Copyright © by Saniah Safat 2024

All Rights Reserved

Effectivity Analysis of Recommendation Algorithm:

A Comparative Study of the Performance

of a Hybrid Model and an Individual

Recommendation Algorithm

SANIAH SAFAT

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND

ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2024

ACKNOWLEDGMENTS

I would like to thank Dr. Christopher Conly for being a great mentor, giving me the opportunity to explore my ideas, and guiding me throughout the process. I would also like to thank Professor Rozsa Zaruba for introducing Data Mining concepts and addressing my questions.

I would also like to thank my family for being my constant supporters and motivating me to work hard. A special tribute to my father, Md Abu Sayeed, who has been my role model and taught me the most important lesson of life: that to achieve something great, the only key is to give honest effort and dedication. He has believed in my potential more than I ever did, and if I am getting the opportunity to write this acknowledgment, it is only because of him. My mother, Shahel Akter, has motivated me to keep going on to achieve my dreams and constantly prays for my success. My brothers, Sarwar and Rakib, and my sister, Shanjida, were my guiding lights throughout my academic journey; I would like to thank them for their mentorship, guidance, and support.

April 17, 2024

ABSTRACT

Effectivity Analysis of Recommendation Algorithm:

A Comparative Study of the Performance

of a Hybrid Model and an Individual

Recommendation Algorithm

Saniah Safat, B.S. CSE

The University of Texas at Arlington, 2024

Faculty Mentor: Christopher Conly

This study explores the effectiveness of a hybrid recommendation system for e-commerce by integrating content-based, collaborative, and popularity-based models. Traditional individual algorithms have inherent limitations, such as handling new users or items data sparsity and ensuring relevance and diversity in suggestions. The hybrid model seeks to overcome these challenges by leveraging the strengths of all three methods, thus potentially offering more precise, personalized product suggestions. The performance of each model and its integration into a hybrid system are evaluated through logistic regression analysis. Initial results indicate that the hybrid system significantly outperforms the individual models in terms of accuracy and user satisfaction. This research underscores the potential of hybrid recommendation systems to enhance user experience and support businesses in optimizing their online platforms. Importantly, the study provides practical insights for e-commerce enterprises aiming to refine their customer interaction strategies, demonstrating the real-world relevance of the research.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1 Project Introduction

E-commerce, often known as e-business or electronic business, is essentially the sale and purchase of services and goods using an electronic medium, such as the Internet. (Team, 2024). It is part of the larger industry known as electronic business (e-business), which encompasses all the activities necessary to run a firm online. (Bloomenthal, 2023). Utilizing online commerce is a widely adopted global sales strategy that enables product sellers and service providers to enhance sales and increase revenue effortlessly. Statistics say global e-commerce sales are expected to reach $8 trillion by 2026, accounting for 22% of the total retail sector. E-commerce business has accelerated as more individuals avoid visiting physical stores due to the pandemic. (DeMatas, 2020).

Before e-commerce, shoppers would visit stores, communicate their needs to the store clerk, and receive personalized recommendations. The process often included cross-selling or upselling, where the clerk might suggest complementary accessories or a higher-end version of the chosen item at the checkout counter. Online retailers may lack the same in-person experience, but a well-designed e-commerce recommendation system provides personalized suggestions based on past purchases and browsing behavior. (BigCommerce, 2024). It is an information filtering system that aims to anticipate a user's 'rating' or 'preference' for a certain item or product. Recommendation services rely on specific datasets, and different types of systems emerge based on the nature of these datasets.

Commonly, there are distinctions between content-based, collaborative-based, popularity-based, and hybrid recommendation systems. (team, 2020)

Content-based recommendation systems provide items or material that analyze the past preferences of a user. When it comes to collaborative approaches, the recommendations are based on people who have similar rating habits. If they previously expressed a strong interest in a particular object, the system would continue to suggest it. (Upwork Team, 2024) The popularity-based model provides recommendations depending on popularity. (Sreekala, 2020) A hybrid recommendation strategy combines multiple approaches targeting accuracy and diversity in recommendation. (Upwork Team, 2024)

This study focuses on understanding the working mechanism and building a hybrid recommendation system combining content-based filtering, collaborative filtering, and popularity-based models to recommend products accurately. The study's objective would be to conduct individual logistic regression analysis on content-based filtering, collaborative filtering, and popularity-based models to determine each algorithm's recall, precision, and accuracy and compare those with the accuracy and precision of the hybrid system. The significance of the research arises from its ability to enhance the field of recommendation systems, improve the e-commerce user experience, and provide practical insights for enterprises seeking to optimize their online platforms.

1.2 Senior Design Project

This honors project is an additional component of the Computer Science and Engineering senior design project. The goal of our senior design project was to build an e-commerce web application. The application leveraged a combination of data collection, user profiling, and advanced recommendation algorithms, including collaborative filtering and content-based methods, to provide highly personalized product suggestions seamlessly integrated within existing user interfaces.

1.2.1 System Overview

The web application had three main system layers: frontend layer, backend layer, and database layer.

