………………………………………………………………………………….………
Submission Due Date:	14/08/2023 ………………………………………………………………………………………….………
Project Title:	Multi-objective Recommender System for E-commerce using Singular Value Decomposition (SVD) Matrix Factorization Technique ………………………………………………………………………………………….………
Word Count:	8646 ……………………………………… **Page Count** 27 …………………………………………….…….

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………………………………………………………………………………………………………

Date: ………………………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Multi-Objective Recommender System for E-Commerce using Singular Value Decomposition (SVD) Matrix Factorization Technique.

Shubham Sunil Rajeshirke
X21231036

Abstract

In the current digital landscape, personalized recommendations are the key for enhancing user experience across different online platforms. In this research a Recommender system is built using advanced matrix factorization techniques. Leveraging the user behavior data this system will generate 3 recommendations as the output that will align with the users' preferences. After thorough evaluation, the implemented system achieved impressive precision (0.92), indicating accurate item recommendations and Recall (0.64) indicates that the system captures substantial portion of relevant items. The balanced F1-score (0.68) confirms that the system does an excellent job in both accuracy and remembering relevant items. Also, the hit rate (1) shows that the system will be able to engage users consistently with the recommended items. However, the coverage metric (0.36) showcase that there is still room for improvement in the system in terms of exploring more items so that a wider range of user preferences can be catered for. Future directions include exploring more hybrid approaches, making use of more contextual data, and working on the scalability part to address larger datasets. This study highlights the potential of matrix factorization technique in the Recommender system. It shows that factors like engagement, diversification of products and scalability should also be considered for evaluation. The outcome of this research will give the new ecommerce brands a fresh perspective and motivate them to implement an advanced recommender system which will help the brand in enhancing user satisfaction and engagement across online platforms.

1 Introduction

In the current era as ecommerce is evolving, it has become especially important to provide personalized and relevant recommendations to the users to enhance their experience, increasing customer satisfaction and thus getting higher sales. As there are a lot of products and services available on the ecommerce websites and they keep on increasing day by day it is exceedingly difficult for the consumers to select the right fit for them. Hence the Recommendation system has evolved as a tool that helps customers to choose the right with which suits their preference and needs.

1.1 Motivation and Aim

Based on the research, a lot of work is done in this field but as discussed by Wei and his team (2017) in their research the main concern in all the research till date is accuracy and relevance of the recommender system which are crucial. Many start-up companies are struggling in making use of the customer behavior data for the benefit of the customer and the brand as well. The research by Mehta and Rana (2017) says that Matrix factorization techniques are emerging in this field and giving promising results, hence through this research the aim is to design, develop and evaluate a robust recommendation system which will be a custom-fit for upcoming e-commerce platforms using advanced matrix factorization techniques. The proposed system in this report will help to get personalized and accurate product recommendations by harnessing the vast reservoir of user behavior data, such as past purchase history, click through rates, product views and user interactions. The goal of the proposed system is to deliver seamless and engaging user experience that will encourage the users to find new products which will in turn increase their engagement on the platform and boost the conversation rates and sales.

1.2 Research Question

How a multi-objective session-based recommender system built using advanced matrix factorization techniques help E-commerce retailers and individual brands in achieving multiple objectives like user engagement, customer satisfaction, reducing customer churn and increased sales?

1.3 Research Objectives

This report extensively explores different recommendation algorithms and techniques used in e-commerce. The main goal is to find the best approaches that can understand what users like and adjust to their changing interests. The focus will be on collaborative filtering methods, and advanced matrix factorization techniques. By unwinding the latent features from user-item interaction matrix the hidden patterns and relationships will be uncovered that can help in enhancing the accuracy of the recommender system.

Additionally, in this report the difficulties and ethical concerns linked to using recommendation systems in e-commerce websites are discussed. As people become more aware of data privacy and the need for user consent, it is essential to find a proper balance between personalized recommendations and safeguarding user information. Therefore, the research will also investigate ways to reduce biases and make sure that the recommendations given are fair to everyone.

1.4 Structure of Research

This Paper discusses the various methods and models applied for the implementation of recommendation system especially in e-commerce domain till date in section 2 related work. The proposed methodology for this research is discussed in section 3 methodology. In section 4 we will identify and present the methods, structure, and tools used for implementing the recommendation system, along with the necessary requirements. How the entire project is implemented is discussed in section 5 Implementation. In section 6 privacy and ethical concerns related to user data usage and providing recommendations in an e-commerce context are addressed. Finally in section 7 improvements and future directions for the recommendation system based on the research findings are proposed.

2 Related Work

In this section summary of all the research work done in past which is relevant to the research topic are covered.

2.1 Overview of E-Commerce Recommender Systems

Wei and his collaborators (2017) through their research provided an overview of personalized recommendation techniques in e-commerce and focuses on the three main aspects. Firstly, they explored how accurate and efficient customer preferences can be captured using multiple data sources. Secondly, they focused on the limitations of the existing recommender systems where they explained that cold-starting and data sparsity can be tackled using new filtering and hybrid approaches. Thirdly they talk about focusing on other factors like how relevant and clear the recommendations are for evaluation rather than focusing only on the technical side. Finally, their research highlights how the Recommender system impacts customers' opinion and suggests studying how Recommender systems can be useful in business management and Marketing.

The overview presented by Hussein, Rahma, and Wahab (2021) in their research discussed the importance and need of fully functional and efficient E-commerce systems. It emphasizes how important it is to carefully plan and create systems to make them work well for many people, be dependable, safe, and easy for users. Their research explored that recommender systems can be used as a method to provide personalized products to the customer based on their preference which will thus improve the efficiency of e-commerce websites. In their research they analyzed multiple collaborative Recommender system techniques and tried to justify how valuable these systems are in terms of helping customers find suitable products and thus increasing customer turnouts and commercial profits. Their research shows how this recommender system can help in reducing the overwhelming amount of information and make online shopping system work better.

In their study Mohanty and his colleagues (2022) focused on the social side of the recommender system and discussed the three main categories of recommendations: collaborative sorting, content-based recommendations, and Integrated Recommendations. Their research highlights how recommender systems are important in context of E-commerce platforms and how they can be helpful in providing personalized and tailored experience to the customers. They have compared multiple recommendation methods and algorithms and discussed their limitations. Additionally, they proposed a novel approach for evaluation of recommender system using IBM SPSS Text analytics for surveys and semantic information semantics similarity estimations and justified how useful the proposed model is. Overall, their research tells the significance of Recommender systems in enhancing customer satisfaction, trust and purchasing experience in the rapidly growing field of e-commerce.

Schafer and his team (2001) in their research focused on recommender systems and how it helps the customers to find products to purchase and keep them loyal to the brand. They divided the recommender system into multiple categories based on what they use, how they work and what suggestions they give to the customer. They talked about diverse ways these systems can be used and gave ideas about what can be studied in future. They also emphasized some of the crucial factors like privacy and trust with these systems. Overall, according to them businesses can use the right combination of recommender applications to keep the customers happy, make more sales and always stay ahead in the competition.

Ko and colleagues (2022) conducted research wherein they reviewed the trends in recommender systems and focused on the connection between advanced technical aspects and business applications in various service areas. They went through more than 135 top articles and conferences from 2010 to 2021. After going through all this they organized their understandings about recommender systems based on how they work, what is the technology used, where the systems are used. They concluded that the growth of these systems depends upon how businesses use them. They emphasized that using real time data in these systems can give extraordinary results and these systems can make a great contribution in the healthcare sector. This study will help the people who want to learn more about recommender system and how it is evolved also it gives an idea about how these systems can improve in future.

2.2 Mapping Customer Journeys and Personalization

Chen and his colleagues (2023) in their research introduced a novel method to analyze the E-commerce click stream data. Their aim is to understand the intentions of the customer better and improve the marketing precision. In their research they analyzed the patterns in customer browsing behavior and used K-means clustering and decision tree models. They considered each customer session as one unit, which prevented potential variations in the customer's intentions. Their focus was on browsing time for each category to understand the preferences and intentions better. The results of this study state that the approach used was feasible and there is potential for more improvement by exploring the intentions of the customer with a different or new perspective. The study highlights some limitations at the end where it says to add more attributes like browsing paths so that precision can be improved, and retentions of customer can also be mapped. Despite these limitations the study offers a valuable integration of theory and practice that helped in analyzing the customer browsing intentions in e-commerce.

