Bachelor's Thesis Computing Science

User Intent Prediction for Improved E-Shopping Product Recommendations

Daniel Matias Suryadi Putra
s1078257

December 12, 2024

First Supervisor/assessor: prof. Djoerd Hiemstra
Second assessor: prof. Arjen De Vries
Internship Supervisor: Jolanda Naafs

Radboud University Nijmegen

Abstract

Predicting user intent in e-commerce environments is critical for delivering personalized product recommendations and enhancing the overall online shopping experience. Understanding user behavior accurately enables more relevant search results, increases user engagement, and improves satisfaction. However, predicting user intent is inherently complex due to the unpredictable and diverse nature of user behavior across various contexts and sessions. Factors such as browsing patterns, purchase history, and session length significantly influence user intent. Addressing these challenges, this recommendation system serves as the foundational implementation for BigBridge, employing a hybrid filtering approach to tackle key issues in e-commerce personalization. The system integrates clustering, content-based, and collaborative filtering methods to effectively resolve challenges such as data sparsity and the cold-start problem. By leveraging relevant evaluation metrics like precision, recall, and ranking quality, the system ensures recommendations are precise, actionable, and capable of boosting engagement rates and user satisfaction. In addition to addressing immediate personalization needs, the system establishes a scalable and robust framework for future deployments. This positions BigBridge to achieve sustained growth through advanced user intent prediction and data-driven personalization strategies.

Acknowledgments

Firstly, I would like to extend my sincere gratitude to Jolanda Naafs, Software Architect at BigBridge, whose mentorship and guidance have been invaluable throughout this journey. Her insights into software architecture and recommendation systems have been instrumental in developing BigBridge's first recommendation system, and her support made this experience very rewarding.

I am also deeply grateful to my university supervisor, Dr. Djoerd Hiemstra, Head of the Data Science Department, for his academic guidance and support. His expertise in data science and commitment to research excellence provided a solid foundation for my work.

I would also like to express my appreciation to Maarten Peeters and Bram van der Wal, founders of BigBridge, for providing me with the opportunity to work in an innovative environment and for giving me the opportunity to work on meaningful, impactful projects during my internship. I am thankful not only for their support but also for the warm and collaborative spirit that defines BigBridge.

Lastly, I am incredibly grateful for the opportunity to serve as the project lead in developing BigBridge's first recommendation system. This role allowed me to apply my skills in a real-world setting, gaining invaluable hands-on experience in data-driven product development. Thank you to everyone who supported and encouraged me along this journey.

Contents

Chapter 1

Introduction

1.1 Background

In today's digital age, e-commerce platforms rely heavily on recommendation systems to provide personalized shopping experiences to users. These systems help predict what products users are likely to buy based on their previous interactions, search logs, and purchase history. Understanding user intent is crucial for generating accurate and relevant product recommendations, which is particularly important in environments where user behavior is unpredictable and context-dependent [2].

One of the most successful implementations of recommendation systems can be observed at Amazon, where personalized suggestions based on user behavior and purchase history significantly improve customer satisfaction and drive 35% of total sales.

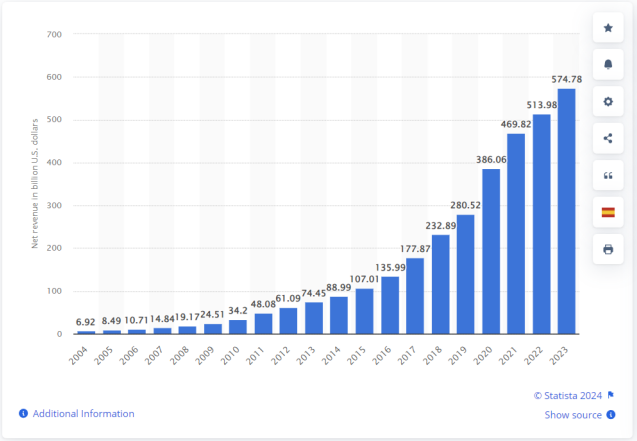

Figure 1.1: Annual net sales revenue of Amazon from 2004 to 2023.

Source: Statista, 2024

Amazon's recommendation engine, using collaborative content and filtering, has set a benchmark in the industry, showing the powerful impact of real-time intent prediction on increasing user engagement and sales conversions [11].