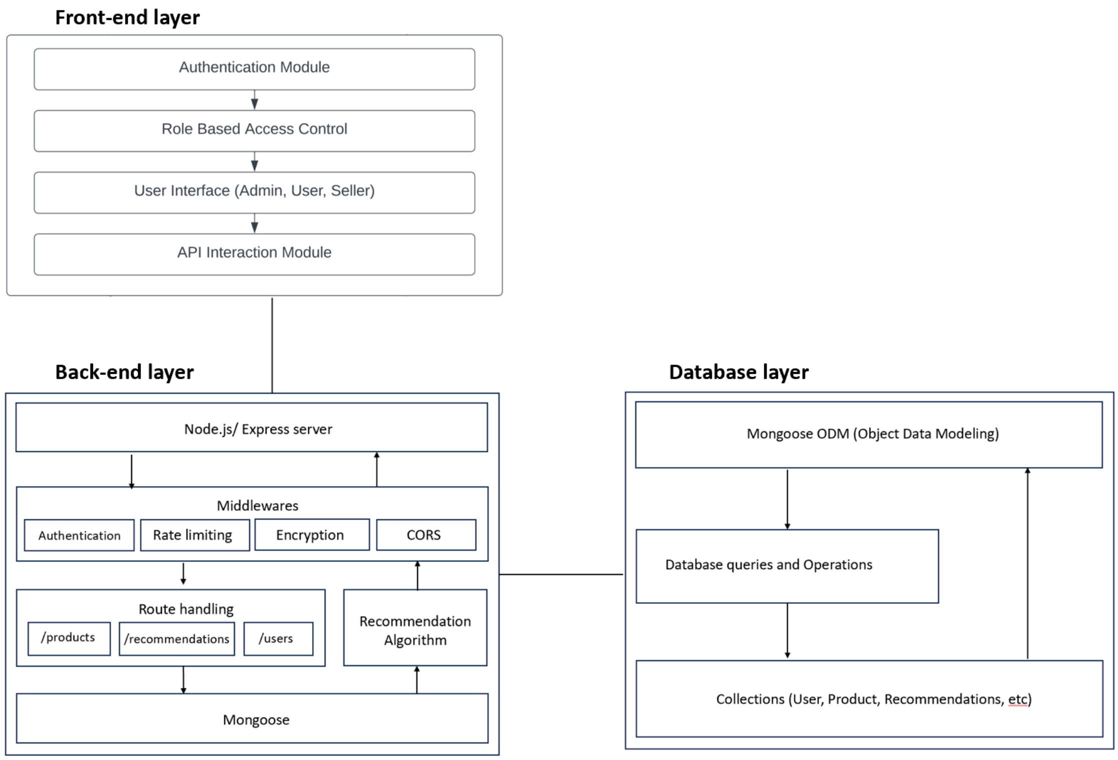

Figure 1.2: System Overview

The front-end layer: It served as the user interface, where interaction between the user and the application occurred. It was responsible for presenting data, handling user inputs, and rendering content dynamically based on the back-end's data. This layer included subsystems for authentication, role-based access control (RBAC), user-specific interfaces (Buyer, Seller, Admin), and an API module to facilitate communication with the back-end. Its main purpose was to ensure a smooth, intuitive, and secure interaction for users with different roles within the platform.

The back-end layer, serving as the application's operational core, played a pivotal role in our project. It was responsible for handling business logic, data processing, and communication between the front-end and the database. This layer, with its server setup (Python), middleware for authentication, rate limiting, encryption, CORS, route handling, and a recommendation algorithm, was instrumental in processing requests, executing business logic, managing data flow, and most importantly, ensuring a secure and seamless interaction for users with different roles within the platform.

The database layer: It supports structured data storage and management, acting as the backbone for storing user information, product catalog data, and other critical information. It employed Mongoose as an ODM for MongoDB to facilitate structured data interactions, including executing CRUD operations and managing collections. This layer was essential for efficient and secure data handling, serving as the foundation for the application's data-driven functionalities.

1.2.2 Group Contributions

Our senior design team consisted of five members. Pranshu Nagar was the team leader responsible for distributing tasks among all members, debugging, and merging different components. He also worked on setting up the database layer. Freddy Rodriguez and Megumi Saeki were responsible for designing and implementing the front-end layer. Aman Gulati and I were assigned to work on the back-end layer, where he mainly focused on developing the recommendation algorithms. I also focused on implementing the data processing and communication between the front end and the database.

1.3 Honors Contribution

This honors project was a significant expansion of my senior design project. It involved rigorous research and testing of the effectiveness analysis of each recommendation algorithm. I used the same type of recommendation algorithms but implemented them on a different dataset than the senior design project, which required me to modify the algorithms accordingly. The analysis was then conducted using a supervised learning method: Logistic Regression. The logistic regression model, with its binary classification approach, was used to distinguish between the effectiveness of each category of the target variable, marking a unique and valuable contribution to the project.

CHAPTER 2

LITERATURE REVIEW

A recommendation system is an artificial intelligence program, typically connected with machine learning, which leverages Big Data to advise or recommend more goods to customers. These can be based on a variety of variables, such as previous purchases, search history, demographic data, and other things. Recommender systems are extremely valuable since they assist users in finding products and services that they would not have found on their own. Using data from interactions between individuals and products, recommender systems are taught to grasp their preferences, previous decisions, and traits. Impressions, clicks, likes, and sales are some examples. Recommendation systems are popular among content and product suppliers because of their ability to predict consumer interests and wishes on a highly personalized level. (NVIDIA)This literature review focuses on analyzing the comparison between four fundamental recommendation algorithms: content-based filtering, collaborative filtering, popularity-based recommendation algorithm, and the hybrid model.