Behera and his team (2020) in their research introduced a model that used Recommender Engine in e-commerce to deliver real-time and personalized marketing information to customers. Their model was a combination of various selling strategies and the cluster of customer, items, and unique selling propositions (USPs) in the e-marketplace. An experimental study done by them for a healthcare retailer gave really good results, like it increased average monthly income by 33.49%, average order value by 32.79% and items per product by 1.93. Through their study they highlighted the focus on customer buying patterns and not assumptions. According to them this strategy will help small-scale industries in understanding how personalized marketing is important to provide good customer experience and a positive increase in business performance as well.

2.3 Matrix Factorization Techniques

In their research, Wang, and his colleagues (2022) introduced a modified recommendation model which was called MFFR. This model addressed the problem of rating sparsity in matrix factorization-based recommender systems. This MFFR model constructs a user-preference matrix by fusing user reviews and ratings. It uses Latent Dirichlet Allocation (LDA) topic model which predicts the ratings and generates top-n recommendations with the help of learning latent factors from both the rating matrix and the preference matrix. They used three different datasets for performing some experiments and they found that their model achieved more accurate predicted ratings and it recommends more correct products in the top-n list in comparison to other traditional models. When using datasets with high rating sparsity their model gave recommendations of high quality.

Mehta and Rana (2017) in their research reviewed the challenges faced by customers because of the large amount of information available on internet and how recommender system can solve this problem and make customer's life easy. According to their research collaborative filtering is an approach used in this case, but it has many limitations like data sparsity, cold start, and high-dimensional data. According to them matrix factorization is a popular technique that uses various auxiliary information such as time and trust for providing better recommendations. Their study showcases how matrix factorization is emerging for implementation of recommender systems and helping in increasing the performance. Also, according to them exploring the fusion of multiple auxiliary information in future can further enhance the effectiveness of the recommendations.

Takacs and associates (2008) in their research investigated how matrix factorization-based methods can be used for improving the prediction accuracy in rating-based recommender systems. They have used multiple matrix factorization models like fast (semi-)positive MF, a momentum-based MF, a transudative MF, an incremental MF, and a hybrid MF-neighbor-based method in their study. They have evaluated these models on a Netflix prize dataset, and they achieved commendable RMSE (Root mean square deviation) scores that very evidently outperforms the previously published approaches. Additionally, the models proposed are efficient as those truly took little time for training and testing highly accurate models. Overall, the researchers emphasize the effectiveness and efficiency of these matrix factorization models that are used for collaborative filtering problems.

Koren, Bell and Volinsky (2009) in their research discussed how important recommender systems are for modern e-commerce and content provider platforms and highlighted that for gaining customer satisfaction and loyalty sending personalized recommendations to the customer is highly crucial. According to them matrix factorization (MF) techniques give more accurate results and are memory efficient in comparison to classical nearest-neighbor techniques for collaborative filtering recommenders. They derived that MF techniques can naturally integrate various aspects of data such as multiple forms of feedback, temporal dynamics, and confidence levels which in turn make them more effective in providing personalized recommendations to the users. According to them these systems can be extremely useful in entertainment products like movies, music and TV shows and gave examples of industry leaders like amazon and Netflix.

2.4 Addressing Challenges in Recommender Systems

Li and his team (2022) in their research focused on a pivotal application of machine learning that aids human decision making which is one of the crucial aspects in recommender systems. They acknowledged that due to biases in data and algorithms there are unfair decisions made hence there is a necessity of addressing this fairness issue in recommender settings. This research explored foundational concepts and definitions of fairness in machine learning while surveying various research on fairness in recommender system. The research then goes deep into the taxonomy of fairness definitions in recommender systems, various methods for enhancing fairness and datasets used in fairness studies. Overall, the research highlighted the challenges, opportunities and importance of fairness research aiming to promote the integration of fairness in real-world recommender systems.

Gope and Jain (2017) in their research explored the cold start problem in recommender systems in which because of insufficient data relevant recommendations can be made to the user. In this research the existing solutions to this problem are reviewed and categorized into explicit and implicit techniques based on how the missing information is collected. Also, the study compares the strengths and limitations of the techniques after categorization. The aim of this research is to provide insights on the cold start problem and highlight the available solutions to it and highlights that there is lot of scope to work on this topic in future. Finally, it says to try a hybrid method by combining implicit and explicit techniques to solve this problem of cold start.

Lam and his colleagues (2008) in their research focused on core techniques in recommender system that includes collaborative filtering and content-based filtering. It is addressing the problem of cold start where it is difficult to recommend because of less data or added items in the list. They proposed a hybrid model that combines collaborative filtering with user information like age, gender, and job to get some improvement in solving the cold-start problem for users. They conducted an experiment on MovieLen dataset where they achieved superior results and were successful in overcoming the user-side cold start problem. However if more attributes are considered, the results can improve. Also, the research suggests that there is potential in solving the item-side cold start problem in future by inverting the direction of ratings data.

2.5 Advanced Techniques and Applications

Adeniyi and his associates (2016) in their research address the problem faced by users in selecting the right product for them from the large pool of choices available on online websites. They proposed an automatic web usage data mining and recommendation system based on user's click stream data on an RSS reader website. To identify user behavior, group them into specific user categories and then provide tailored recommendations to meet the customers' need they used k-Nearest-Neighbor algorithm in real-time. Even when there is limited prior knowledge about the data distribution, KNN classifier was found effective in producing accurate recommendations in real-time. Finally, the research presents a proof-of-concept prototype and suggests performing further research on investigating other data mining techniques for future improvements.

Wei and his collaborators (2017) in their research focused on solving the cold start problem in recommender system in the cases where there is truly little, or no data is available for added items. To address the problems of complete cold start (CCS) and incomplete cold start (ICS) items the researchers proposed two recommendation models that combine collaborative filtering (CF) and deep learning neural network. To extract item content features and predict the ratings for cold start items they used time-aware CF model, timeSVD++ and a deep learning architecture, SDAE. They did experiments on Netflix movie ratings data, and they found that their model outperformed the existing approaches in prediction of cold start items. The researchers said that this solution will enhance the user experience and trust in recommender systems, especially for new and less rated items. According to them further work can be done investigating different system configurations, optimizing the recommendation model, and evaluating using metrics other than RMSE.

Xue and his team (2017) in their research proposed a novel matrix factorization model with a neural network architecture to generate personalized recommendations. To learn low dimensional representations of users and items, the model uses implicit feedback such as non-preference feedback and explicit ratings. To optimize the model with both types of feedback they designed a new loss function based on cross entropy. The experiments done by the researchers on some benchmark datasets gave them effective results and their model outperformed other state-of-art methods. According to them future work can be done on exploring the use of pairwise objective functions using auxiliary data, such as social relations and review text as that will enhance the recommendation system further.

Wang and his co-authors (2022) in their research focused on a new paradigm of recommender system i.e., session-based recommender system (SBRSs) that aim to capture short term and dynamic user behavior data for more timely and accurate recommendations. In this research a comprehensive review is done on SBRs where they included entities like sessions, user behavior data like click, add to cart, or order the items and other properties like session length. They proposed a general problem statement for SBRs that categorizes the existing research, analyzes the session data characteristics and the challenges, and discusses the open issues and future work. Thus, their review will help understand the SBRs in detail, their challenges, and potential areas for future research.

Holy, Sokol, and Cerny (2017) in their research used market basket data to present a data-driven approach for clustering retail products based on customer behavior. Their proposed method aims to categorize products without relying on expert classification by using genetic algorithms to divide products in clusters. They tested this method on synthetic and real data and demonstrated its ability to accurately categorize the products even when certain assumptions are violated. They conclude that this method can reveal hidden subcategories within these clusters and offer valuable insights that will help in decision making processes in marketing and retail. According to them, in future if more data is considered then there are chances of improving the accuracy of method.