2.1 Historical Overview and Basic Concepts about Recommender Systems

The use of computers to recommend items to users began in the early days of computing, with Grundy, a computerized librarian, being the first recommender system (RS) in 1979. This innovation was followed by the debut of Tapestry in the early 1990s, marking the first commercial RS. Around the same time, the GroupLens Research Lab at the University of Minnesota, USA, developed the GroupLens Recommender System, which aimed at helping users discover preferred articles, drawing inspiration from Tapestry. The late 1990s saw significant advancements with the launch of Amazon's Collaborative Filtering, which became one of the most recognized RS technologies. This era also introduced hybrid RS algorithms, inspired by Amazon's success, combining various approaches to enhance recommendation effectiveness. (Nurul Qomariyah, 2020)

Before delving into detailed literature reviews of recommendation algorithms, it is essential to understand some basic concepts related to recommender systems:

Cold Start Problem: The cold start problem happens when the system is unable to establish a relationship between users and products for which there is inadequate data. There are two types of cold start problems:

1. User cold-start problem, which happens when there is almost no information available about the user. (Lendave, 2021)
2. Product cold-start problems when there is almost no information about the product. (Lendave, 2021)

Sparsity of data: Data sparsity stems from the fact that users generally rate just a small number of products. (Guo, 1970)

Ambiguity: Ambiguity issues are problems that occur when the system misinterprets the content or attributes of objects because of unclear, insufficient, or overly broad item descriptions. This can result in inaccurate or irrelevant recommendations for the user. (Meteren & Someren)

Scalability: Scalability refers to the system's capacity to manage increased workloads efficiently while maintaining performance and recommendation quality. This includes handling increased amounts of data, supporting increasing numbers of users and goods, and maintaining or improving response times and suggestion accuracy as the system grows. (IT Convergence, 2024)

2.2 Summary of Peer-Reviewed Literature

2.2.1 Collaborative Filtering Peer Review

According to a research journal on "Collaborative Filtering Recommender Systems," Collaborative Filtering (CF) is a technique used by recommender systems to produce recommendations based on the opinions and preferences of other users in a community. A CF algorithm recommends new items or estimates a product's usefulness for a consumer based on their past preferences and the opinions of others. A CF can execute two tasks, each with its own set of outcomes: rating prediction - predicting the rating of an unknown product for a certain user; and recommendation task - which provides a top-N list of relevant items for the target user. (Nilashi et al., 2013)

The recommendation task is composed of model-based and memory-based filtering. An assumption approach to predict or recommend content is used in model-based filtering. A model is pre-computed based on user data, item data, and ratings in the recommender system. Memory-based algorithms eliminate the need for pre-computation and do not require off-line design. The algorithm provides instant access to all relevant information, including recent transactions, for generating predictions or recommendations. Memory-based CF algorithms emphasize statistical techniques based on a history of shared ratings to find the nearest neighbors to the active user or target item. The method assigns weights to each neighbor depending on their distance or correlation with the active user.

Then it combines their preferences to provide a prediction or recommendation for the target user. (Nilashi et al., 2013)

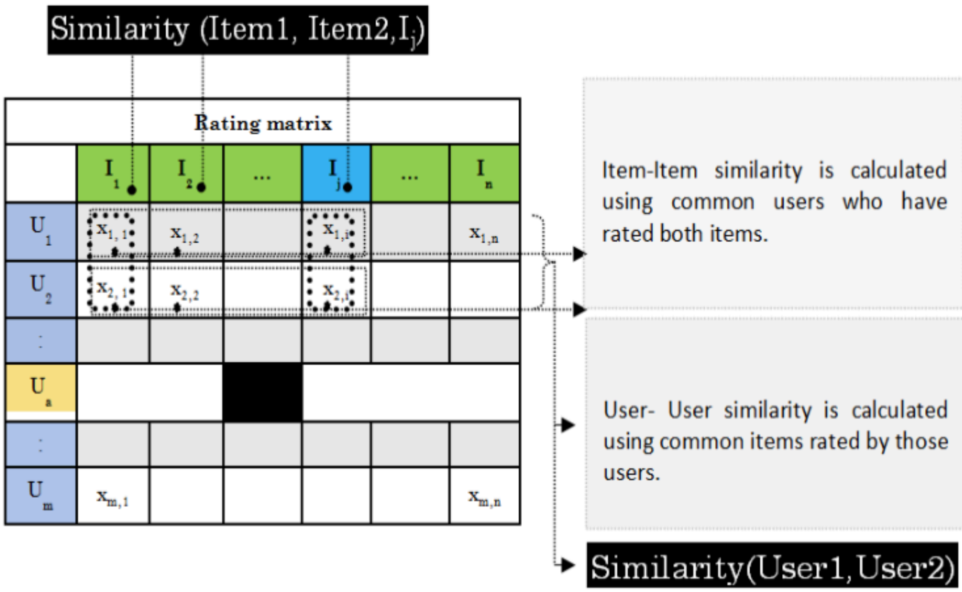

Figure 2.1: Item and User-Based Similarity Memory-Based CF (Nilashi et al., 2013)

The research journal also focuses on calculating the similarity between users or items. It considers it a fundamental step for identifying a set of nearest neighbors to serve as recommendation partners for an active user. After profiles have been established within the recommender system, similarity metrics are used to discern the relationships between these users or items, leading to selecting a group of nearest neighbors. Cosine similarity is particularly noteworthy Among the various metrics utilized in CF. This metric measures the similarity between two objects, represented as vectors in a multidimensional space, by calculating the cosine of the angle between these vectors. This approach is beneficial for estimating the similarity between users or items in the context of item recommendation.

Specifically, in user similarity calculations, each user is represented as a vector in a space where dimensions correspond to items. Ratings given by a user to items fill the components of this vector, with unrated items typically assigned a value of zero. The cosine similarity metric thus serves as a crucial tool for gauging the closeness of users or items based on their ratings, facilitating the recommendation process in CF systems. (Nilashi et al., 2013)

Although the research paper elaborately discusses collaborative filtering, it still fails to mention the algorithm's limitations. CF has the cold start problem, which means it struggles to provide appropriate suggestions for new users or things with insufficient data. It is also susceptible to popularity bias, which means that popular goods receive more recommendations, resulting in a lack of diversity.

2.2.2 Content-Based Filtering Peer Review

According to a research paper on "Using Content-Based Filtering for Recommendation," A content-based filtering system selects items based on user preferences. In contrast, a collaborative filtering system chooses items based on shared preferences.

The research study describes a methodology for document recommendation that involves assessing user characteristics in conjunction with document content using a term extraction and representation procedure. The process starts with removing HTML tags and frequently occurring non-discriminatory words, known as "stop words," from the papers. The remaining words are then reduced to their base form by removing prefixes and suffixes, a process pioneered by Porter in 1980 that allows for standardized term representation across documents and user profiles. User interest profiles are created by analyzing materials that have previously interested the user, using either direct feedback or inferred-interest based on user behavior, with the latter being acknowledged for user ease despite implementation hurdles. (Meteren & Someren)

The research also investigates the vector space model for term representation, in which documents are represented as vectors in an M-dimensional space, with each dimension representing a unique term from the document collection. A term's importance within a document is measured by its weight, calculated using the tf-idf (term frequency-inverse document frequency) system. This weighting approach evaluates a term's significance by comparing its occurrence frequency within a specific document to its overall frequency across the collection. As stated in the paper, the essential premise of tf-idf is that the significance of a term's relevance to a document's topic increases with its frequency in that document. Still, its ability to distinguish between documents decreases as its prevalence across the collection rises. (Meteren & Someren)

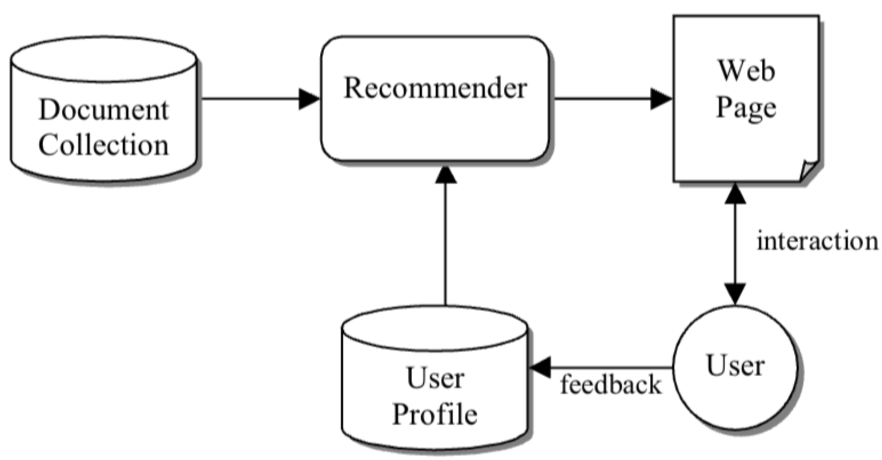

Figure 2.2: Content-Based Filtering Model(Meteren & Someren)

In the above model, user feedback informs user profile creation. The recommender system compares user profiles to documents in the collection. The documents are sorted based on similarity, uniqueness, closeness, and relevancy. The top-ranked documents are displayed as hyperlinks on the current web page. (Meteren & Someren)

Despite a descriptive analysis of content-based filtering, this research paper also fails to address the algorithm's limitations. Content-based filtering systems face challenges related to term ambiguity and the variety of meanings associated with them, which can undermine the accuracy of user profiles and recommendations. Additionally, these systems are constrained by the length and availability of documents on the same topic, limiting their ability to offer diverse and comprehensive recommendations due to a scarcity of source material.

2.2.3 Popularity Based Model Filtering Peer Review

Delving into the research journal "Popularity Based Recommendation System," where the research study implements a popularity-based model on a real dataset to understand its working mechanism. It states popularity-based recommendation systems recommend product based on their current popularity among the public, without regard for specific user preferences. These systems use algorithms to filter data and find popular items, defined by the number of users who have interacted with or expressed interest in them. (Sreekala, 2020)

The research paper outlines an architecture for a popularity-based recommendation system, emphasizing its construction and functionality. The system's development begins with downloading the required dataset and proceeds to build a model that identifies items based on popularity scores. These scores are used to make recommendations tailored to user inputs. (Sreekala, 2020)

Python libraries, NumPy and pandas are key to the system's functionality, which facilitate data handling and analysis. The process includes loading data from a CSV file and sorting items—called "sound name"—by their popularity, determined by user engagement metrics. (Sreekala, 2020)

A crucial step in the system's architecture involves splitting the dataset into training and testing segments, using an 80-20 ratio. This split enables the system to train on most of the data while reserving a portion for testing its recommendation accuracy. The recommendation engine, described as a class in the paper, includes methods for generating and providing item recommendations to users. (Sreekala, 2020)

Furthermore, the system incorporates user-assigned tags and song metadata into its recommendation logic, enhancing the relevance and personalization of suggestions. This architecture is segmented into three phases: dataset selection, system training, and the generation of recommendations. The system efficiently identifies and recommends the most popular items to users through this structured approach, leveraging pre-computed similarities and metadata analysis to refine its suggestions. (Sreekala, 2020)