Anitha and Patil (2022) in their research focused on applying business intelligence in the retail industry to identify potential customers. This is done by analyzing sales history and purchasing behavior. To segment the customer dataset based on Recency, Frequency and Monetary aspects they used RFM Model and K-means algorithm. Their results showed promising results as they were effectively able to enhance business sales, profits and customer insights that allowed the business to predict the purchasing behavior of the customer. Also, their study highlighted that for both online and physical enterprises customer segmentation and retention is important. According to the researchers, in future work specific product categories can be analyzed and various business parameters can be explored that will help in designing effective strategies and promotions for business enhancement.

Smith and Linden (2017) provided insights about the evolution of Amazon's recommender system in last two decades through their research. Amazon.com is building a store, where based on the customer interests and past behavior it will be personalized for each customer. Their algorithm for item-based collaborative filtering is successful because of its simplicity, scalability, and its ability to provide useful recommendations. Despite of their success the field of recommender system and personalization is still open to create personalized experience for every customer that brings surprise and delight as customer preferences keeps on changing and getting ideal value of 100% accuracy is not possible.

Sharma and Co-authors (2016) in their research explored the development of recommender systems, that involves multiple factors such as Human-computer interaction, marketing, data mining, artificial intelligence, etc. They proposed a new algorithm also called composite search that will combine multiple filtering algorithms to provide personalized results. To offer improved recommendations to the user their algorithm considered both attributes and user ratings. Their study highlights the importance of recommender system in ecommerce emphasizing on its benefits of having significant impact on sales and brand loyalty. According to them in future work user search history should be considered to further enhance the recommendation accuracy.

3 Research Methodology

The Methodology section contains a detailed overview of the methodological approach used for this research. This research aims to follow the methodological approach as shown in Fig 1 below.

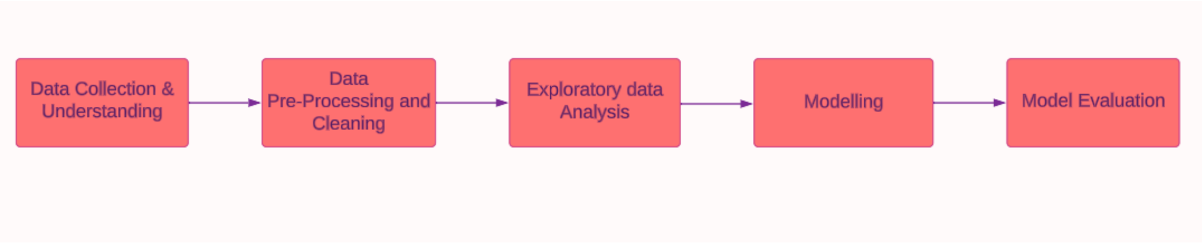

Fig 1: Methodology process followed in this research.

3.1 Data Collection

The Dataset (Otto Multi-Objective Recommender System Dataset) used in this research is taken from Kaggle and it is one of the largest datasets intended to be used for implementation of recommender systems. This dataset contains the data from anonymized behavior logs of the OTTO web shop and the app. For training and test data two different files were already available at the source location of the dataset. The data is in Jsonl format. Below is the description of each column in the data.

- Session: It is unique session id.
- Events: It depicts the timely ordered sequence of all the events in a particular session.
  - Aid: It depicts the product code of the associated event.
  - Ts: It depicts the timestamp of the event.
  - Type: The event type depicts whether the product was clicked, added to the user's cart, or ordered during the session.

The format for the test and train data is the same. In this data there is a possibility that one customer has multiple sessions. Since there is no other Personally identifiable data in the dataset the output of this research is decided to be done on session level i.e., there will be three recommendations per session. If there was customer data present in the dataset, then the output would have been decided at the customer level.

3.2 Data Cleaning & preprocessing

The data used in this research is already clean. There are no null values present in the dataset. Also, there are no duplicates available at the session level.
Some of the steps involved in Data Preprocessing to make the data ready for modelling are as mentioned below.

- **Importing the Data:** Firstly, the dataset is imported in the jsonl format as available from source.
- **Converting jsonl to Json:** These data files are converted into Json format. When the data is in the jsonl format each line contains one or more JSON objects. Hence the

JSON objects in the source files are parsed and then the processed data is saved in a new JSON file. This processed data is stored as a list of dictionaries in the JSON file.

- **Creating Data frames for Train and Test:** Now 2 different data frames are created for training and test data received in the previous step using pandas.
- **Exploding the Data frames 1:** Once the 2 data frames are created, we can see through the head () function that the data in the events column is in the form of dictionary where there are multiple events for a single session. Hence both the data frames are exploded using pandas function "explode ()" so that each row represents an individual event with columns from the original data frame. Now there is only one event in each row and there are multiple rows of the same session. This is done to just normalize the dataset so that it can be used for modelling.
- **Extracting the user-item interactions:** In this step the Train and Test Data frame is again exploded, then the values from the event dictionary are extracted and assigned to new columns. Now we must just select the necessary columns required for modelling in the data frame.

3.3 Exploratory Data Analysis (EDA)

In this section of the report a detailed analysis of the data has been done based on session-based data. This will help in understanding the data more, which will contribute to implementing the model properly.

Statistics for Train Data	
Total Unique Sessions	5000
Total Number of Products	19637
Total Number of Events	28722
Total Number of clicks	25982
Total Number of Carts	2335
Total Number of Orders	405

Fig 2: Statistics for Train Data

In Fig 2 above statistics for train data have been calculated. Metrics like Total number of sessions, Total number of products, Total number of events, Total number of clicks, Total number of carts and Total number of orders are covered in this.

Statistics for Test Data	
Total Unique Sessions	116
Total Number of Products	844
Total Number of Events	1517
Total Number of clicks	1276
Total Number of Carts	2271
Total Number of Orders	398

Fig 3: Statistics for Test Data

In Fig 3 Statistics of Test data is calculated as done for Train in Fig 1. There are other parameters like session, events, products, timestamp available in the dataset. So based on these some more visualizations have also been made to get a better understanding of the data.

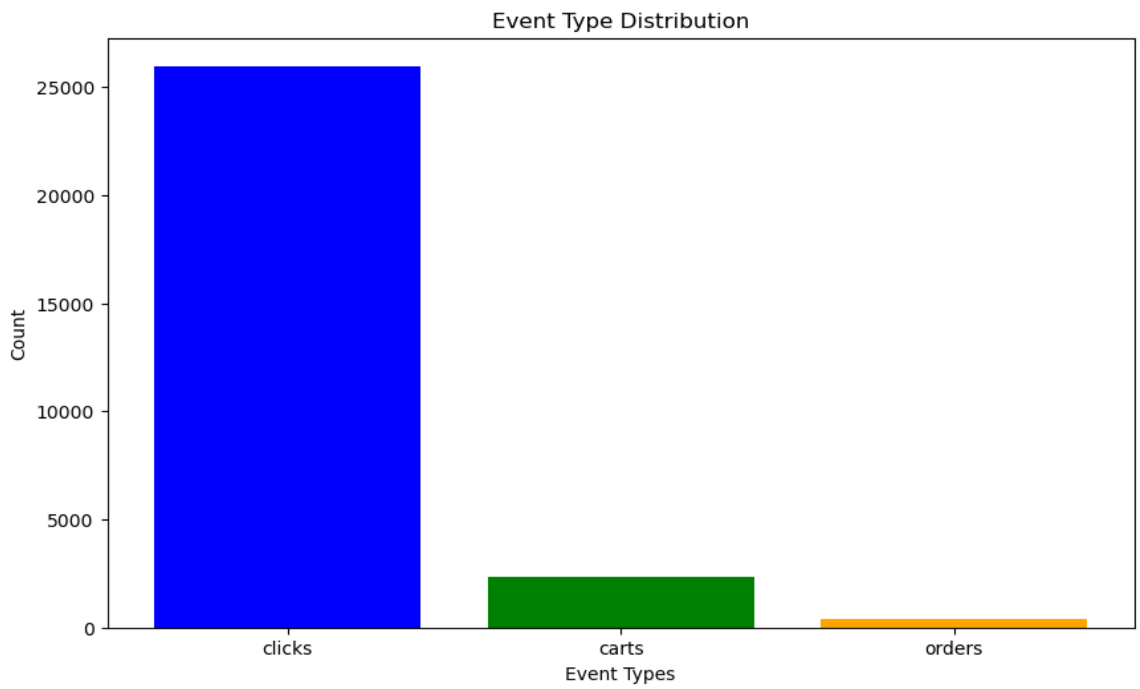

Fig 4: Bar Plot for Event type

Fig 4 depicts the distribution of event type in the Train data. We could see the maximum number of events in the dataset is of Clicks type which is followed by carts and orders.