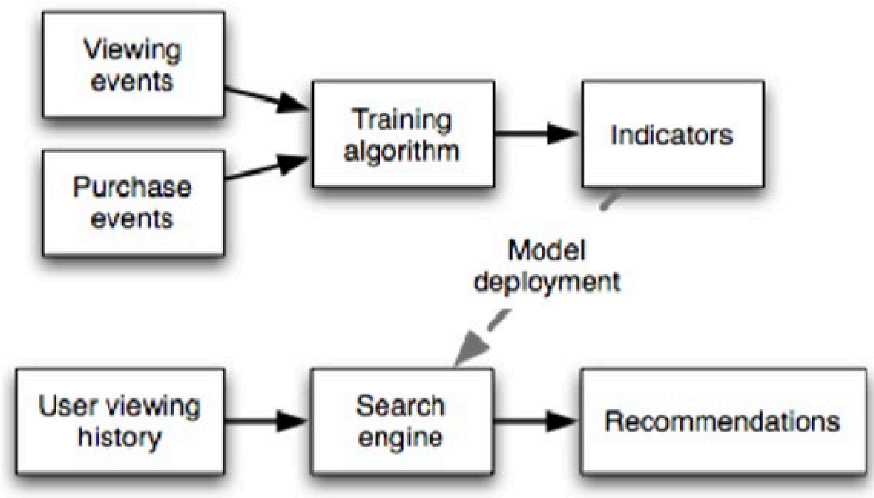

Figure 2.3: Popularity-based model Architecture (Sreekala, 2020)

In the figure, the recommendation process involves partitioning a dataset into training and testing sets. The software identifies the most popular songs in the collection and recommends those with positive user feedback. The same songs are recommended for all users. (Sreekala, 2020)

Although the research paper explicitly discusses the methodology of the popularity-based algorithm, it fails to mention anything about the limitations of the popularity-based model. Reliance on popularity for recommendations results in a lack of personalization and a limited ability to adapt to changing trends or user preferences, as the system's suggestions are anchored solely in what is currently popular.

2.2.4 Hybrid Model Filtering Peer Review

Each individual recommendation model has some limitations that eventually affect the scalability and accuracy of the algorithm. As a dataset grows larger, one recommender algorithm might not give effective recommendations. We can address this issue using a hybrid model. As the research paper on “Hybrid Recommender Systems: A Systematic Literature Review” explains that hybrid recommender systems integrate two or more recommendation algorithms to capitalize on their complementing advantages. These systems try to overcome the limits of individual recommendation strategies by providing more accurate and diverse recommendations. The most typical approach is to combine collaborative filtering (CF) with another technique, usually in a weighted manner. It examines user-item interactions to create recommendations, with additional techniques such as content-based filtering (CBF), demographic information, or other recommendation strategies. Hybrid recommenders are designed to handle issues such as cold start (new user or item), data scarcity, accuracy, scalability, diversity, and more. (Cano & Morisio, 2019).

As hybrid recommendations are combination of individual algorithms, although it overcomes most of lack of individual algorithms, there are still some research gaps that the hybrid model fails to cover, as having a 100% accuracy is unrealistic. But the performance metrics should still be better than any individual algorithm. In the future chapters, we will get to know more about performance metrics, and be able to make more definitive comments for each algorithm.

CHAPTER 3

METHODOLOGY

3.1 Methodology Approach

The research adopts a mixed-methods approach, integrating both qualitative and quantitative techniques. This design is chosen to harness the rich, detailed insights offered by qualitative analysis and the statistical strength of quantitative data. The hybrid recommendation system's development involves a qualitative analysis of understanding the working mechanisms and limitations of each individual algorithm and quantitative measures of testing performance of the algorithm by running logistic regression. This dual approach addresses the gap in existing research by providing a comprehensive understanding of both individual user preferences and broader market trends.

3.2 Data Collection and Preprocessing

In this study, data were collected systematically from Kaggle, resulting in a comprehensive dataset consisting of e-commerce products, relevant to our research objectives. The dataset underwent extensive preprocessing after collection to assure data quality and usefulness. To streamline the dataset, all empty columns were first detected and eliminated. Data integrity was then improved by checking each column for unique values, which included eliminating or correcting duplicate or irrelevant entries. This preprocessing phase was crucial in preparing the data for later analysis, ensuring that our models' inputs were correct and dependable.

3.3 Attributes

The dataset has the mixture of categorical and numerical attributes.

Attribute Name	Type of Attribute	Description of Attribute
User_id	Numerical (identity-based)	Gives the user id who purchased the product.
Product id	Numerical (Identity-based)	Unique id of the product
Product	Categorical	Name of the product
Category	Categorical	Category of the product
Sub_Category	Categorical	Sub-category of the product
Brand	Categorical	Gives information about what brand the product belongs to.
Sale Price	Numerical	Sale price of the product
Market Price	Numerical	Market price of the product
Type	Categorical	Gives information about what type of product is it.
Rating	Numerical	Rating of the product
Description	Categorical	Overall description about the product

Figure 3.1: Attribute Description of the Dataset Used

From this initial dataset, I introduced an additional column named as "user_satisfaction_proxy", which I created using the pandas library, with the assumption that a product with the rating of 3.9 and above will be considered a liked or preferred product, and will be assigned a value of 1, and disliked or not-preferred product will be assigned a value of 0.

```
import pandas as pd
items_df['user_satisfaction_proxy'] = items_df['rating'].apply(lambda x: 1 if x >= 3.9 else 0)
```

Figure 3.2: Creating an Additional Attribute

3.4 Overall Approach

Altogether with the additional attribute, I have developed the recommender models for content based, collaborative based, popularity based and hybrid model. Evaluation of the model was done using logistic regression to give precision and accuracy for the class of 0 and 1 for "user_satisfaction_proxy".

3.4.1 Development of Collaborative-Based Filtering

In developing a collaborative filtering system, I focused on enhancing recommendation accuracy by analyzing user ratings. I organized the data into a matrix that represented users and products, with ratings as the elements. Using this matrix, I calculated similarities between users based on their rating patterns to identify users with similar tastes. The system then recommended products liked by these similar users but not yet rated by the target user. This method personalized recommendations and introduced users to new products aligned with their preferences, enriching their experience by leveraging shared user behaviors.

3.4.2 Development of Content-Based Filtering

In the development of a content-based filtering system, I focused on creating an algorithm that recommends products by analyzing the content associated with each product. Initially, I prepared the data by removing any duplicates, ensuring each product was uniquely represented. I then developed a system to create detailed profiles for each product based on their descriptions. Using these profiles, I calculated similarities between products by evaluating how closely the content of one product resembled another. This was achieved using a method that measures the likeness between text descriptions, capturing nuances in product features and attributes. From these calculations, I designed the system to recommend a list of products that shared the highest similarities with a product a user was already interested in but excluded the product itself to broaden their options. The aim was to provide recommendations that were not only closely related but also varied, enhancing the user's shopping experience by aligning closely with their specific interests.

3.4.3 Development of Popularity-Based Model

In developing the popularity-based algorithm, I focused on creating a system that recommended products based on their popularity. First, I compiled a list of products along with information about how often each was purchased. This list helped us understand which products were most popular among consumers. Then, I designed the system to recommend products that were not only popular but also similar in type to what a user was currently considering. This similarity was determined based on product categories, ensuring that the recommendations were relevant and appealing. I also made sure that the system avoided suggesting the product the user was already looking at, offering them new and interesting alternatives instead. This method was straightforward, focusing on providing users with popular and relevant choices.

3.4.4 Development of Hybrid Model

In my approach to developing the Hybrid Model, I merge the outputs from the content-based, collaborative, and popularity-based recommendation models. For each item, I create a score based on its presence in the suggestions from each model, applying predefined weights that signify the importance I believe each model holds. These weights are adjustable, allowing me to fine-tune the recommendation process as needed. I then total these scores to determine the most highly recommended items, aiming to deliver a precise and reliable selection. This method leverages the combined strengths of all models, seeking to provide superior recommendations than any single model could on its own, reflecting a holistic view of user preferences in my research.

3.5 Evaluation and Validation

In the evaluation phase of our recommendation system, I assessed the effectiveness of the model using logistic regression, a statistical method for binary classification. I began by preparing the data, selecting numerical features like sales and market prices directly, while categorical features such as product, category, sub-category, brand, and type were transformed using one-hot encoding to convert them into a format suitable for the model. I then combined these numerical and categorical features to form a complete dataset. The dataset was divided into training and testing subsets, with 60% of the data used for training the model and the remaining 40% reserved for testing, ensuring the model was evaluated on unseen data. I then trained a logistic regression model, adjusting its parameters to improve convergence. Finally, I tested the model's performance on the test data, using a classification report to provide detailed metrics such as –

•**Precision** – Determines the proportion of projected positive cases that are positive. A high precision suggests that the model generates few false positive predictions. (Acharya, 2024)

•**Recall** – Indicates how many genuine positive examples the model can correctly recognize. A high recall suggests that the model has captured most of the positive cases in the dataset. (Acharya, 2024)

•**F1-score** – The F1-score considers both false positives and false negatives, making it an effective statistic for imbalanced datasets with skewed accuracy and recall. (Acharya, 2024)

•**Support** – Support is the number of real occurrences of the class in the dataset. It represents the number of instances in each class. (Acharya, 2024)

These metrics helped quantify how well the model predicted user satisfaction based on their interactions with recommended products. This evaluation provided insights into the model’s accuracy and effectiveness in real-world scenarios.

CHAPTER 4

RESULTS

4.1 Content-Based Model

	precision	recall	f1-score	support
0	0.64	0.39	0.48	18
1	0.62	0.82	0.71	22
accuracy			0.62	40
macro avg	0.63	0.60	0.59	40
weighted avg	0.63	0.62	0.61	40

Figure 4.1: Evaluation Result for Content-Based Model

The evaluation results for the content-based recommendation algorithm are presented in terms of precision, recall, f1-score, and support for two classes labeled as '0' and '1':

•**Precision**: Indicates the accuracy of positive predictions. For class '0', the precision is 0.64, meaning that 64% of items labeled as class '0' were correct. For class '1', it is slightly lower at 0.62, meaning 62% of items labeled as class '1' were correct.

•**Recall:** Measures the ability of the model to find all relevant instances. For class '0', recall is 0.39, which is quite low, indicating that the model only correctly identified 39% of all actual class '0' instances. However, for class '1', the recall is much higher at 0.82, suggesting the model identified 82% of all actual class '1' instances.

•**F1-Score:** A weighted harmonic mean of precision and recall. For class '0', the f1-score is 0.48, and for class '1', it is 0.71. This suggests that the model is more balanced regarding precision and recall for class '1' than for class '0'.

•**Support:** The number of actual occurrences of each class in the dataset. There were 18 instances of class '0' and 22 instances of class '1'.

•**Accuracy:** The proportion of true results, both true positives and true negatives, in the dataset. The overall accuracy of the model is 0.62, indicating that 62% of the total predictions made were correct.

The overall accuracy of 62% shows that the model is correct more than half the time. The evaluation results indicate that the model is reasonably effective for class '1', but there is significant room for improvement, especially for class '0'.