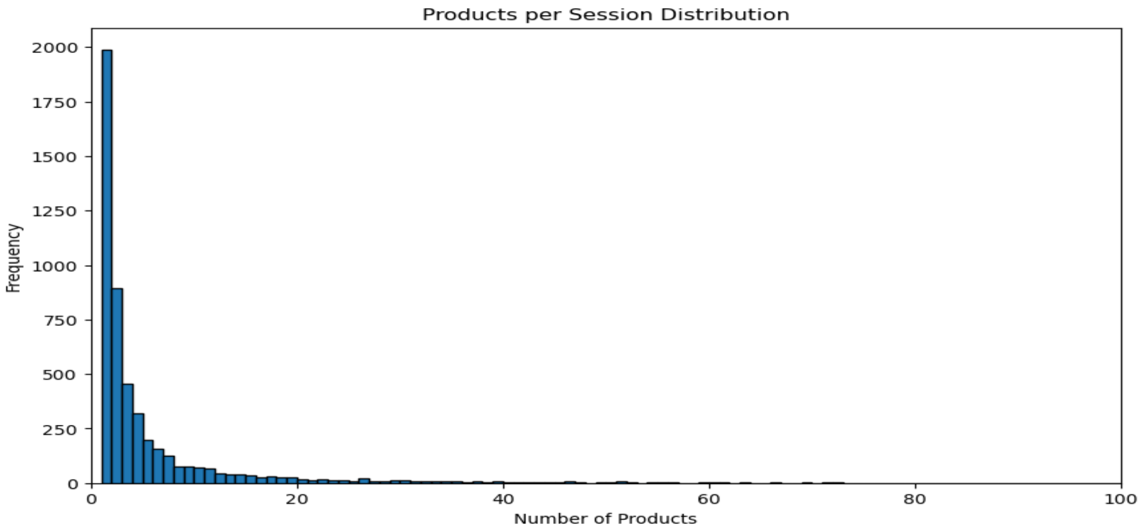

Fig 5: Histogram for Products per session

In Fig 5 we can see the distribution of products per session. This can also be depicted as events per session because there is one product in each event. Looking at the histogram it can be said that most of the sessions have 1-50 products as we can see that there are very small bars after 50 on the x-axis which depicts the number of products. According to the stats the minimum number of events per session is 1 and the maximum number of events per session is 241.

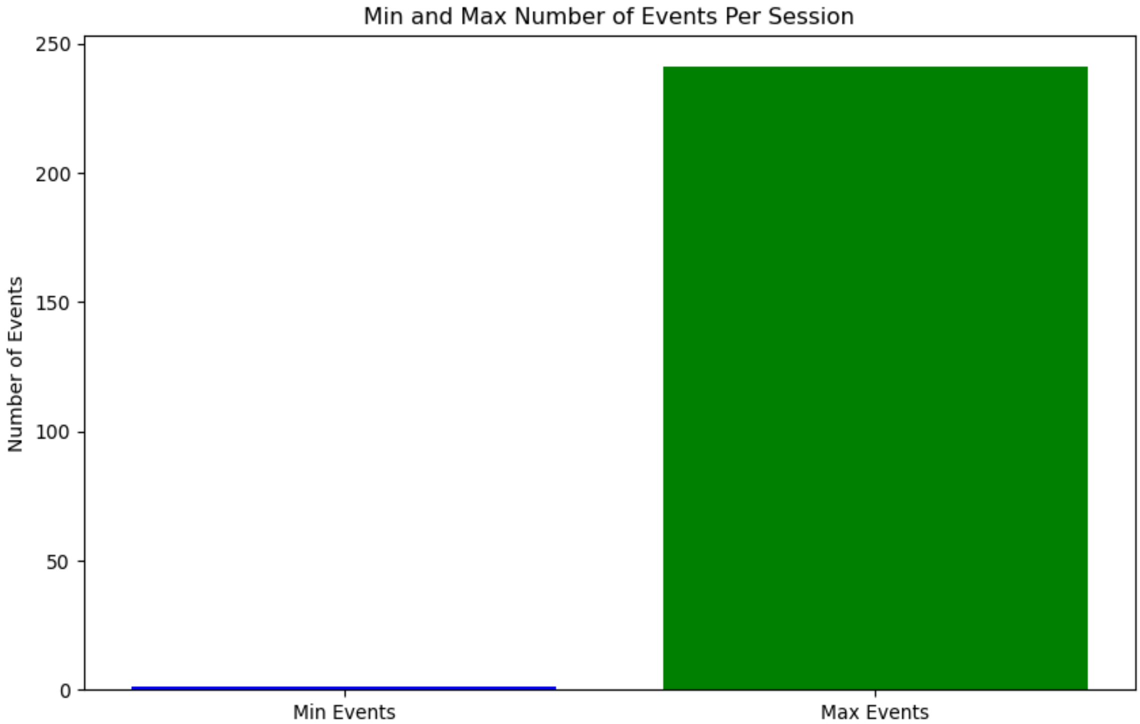

Fig 6: Min and Max number of events per session

According to Fig 6, the minimum number of events per session is 1 and the maximum number of events per session is 241.

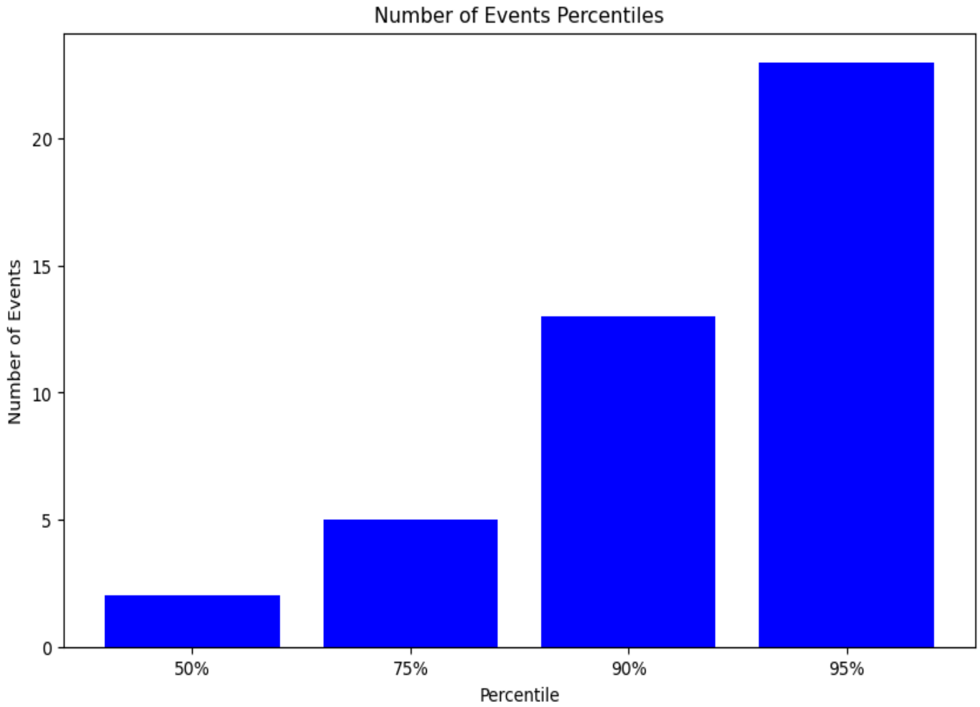

Fig 7: Number of Events Percentiles

Fig 7 above describes the insights into the distribution of the number of events pers session in the Train data. On average each session has around 5.74 events with a standard deviation of 12.76. This indicates the variability in the number of events. The median (50^{th} percentile) number of events pers session is 2 which suggests that half of the sessions have two or fewer events. The 75^{Th} percentile indicates that 75 % of the sessions have up to 5 events. The 90^{Th} percentile, at 13 events indicates that 90% of sessions have 13 events or fewer. The 95^{Th} percentile value of 23 events or fewer. These statistics provide an overview of the distribution and variability of events across sessions.