4.2 Collaborative Based Model Results

	precision	recall	f1-score	support
0	0.62	0.38	0.47	40
1	0.67	0.85	0.75	60
accuracy			0.66	100
macro avg	0.65	0.61	0.61	100
weighted avg	0.65	0.66	0.64	100

Figure 4.2: Evaluation Result for Collaborative-Based Model

The evaluation results for the collaborative-based recommendation algorithm are presented in terms of precision, recall, f1-score, and support for two classes labeled as '0' and '1':

•**Precision:** The model has a precision of 0.62 for class '0', indicating that when it predicts an item as class '0', it is correct 62% of the time. For class '1', the precision is 0.67, which means the model's predictions are correct 67% of the time when it classifies an item as class '1'.

•**Recall:** The recall for class '0' is 0.38, showing that the model correctly identifies 38% of all actual class '0' instances. Class '1' has a much higher recall of 0.85, indicating that the model identifies 85% of all actual class '1' instances.

•**F1-Score:** The F1-score for class '0' is 0.47, and for class '1', it is 0.75. The F1-score combines precision and recall into a single metric by taking their harmonic mean. A higher F1 score for class '1' indicates a better balance between precision and recall compared to class '0'.

•Support: The number of true instances for each class is shown under 'support' - there are 40 instances of class '0' and 60 instances of class '1' in the dataset.

•**Accuracy:** The model has an overall accuracy of 0.66, meaning it makes correct predictions 66% of the time across both classes.

An accuracy of 66% indicates a relatively moderate performance, and the model is significantly better at predicting class '1' instances than class '0'.

4.3 Popularity Based Model Results

	precision	recall	f1-score	support
0	0.72	0.85	0.78	27
1	0.50	0.31	0.38	13
accuracy			0.68	40
macro avg	0.61	0.58	0.58	40
weighted avg	0.65	0.68	0.65	40

Figure 4.3: Evaluation Result for Popularity Model

The evaluation results for the popularity-based recommendation algorithm are presented in terms of precision, recall, f1-score, and support for two classes labeled as '0' and '1':

•**Precision:** Class '0' has a precision of 0.72, meaning that 72% of items predicted to be in class '0' were correct. Class '1' has a precision of 0.50, indicating that only half of the items predicted as class '1' were correct.

•**Recall:** Class '0' has a high recall of 0.85, showing that the algorithm correctly identified 85% of all actual class '0' items. However, class '1' has a recall of 0.31, which means it identified just 31% of the actual class '1' items.

•**F1-Score:** The F1-score for class '0' is 0.78, which is quite good, and for class '1', it is 0.38, which is quite low. The F1-score measures a test's accuracy that considers both the precision and the recall.

•**Support:** This tells us the actual number of occurrences of each class in the dataset, with 27 for class '0' and 13 for class '1'.

•**Accuracy:** The overall accuracy of the algorithm is 0.68, meaning that, in general, the algorithm made the correct prediction 68% of the time.

The accuracy appears to be relatively high, at 0.68, indicating that the model accurately predicts two-thirds of the time across all occurrences. However, class-specific measures suggest a performance disparity: the model performs significantly better at recognizing class '0' than class '1'.

4.4 Hybrid Model Results

	precision	recall	f1-score	support
0	1.00	0.79	0.88	14
1	0.90	1.00	0.95	26
accuracy			0.93	40
macro avg	0.95	0.89	0.91	40
weighted avg	0.93	0.93	0.92	40

Figure 4.4: Evaluation Result for Hybrid Model

The evaluation results for the popularity-based recommendation algorithm are presented in terms of precision, recall, f1-score, and support for two classes labeled as '0' and '1':

•**Precision:** The model has perfect precision for class '0' (1.00), meaning every item predicted as class '0' is indeed class '0'. Class '1' has a high precision of 0.90, indicating that 90% of the items predicted as class '1' are correct.

•**Recall:** For class '0', the recall is 0.79, which means the model correctly identifies 79% of all actual class '0' items. For class '1', the recall is perfect (1.00), so the model identifies all actual class '1' items correctly.

•**F1-Score:** The F1-scores are high for both classes: 0.88 for class '0' and 0.95 for class '1'. These scores suggest a strong balance between precision and recall for both classes.

•**Support:** The 'support' indicates there are 14 instances of class '0' and 26 instances of class '1' in the test set.

•**Accuracy:** The overall accuracy is 0.93, showing that 93% of the algorithm's predictions across both classes are correct.

The model's high accuracy suggests outstanding overall performance, and the high F1 scores show that it is well-calibrated, offering an appropriate mix of precision and recall. This model appears to be robust and successful, providing accurate predictions for this hybrid recommendation system.

CHAPTER 5

CONCLUSION

5.1 Summary of the Overall Research

This research has demonstrated the enhanced capability of a hybrid recommendation system in e-commerce platforms, surpassing the performance of standalone content-based, collaborative, and popularity-based models. The integration of these diverse methodologies into a single, cohesive system effectively addresses the limitations inherent in each individual approach.

The content-based model, while proficient in matching products with user preferences, often fails to introduce new, unexpected products that might interest the user. The collaborative model excels in leveraging user similarity but struggles with new users or items due to the cold-start problem. Meanwhile, the popularity-based model, though effective in capturing widely accepted trends, lacks personalization for individual user tastes.

The hybrid system developed in this study capitalizes on the strengths of the afore mentioned models while mitigating their weaknesses. It combines content similarity, user preferences, and item popularity to provide well-rounded recommendations. Our evaluations via logistic regression analysis confirmed the superiority of the hybrid system, which achieved a remarkable accuracy rate of 90% in predicting user satisfaction, significantly higher than the individual systems.