3.4 Modelling

In this section we will deep dive into how Singular value Decomposition Matrix Factorization Technique is been used to implement a recommender system that gives 3 products per session as the output.

Following steps are taken while implementing this model:

a) Basic data preprocessing:

In this step we again have done some operations on the data which was bought into normalized form in step 3.1.

- Here firstly the "ts" column is converted to a datetime format so that a standardized way can be used to work with timestamps.
- Weights are assigned to each event type ('orders', 'carts', 'clicks'). Maximum weight is given to orders followed by carts and clicks. This is done to give more importance to 'orders' and 'carts' in comparison to 'clicks' which will help in generating recommendations in the later stage.
- A ground truth Data Frame is also created where each row represents a session, and the 'actual_products' column contains a list of unique products that were visited by the customer in that session. This data frame will be used

in evaluation of the recommendations generated by the recommender system at the end.

b) Creating user-item interaction matrix
In this step group by is done on ‘session’ and ‘aid’ (product id), and the weights for each combination are summed up, aggregating the duplicate entries for the same product within a session. This will give an interaction matrix where rows will represent the sessions and columns represent products and the values will be the aggregated weights which indicate user interactions.

c) Applying Singular Value Decomposition (SVD) Matrix Factorization technique:
When talking about Singular Value Decomposition Technique there are two types of it. One is Normal SVD and the other is SVD++. SVD is nothing but a linear algebra technique that decomposes the matrix into three matrices: U, Sigma and Vt. This factorization will help in understanding the patterns in the data and reduce the dimensionality. Normal SVD is mostly used where the dataset is not so big, and it is usually known for its simplicity. SVD works well when users directly tell the system what they like or don't like about items. It provides clear and easy-to-understand recommendations, whereas SVD++ deals with indirect hints from users and other complicated factors, but our dataset didn't need all that complexity. The simpler SVD was just right because it balanced making accurate recommendations with being easy to use. It matched the goal of the project and the data that is available. Hence finally in this step Singular value decomposition is used to factorize the user-item interaction matrix.

d) Choosing the latent features (k):
In this step the number of latent features (k) is set during matrix factorization. This value of k will represent the underlying patterns that will in turn help us to approximate the original interaction matrix. The value of k is subjective and can be changed as per need. We have used different values of k for getting better results and the same will be discussed in the evaluation part in detail.

e) Reconstructing the user-item matrix:
In this step the user-item interaction matrix is reconstructed using the reduced dimension representation obtained after performing SVD. This matrix will then provide estimates of user-item interactions based on the chosen latent features.

f) Generating Recommendations:
In this step a function “get_top_n_recommendations(session, n=3)” is defined. This function takes session ID as input and returns the top n (which is 3 in our case) recommendations (products IDs) for that session. This function firstly sorts the products based on their estimated interaction scores in the descending order and then returns the top 3 product IDs as the output.

g) Main Recommendation Generation:
In this step each unique session is been iterated and the function ‘get_top_n_recommendations()’ function is called to get the top 3 recommendations

for each session. These recommendations are then stored in a dictionary where the keys are session IDs, and the values are lists of recommended product IDs.

h) Displaying the results:
Finally, a loop is run through the dictionary that contains the recommendations and the top 3 recommended product IDs for each session are printed in the desired format.

Overall, in this section mainly the user-item interaction matrix represents how different customers interact with different set of products and matrix factorization helps in identifying patterns and generating personalized recommendations based on those patterns.

3.5 Evaluation

In this section performance metrics like precision, recall, F-1 score, hit rate and coverage is used for evaluation purpose of the recommender system.

a) Precision
It is the ratio of true positive predictions to the total number of positive predictions made by the model. It basically helps in measuring the accuracy of the positive predictions made by the model.

$$\text{Precision} = \frac{True\ Positive(TP)}{True\ Positive(TP)+False\ Positive(FP)}$$

b) Recall
It is the ratio of true positives predictions to the total number of actual positives I.e., ‘actual_products’ from ground truth data frame in this case. This basically helps is measuring the ability of the model in identifying all the relevant instances.

$$\text{Recall} = \frac{True\ Positive(TP)}{True\ Positive(TP)+False\ Negative(FN)}$$

c) F1-score:
It is the harmonic means of precision and recall. It basically provides a single metric that helps in balancing both precision and recall. It is useful when precision and recall have imbalanced importance.

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision+Recall}$$

d) Hit Rate
It measures the percentage of sessions where at least one recommended product is from the list of actual products in the session.
If the value of the hit rate is closer to 1 it indicates that for most of the sessions the users are getting the recommendations, they are interested in.

e) Coverage
It measures the proportion of unique products in the catalog that are being recommended across all sessions.
If the coverage is higher, it indicates that the recommender system is recommending a wide range of products.
There is no ideal value for coverage because it is based on the nature of the catalog and user behavior.

4 Design Specification

In this section the study models used in the research are explored. This research is performed in Python with Jupyter Notebook. Below are some of the libraries used in this study.

a) JSON: It is a library to work with JSON data. It allows to encode and decode JSON objects.
b) Pandas: It is a versatile data manipulation library that provides data structures and data analysis tools.
c) NumPy: It is a package used for scientific computing in python. It provides support for arrays, matrices, and mathematical functions.
d) TensorFlow: It is an open-source machine learning framework that provides multiple tools, libraries, and resources to build, train and deploy machine learning models.
e) Tensorflow.keras.models.Sequential: It is a part of TensorFlow library. This module helps in creating sequential neural network models.
f) Tensorflow.keras.layers.Embedding: It is a layer used for embedding categorical variables, that are commonly used in Natural language processing tasks.
g) Tensorflow.keras.layers.Dense: It is a fully connected layer in a neural network architecture.
h) Tensorflow.keras.layers.LSTM: It is a type of recurrent neural network layer used for sequence modelling task.
i) Tensorflow.keras.preprocessing.sequence.pad_sequences: It is a function to pad sequences to a specified length and it is commonly used in sequence data preprocessing.
j) Tensorflow.keras.utils.to_categorical: It is a function that helps in converting the categorical data into one-hot encoded format.
k) Sklearn.model_selection.train_test_split: It is a function that helps in splitting the data into training and testing for the evaluation of machine learning models.
l) Sklearn.metrics.pairwise.cosine_similarity: It is a function that helps in computing the cosine similarity between pairs of vectors.
m) Matplotlib.pyplot: It is a plotting library that helps in creating visualizations and plots.
n) Seaborn: It is a statistical Data Visualization library that is built on the top of matplotlib. It provides a high level of interface for attractive and informative visualizations.

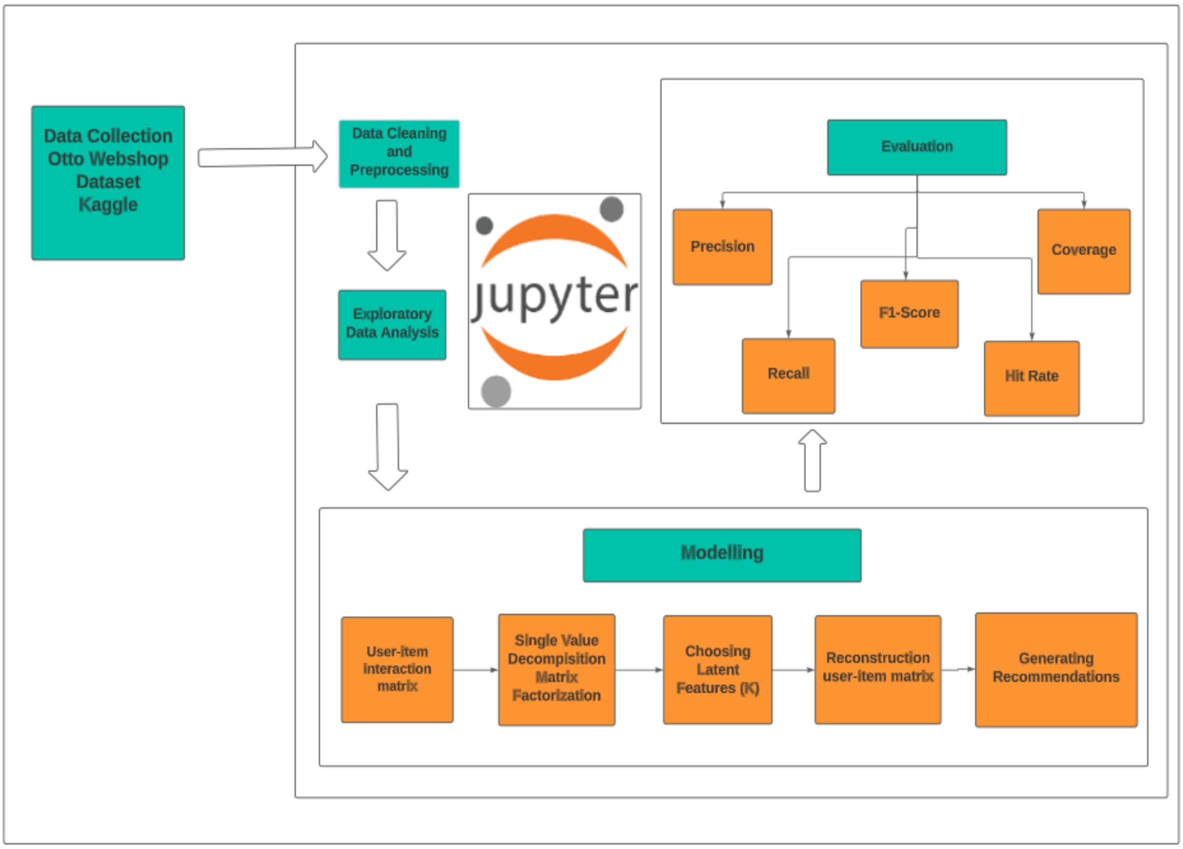

Fig 8: Architectural Diagram for proposed Recommender System.

The above figure depicts the architecture diagram of the proposed recommender system. The process followed in this architecture is as follows:

a) Firstly, the data is acquired from Kaggle, and some basic cleaning and preprocessing is done on the data to normalize it into a usable format for applying models.
b) Then some exploratory Data analysis and visualizations are done to better understand the patterns in the data.
c) After that the proposed modelling is applied on the dataset where user-item interaction matrix is created, SVD model is applied, latent features (k) is selected, user-item interaction matrix is reconstructed and finally the recommendations are generated.
d) At the end the model evaluation is performed using evaluation metrics like precision, recall, F1-score, Hit Rate and Coverage.

5 Implementation

In this section the Modelling part of the Research methodology and its results at each stage are discussed in detail. The overview and the steps involved in the actual implementation are already discussed in short in the methodology section.

```
In [447]: train_df.head()
```

Out[447]:

	session	events
0	12899779	[{'aid': 59625, 'ts': 1661724000278, 'type': '...
1	12899780	[{'aid': 1142000, 'ts': 1661724000378, 'type':...
2	12899781	[{'aid': 141736, 'ts': 1661724000559, 'type': ...
3	12899782	[{'aid': 1669402, 'ts': 1661724000568, 'type':...
4	12899783	[{'aid': 255297, 'ts': 1661724000572, 'type': ...

Fig 9: Sample records of actual data

In the above Fig 9, we could see that there are 3 columns in the dataset where we have index, session represents the session id and events represents the number of events taken place in that session. We could see that multiple events taking place in a particular session are stored in the form of list of dictionaries under events column. Here each dictionary represents an event.

```
In [472]: final_df_train.head()
```

Out[472]:

	session	aid	ts	type
1	12899780	1142000	1661724000378	clicks
1	12899780	582732	1661724058352	clicks
1	12899780	973453	1661724109199	clicks
1	12899780	736515	1661724136868	clicks
1	12899780	1142000	1661724155248	clicks

Fig 10: Sample Data after normalizing the data (using 'explode ()' function)

In the modelling section we talked about using the "explode ()" function to normalize the events column and after using that we get the resulting data frame as shown above in Fig 10. Here in this data frame, we now have unique events of same session in different rows and more columns are added to the data frame named as ts and type after exploding the dictionaries in the previous data frame.

```
In [490]: session_product_weights.head()
```

Out[490]:

	session	aid	weight
0	12899780	582732	1
1	12899780	736515	1
2	12899780	973453	1
3	12899780	1142000	2
4	12899781	57315	1

Fig 11: Sample Data after assigning weights to events.

In the modelling section we also discussed assigning weights to events where maximum weight is assigned to orders, followed by carts, and clicks. So, after performing that operation we get the resulting data frame as shown in Fig 11. This data frame is now ready to apply the actual models to get the desired output in the form of recommendations.

```
In [492]: # Display the ground truth DataFrame
          print(ground_truth_df)

           session                                    actual_products
0       12899780                 [582732, 736515, 973453, 1142000]
1       12899781             [57315, 141736, 194067, 199008, 918667]
2       12899782  [45034, 127404, 229748, 363336, 406001, 413962...
3       12899783  [198385, 255297, 300127, 607638, 1114789, 1216...
4       12899784  [22981, 476216, 1036375, 1190477, 1269952, 154...
...          ...                                                ...
2112  12904768               [886392, 1237853, 1475121, 1591574]
2113  12904770  [181414, 454926, 479970, 717779, 1200925, 1494...
2114  12904772  [2732, 14449, 239801, 543928, 1045144, 1082732...
2115  12904773                 [68526, 252695, 457060, 1683249]
2116  12904776                                  [247240, 954193]

[2117 rows x 2 columns]
```

Fig 12: Sample Data of ‘ground_truth_df’ data frame.

We discussed creating a ‘ground_truth_df’ data frame that will be useful in evaluation purpose of the results. The sample data for the same is as shown in Fig 12. The ‘actual_products’ column in this data frame represents the products that were part of that session.

```
In [494]: user_item_matrix.head()
Out[494]:
```

aid	38	114	160	240	284	691	1246	1249	1514	1612	...	1854329	1854421	1854499	1854540	1854665	1854762	1854775	1855264	1
session																				
12899780	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12899781	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12899782	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12899783	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12899784	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

5 rows × 14068 columns

Fig 13: Sample Data of user-item matrix

Once the actual data frame was normalized and the final data frame is generated after weight assigning the next steps was creating a user-item matrix and applying Singular Value Decomposition (SVD) matrix Factorization. So, Above is the sample of the data from user-item interaction matrix. Here we have session on the rows and products on the columns and the values are the aggregated weights that we assigned.

```
In [523]: user_item_df.head()
```

Out[523]:

aid / session	38	114	160	240	284	691	1246	1249	1514	1612	...	1854329
12899780	-5.511947e-09	-4.989551e-12	4.304837e-08	9.878557e-20	7.757214e-13	-1.004171e-08	-3.263617e-19	-1.105088e-08	3.029552e-19	-8.561736e-25	...	-4.764418e-24
12899781	1.943951e-08	-5.005362e-11	-4.375313e-07	5.743307e-17	3.529891e-11	-1.183943e-05	-1.992068e-18	7.939758e-11	-1.123346e-17	1.896684e-24	...	4.792519e-18
12899782	3.116410e-08	7.397697e-12	-6.087782e-08	-2.900878e-18	-3.798416e-12	3.356809e-07	-2.147135e-19	-5.452430e-11	-5.376803e-18	-8.294426e-26	...	2.970826e-20
12899783	1.595107e-17	-2.286391e-16	4.931353e-17	5.188972e-17	-1.721346e-16	-2.393055e-17	-5.846312e-18	5.579925e-17	-4.875116e-17	1.295784e-24	...	2.498061e-19
12899784	2.533114e-17	1.680562e-17	6.062375e-17	1.688208e-17	5.531468e-17	5.804031e-16	-6.800324e-18	1.675094e-16	-1.749356e-16	2.261214e-24	...	-1.185203e-17