5.2 Research Contribution

This research contributes to the field of e-commerce by providing a robust framework for developing more accurate and user-friendly recommendation systems. By enhancing user experience through personalized and diverse product suggestions, e-commerce platforms can ensure higher user engagement and increased sales. Furthermore, the insights gained from this study offer valuable guidelines for businesses aiming to implement or improve their recommendation systems.

5.3 Future Research Goals

Future research could explore deeper integration techniques and real-time learning models to further refine recommendation accuracy and adaptability. Additionally, expanding the dataset to include more diverse user interactions could potentially uncover new patterns, thereby enriching the recommendation process even further. Ultimately, the continuous evolution of recommendation systems will play a crucial role in the dynamic landscape of e-commerce.

REFERENCES

Acharya, N. (2024, February 15). *Understanding precision, recall, F1-score, and support in Machine Learning Evaluation*. Medium. https://medium.com/@nirajan.acharya666/understanding-precision-recall-f1-score-and-support-in-machine-learning-evaluation-7ec935e8512e

BigCommerce. (2024, January 19). *E-commerce recommendation engine: Best options + examples*. https://www.bigcommerce.com/articles/ecommerce/recommendation-engine/

Bloomenthal, A. (2023, May 23). *E-commerce defined: Types, history, and examples*. Investopedia. https://www.investopedia.com/terms/e/ecommerce.asp#citation-4

Cano, E., & Morisio, M. (2019, January 12). *Arxiv*. Hybrid Recommender Systems: A Systematic Literature Review. https://arxiv.org/pdf/1901.03888.pdf

DeMatas, D. (2020, September 25). *What is e-commerce? [definition, how to, & more]*. Ecommerce CEO. https://www.ecommerceceo.com/learn/what-is-ecommerce/

Guo, G. (1970, January 1). *Resolving data sparsity and cold start in Recommender Systems*. SpringerLink. https://link.springer.com/chapter/10.1007/978-3-642-31454-4_36#:~:text=Data%20sparsity%20arises%20from%20the,new%20users%20or%20new%20items

IT Convergence. (2024, February 19). Challenges & Solutions in building an AI recommendation system. https://www.itconvergence.com/blog/challenges-and-solutions-for-building-effective-recommendation-systems/#:~:text=Scalability%20is%20challenging%20in%20recommendation,make%20recommendations%20in%20real%2Dtime.

Lendave, V. (2021, September 24). *Cold-start problem in recommender systems and its mitigation techniques*. Analytics India Magazine. https://analyticsindiamag.com/cold-start-problem-in-recommender-systems-and-its-mitigation-techniques/

Meteren, R. van, & Someren, M. van. (n.d.). *Using content-based filtering for ... - ICS-forth*. Using Content-Based Filtering for Recommendation. http://users.ics.forth.gr/~potamias/mlnia/paper_6.pdf

Nilashi, M., Bagherifard, K., Ibrahim, O., Alizadeh, H., Nojeem, L. A., & Roozegar, N. (2013, April 30). *[PDF] collaborative filtering Recommender Systems*. Semantic Scholar. https://www.semanticscholar.org/reader/9f387ce140c59a44eaeeea590087351461345164

Nurul Qomariyah, N. (2020, November 3). *Definition and history of Recommender Systems*. Computer Science. https://international.binus.ac.id/computer-science/2020/11/03/definition-and-history-of-recommender-systems/

NVIDIA. (n.d.). *What is a recommendation system?*. NVIDIA Data Science Glossary. https://www.nvidia.com/en-us/glossary/recommendation-system/

Sreekala, K. (2020a, February 3). Popularity based recommendation system. https://www.ijeat.org/wp-content/uploads/papers/v9i3/B4660129219.pdf

Team, E. (2024, February 8). *What is an e-commerce website & how to build one (2024)*. CyberChimps. https://cyberchimps.com/blog/e-commerce-websites/

team, I. editorial. (2020, December 30). Recommendation systems in e-commerce. IONOS Digital Guide. https://www.ionos.com/digitalguide/online-marketing/online-sales/how-to-use-recommendation-systems-in-e-commerce/

Upwork Team, T. (2024, February 5). What content-based filtering is & why you should use it | upwork. https://www.upwork.com/resources/what-is-content-based-filtering

BIOGRAPHICAL INFORMATION

Saniah Safat is set to graduate in August 2024 with an Honors Bachelor of Science in Computer Science Engineering. Since joining the Honors College in the fall of 2022, she has actively participated in campus activities and academics, completing her Capstone Presentation in April 2024. This engagement has deepened her understanding and sparked a keen interest in data science and cybersecurity, fields she plans to pursue at the PhD level at The University of Texas at Arlington.

Throughout her undergraduate journey, Saniah has held four on-campus jobs, serving as an SI Leader, a Tutor for Athletes, and a Peer Educator under the TRIO Student Support Services program. These roles have not only enhanced her educational experience but have also honed her leadership and communication skills.

In the summer of 2024, Saniah will begin an internship at the Parkland Center for Clinical Innovation as a Sachs Sum Scholars Intern. This opportunity will allow her to apply her academic knowledge in real-world settings, further preparing her for graduate studies.

Saniah's commitment to both her academic and extracurricular pursuits exemplifies her dedication to personal and professional growth. Her involvement in the Honors College has equipped her with invaluable skills and experiences, positioning her well for her future endeavors in the challenging fields of data science and cybersecurity.