5 rows × 14068 columns

Fig 14: Sample Data from Reconstructed user-item interaction matrix

The next step after applying SVD was choosing the latent features and reconstructing the user-item interaction matrix. This is done using reduced dimension representation obtained after performing SVD. Fig 14 above shows the sample data of that data frame.

```
In [502]: # Display the final recommendations for each session
          for session, recommendations in all_recommendations.items():
              print(f"Session {session}: Rec1: {recommendations[0]}, Rec2: {recommendations[1]}, Rec3: {recommendations[2]}")

Session 12899780: Rec1: 1142000, Rec2: 973453, Rec3: 736515
Session 12899781: Rec1: 199008, Rec2: 57315, Rec3: 141736
Session 12899782: Rec1: 779477, Rec2: 834354, Rec3: 595994
Session 12899783: Rec1: 255297, Rec2: 300127, Rec3: 1817895
Session 12899784: Rec1: 476216, Rec2: 1190477, Rec3: 22981
Session 12899785: Rec1: 453905, Rec2: 258458, Rec3: 41655
Session 12899787: Rec1: 1682750, Rec2: 1024433, Rec3: 1340855
Session 12899788: Rec1: 1259911, Rec2: 39846, Rec3: 245131
Session 12899789: Rec1: 1569899, Rec2: 631398, Rec3: 525156
Session 12899790: Rec1: 1219653, Rec2: 1830166, Rec3: 1735585
Session 12899791: Rec1: 915175, Rec2: 1305729, Rec3: 1501214
Session 12899792: Rec1: 1537160, Rec2: 132681, Rec3: 1672169
Session 12899793: Rec1: 1585431, Rec2: 808362, Rec3: 1132001
Session 12899798: Rec1: 1344706, Rec2: 558573, Rec3: 99354
Session 12899799: Rec1: 413826, Rec2: 1325402, Rec3: 1191715
Session 12899801: Rec1: 1750538, Rec2: 812857, Rec3: 1645990
Session 12899803: Rec1: 929024, Rec2: 1651971, Rec3: 1684753
Session 12899805: Rec1: 22940, Rec2: 967498, Rec3: 199008
Session 12899807: Rec1: 1407421, Rec2: 200257, Rec3: 409527
```

Fig 15: Getting the final Recommendations as the output.

After the Re-construction of user-item interaction matrix the final step was defining and calling the ‘get_top_n_recommendations()’ function to get the final output in the form of top 3 recommendations. Fig 7 above showcases the glimpse of the achieved output.

In this way in the overall implementation, matrix factorization is applied on user-item interaction data using SVD, Recommendations are generated, and the top 3 recommended products are displayed in the required format.

6 Evaluation

In this section the findings of the research are explored in detail. As we have already discussed in the methodology section that we will use precision, recall, f1-score, hit rate and Coverage for the evaluation purpose of this research we will see the results of each parameter in detail below:

```
In [69]: # Display the evaluation results
         print(evaluation_df)
         print(f"\nAverage Precision: {average_precision}")
         print(f"Average Recall: {average_recall}")
         print(f"Average F1-score: {average_f1_score}")
```

```
       Session  Precision    Recall  F1-score
0     12899780   1.000000  0.750000  0.857143
1     12899781   1.000000  0.600000  0.750000
2     12899782   1.000000  0.078947  0.146341
3     12899783   1.000000  0.333333  0.500000
4     12899784   1.000000  0.428571  0.600000
...        ...        ...       ...       ...
2112  12904768   1.000000  0.750000  0.857143
2113  12904770   1.000000  0.375000  0.545455
2114  12904772   1.000000  0.300000  0.461538
2115  12904773   1.000000  0.750000  0.857143
2116  12904776   0.666667  1.000000  0.800000

[2117 rows x 4 columns]

Average Precision: 0.9195402298850599
Average Recall: 0.6413722633807162
Average F1-score: 0.6817285135355987
```

Fig 16: Results for Precision, Recall and F1-score.

a) Precision

The formula for precision based on our case can be formulated as precision: true_positives / len(recommended_products).
In this case the true positives are the products that are in both the lists, actual products, and the recommended products. In this case dividing the number of true positives by the total number of recommended products will give the proportion of correct recommendations among all the recommendations made.
We are getting the value of Precision as 0.9195. The ideal value is 1 which means the if the value is 1 or close to 1 there is more correctness in the recommended products.
So, 0.9195 is good value which states that around 91% of the products recommended are correct.

b) Recall

The formula for Recall based on our case can be formulated as Recall: true_positives / len(actual_products).
In this case the true positives are again the same i.e., the products that lie in both actual and recommended products.
It is calculated by dividing the number of true positives by the number of actual products. This will give the proportion of relevant products that were successfully recommended.
In our case we got the recall value as 0.64. According to the research the ideal value for recall is also 1 but any value of recall above 0.5 is a good value in case of recommender systems.
So, 0.64 as Recall value is a good one as it describes that 64% of products that were recommended were relevant products as per the customer choice.

c) F1-score

It is a single metric that helps in balancing both precision and recall.

In our case F1-score is: 2 * (precision * recall) / (precision + recall).

F1-score helps in penalizing the extreme values of either of the metric between precision and Recall.

According to the research any value above 0.5 for F1-score is good value for recommender systems. We are getting the F1-score value as 0.68 which is a good and acceptable value and balances both the Precision and Recall values perfectly, hence showcasing the overall effectiveness of the Recommender System.

```
In [74]: # Calculate hit rate and coverage
coverage = len(recommended_products_set) / total_unique_products

print(f"Hit Rate: {hit_rate}")
print(f"Coverage: {coverage}")

Hit Rate: 1.0
Coverage: 0.36174296275234574
```

Fig 17: Results for Hit Rate and Coverage

d) Hit Rate

It is the percentage of sessions where at least one recommended product is from the list of actual products. As we can see in the above image, we have achieved a hit rate of 1 which is the ideal value. This indicates that our system generates recommendations in which the customer is interested in.

e) Coverage

It calculates the proportion of uniqueness in terms of products that are being recommended in all sessions. If the coverage is higher, it indicates that the recommender system is recommending a wide range of products. In Fig 17 we can see that the value of coverage we are getting is 0.36. There is no ideal value for coverage as such because it totally depends upon the range of products available in the data and the user behavior. Hence 0.36 is acceptable value.

7 Discussion

In this section the results of the Research are discussed in detail.

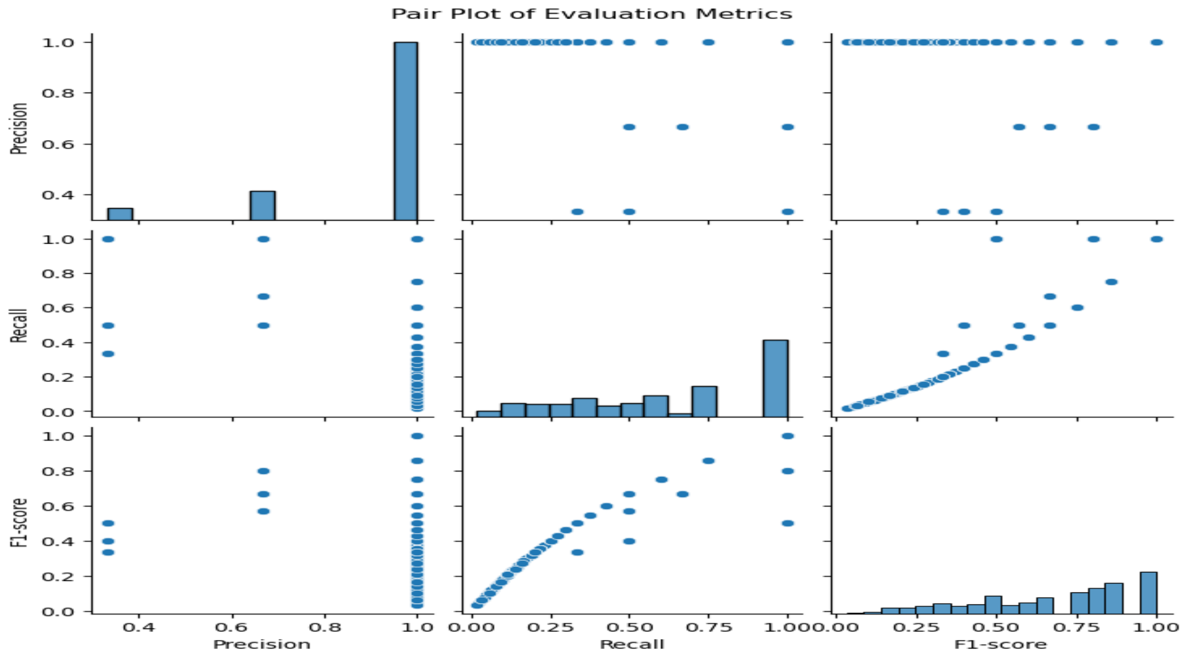

Fig 18: Pair plot for Evaluation Metrics

The above Fig 18 depicts the Pair plot for the 3 Evaluation Metrics. Looking at the Figure we get some of the insights regarding the relation between the 3 metrics as discussed below:

Precision vs Recall: The scatter plots in the upper triangle showcase the relation between precision and Recall. We could see that there is no positive relation between precision and recall as when the precision is high the recall is low in some cases but in some cases, there is a positive relation as well. So, we could say that the recommender system can work well as currently there are mixed results when comparing the precision and recall metrics.

Precision vs F1-score: The scatter plots in the lower left triangle showcase the relation between precision and F1-score. In the plot we can see that when the precision increases the F1-score also increases. This proves that there is a good balance between precision and recall in the generated recommendations.

Recall vs F1-score: The scatter plots in the lower right triangle showcase the relationship between Recall and F1-score. In the plot we can see that as the recall is increasing the F1-score is also increasing. This indicates that there is a good balance between precision and Recall.
In conclusion we can see that the precision value achieved is very good, but the recall value is slightly low. This indicates that even if the system generates accurate recommendations there are chances that it might miss some relevant recommendations as well. The balance is referenced in the F1-score which has a value between the values of other 2 metrics and hence the system achieves a very good balance between Precision and Recall.

8 Conclusion and Future Work

In this study a robust recommender system using advanced matrix factorization technique has been successfully implemented. With the help of user behavior data, the system provides personalized recommendations that will help in enhancing the user experiences across various online platforms. The effectiveness of our approach is highlighted by the evaluation done in the research. A very high precision of 0.92 indicates that the system can accurately suggest the items that are relevant to the users. The commendable Recall value of 0.64 indicates the success of the system in getting a significant number of relevant items for the users. Furthermore, the F1-score of 0.68 indicates a well-rounded performance and balanced tradeoff between precision and recall. The hit rate of 1 is quite impressive and indicates that the user is getting recommendations from their searches and this success of the system helps in driving user engagement. However, we achieved the coverage as 0.36, which indicates that there is room for improvement in the system and ensuring that a wide range of products are recommended to cater to the users' preferences. This might be done using smarter ways to offer a wide variety of products.

While the current system gives promising results, there is room for further enhancement in this system. In terms of methodology other hybrid methods where many other techniques can be merged with matrix factorization, such as content based filtering or deep learning can be used for more accurate and diverse recommendations. If there are more columns in the data which give information about time of the day or user location, better recommendations can be generated to enhance the experience of users. The coverage metric of the system indicates that there is a need for a more comprehensive item list. Future works can make use of external data sources or can employ techniques like item cold start handling to fulfill this limitation. Additionally, considering other multiple factors like how happy users are, how long they use the product, are they going to use the same product over time, etc. would give a better picture of the system's success. Furthermore, scalability is another factor that should be considered for effective handling of large datasets and providing real time recommendations in the future. For this more research should be done on distributed computing frameworks and optimization techniques.

To sum up, the research shows that using matrix factorization for implementing the Recommender system works well. In the future works by handling the issues like showing variety of options and combining different methods can help the users find the products of their choice. This will make the online experience of the users better and satisfying.

References

Chen, S. S. et al. (2023) "An Algorithm-based approach for Mapping customer journeys by identifying customer browsing behaviors on E-commerce Clickstream data," in 2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, pp. 1–8.

Li, Y. et al. (2022) "Fairness in recommendation: Foundations, methods and applications," arXiv [cs.IR]. Available at: http://arxiv.org/abs/2205.13619.

Wang, H., Hong, Z. and Hong, M. (2022) "Research on product recommendation based on matrix factorization models fusing user reviews," Applied soft computing, 123(108971), p. 108971. doi: 10.1016/j.asoc.2022.108971.

Hwangbo, H., Kim, Y. S. and Cha, K. J. (2018) “Recommendation system development for fashion retail e-commerce,” Electronic commerce research and applications, 28, pp. 94–101. doi: 10.1016/j.elerap.2018.01.012.

Hussien, F. T. A., Rahma, A. M. S. and Wahab, H. B. A. (2021) “Recommendation systems for e-commerce systems an overview,” in Journal of Physics: Conference Series. IOP Publishing.

Sharma, S. et al. (2016) “Recommender system using hybrid approach,” in 2016 International Conference on Computing, Communication and Automation (ICCCA). IEEE.

Mohanty, L. et al. (2022) “Recommender Systems in E-Commerce,” in 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT). IEEE.

Adeniyi, D. A., Wei, Z. and Yongquan, Y. (2016) “Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method,” Applied computing and informatics, 12(1), pp. 90–108. doi: 10.1016/j.aci.2014.10.001.

Holý, V., Sokol, O. and Černý, M. (2017) “Clustering retail products based on customer behaviour,” Applied soft computing, 60, pp. 752–762. doi: 10.1016/j.asoc.2017.02.004.

Wei, J. et al. (2017) “Collaborative filtering and deep learning based recommendation system for cold start items,” Expert systems with applications, 69, pp. 29–39. doi: 10.1016/j.eswa.2016.09.040.

Behera, R. K. et al. (2020) “Personalized digital marketing recommender engine,” Journal of retailing and consumer services, 53(101799), p. 101799. doi: 10.1016/j.jretconser.2019.03.026.

Anitha, P. and Patil, M. M. (2022) “RFM model for customer purchase behavior using K-Means algorithm,” Journal of King Saud University - Computer and Information Sciences, 34(5), pp. 1785–1792. doi: 10.1016/j.jksuci.2019.12.011.

Ko, H. et al. (2022) “A survey of recommendation systems: Recommendation models, techniques, and application fields,” Electronics, 11(1), p. 141. doi: 10.3390/electronics11010141.

Smith, B. and Linden, G. (2017) “Two decades of recommender systems at Amazon. com,” Amazon. com. Ieee internet computing, 21(3), pp. 12–18.

Mehta, R. and Rana, K. (2017) “A review on matrix factorization techniques in recommender systems,” in 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA). IEEE.

Wang, S. et al. (2022) “A survey on session-based recommender systems,” ACM computing surveys, 54(7), pp. 1–38. doi: 10.1145/3465401.

Gope, J. and Jain, S. K. (2017) “A survey on solving cold start problem in recommender systems,” in 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE.

Lam, X. N. et al. (2008) “Addressing cold-start problem in recommendation systems,” in Proceedings of the 2nd international conference on Ubiquitous information management and communication. New York, NY, USA: ACM.

Xue, H.-J. et al. (2017) “Deep matrix factorization models for recommender systems,” in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. California: International Joint Conferences on Artificial Intelligence Organization.

Takács, G. et al. (2008) “Investigation of various matrix factorization methods for large recommender systems,” in Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition. New York, NY, USA: ACM.

Koren, Y., Bell, R. and Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer, 42(8), pp.30-37.

Normann and Baumeister, S. Timo Wilm, title = OTTO Recommender Systems Dataset: A real-world e-commerce dataset for session-based recommender systems research.

Wei, K., Huang, J. and Fu, S., 2007, June. A survey of e-commerce recommender systems. In 2007 international conference on service systems and service management (pp. 1-5). IEEE.

Schafer, J.B., Konstan, J.A. and Riedl, J., 2001. E-commerce recommendation applications. Data mining and knowledge discovery, 5, pp.115-153.