BigBridge, a web development agency where i am conducting my thesis, is developing a subscription-based recommendation system as a product for clients, part of their "SmartPeak" project. This system aims to enhance client sales and customer engagement by implementing hyper-personalization. Drawing from successful strategies used by e-commerce giants, such as Amazon, SmartPeak is designed to deliver relevant product recommendations by analyzing real-time user data, with the goal of increasing user engagement by at least 25%[10].

The system's hyper-personalization goal extends beyond predicting products users may like based on prior purchases; it tailors recommendations to specific user preferences. For example, if a user shows interest in a particular style of white shoes, SmartPeak will suggest similar styles in various colors or brands that align with their taste. Similarly, for a new golfer, SmartPeak could recommend affordable beginner-friendly clubs, while experienced golfers could see high-quality premium options.

Despite the success of such systems, several challenges persist. Existing recommendation techniques, including collaborative filtering and content-based filtering, struggle with sparse and incomplete datasets, the cold-start problem, and the difficulty of balancing precision, recall, and ranking quality in predictions. Moreover, evaluating such systems is inherently challenging due to the lack of standardized ground truths, especially for new users or diverse product categories [1].

Recent developments in recommendation system technology have introduced hybrid approaches that combine different techniques to overcome these challenges. These hybrid systems are more effective at addressing the limitations of traditional methods by improving recommendation diversity and relevance [5]. The goal of this research is to develop a powerful recommendation system that not only improves user engagement but also offers better personalization through advanced machine learning models [7].

1.2 Contributions

This thesis makes the following contributions:

- Development of "SmartPeak," BigBridge's first subscription-based recommendation system designed as a product for clients, combining clustering, content-based, and collaborative filtering to create a personalized, real-time product suggestions (Chapter 5).
- A hybrid model that mitigates cold-start and data sparsity challenges

by balancing product attributes with user interaction data (Chapter 5.3).

- An in-depth empirical evaluation to assess the system's performance and highlight the impact of real-time intent prediction on e-commerce engagement and conversions (Chapter 6).

Chapter 2

Preliminaries

The preliminaries for this thesis cover the fundamental concepts and the methodology for developing a hybrid recommendation system. This system combines two widely used techniques in recommendation systems: collaborative filtering and content-based filtering. Below, we explain these concepts and outline the steps taken to implement and evaluate the hybrid system. These approaches are explored to address common challenges in e-commerce recommendations, such as the cold-start problem and data sparsity.

2.1 Data Collection and Preprocessing

The first step involves collecting and preparing data, which is often a challenging process in the e commerce domain. For this project, real-world e-commerce data sets are used. These data sets include customer behavior data, item metadata (e.g., product categories, descriptions, and images), and user ratings. However, working with e-commerce data is notably difficult due to its incomplete and noisy nature. Datasets in e-commerce often contain missing or ambiguous information, making it essential to perform rigorous data cleaning and preprocessing to ensure the quality and usability of the data to be later used in the recommendation algorithm [20].

2.2 Clustering

Customer segmentation is a key application of data mining that involves grouping customers with similar behavior patterns into distinct groups. This process simplifies the management of a large customer base for businesses.

Clustering is a proven technique for effective customer segmentation. It falls into the category of unsupervised learning, allowing the identification of clusters within unlabeled datasets. Among the various available clustering algorithms, this analysis will focus on implementing the K-means algorithm[24].

2.2.1 Feature Scaling and Dimensionality Reduction

Before proceeding with clustering, it is crucial to scale our features. This step is essential since clustering algorithms like K-means and Dimensionality Reduction techniques such as PCA rely on calculating distances between data points, and having features on different scale might lead to a biased result since large numerical ranges can dominate the distance calculation and distort the analysis which leads to misleading clusters [3].

The Standard Scaler formula used for normalization is as follows [24]:

$$z = \frac{x - \text{mean}(X)}{\text{stdev}(X)} \tag{2.1}$$

where:

- x is an entry in the feature set X
- $\text{mean}(X)$ is the mean of the feature set X
- $\text{stdev}(X)$ is the standard deviation of the feature set X

Dimensionality reduction is a key step when working with high-dimensional data, as it helps simplify the dataset while retaining the essential patterns. Principal Component Analysis is one popular technique (PCA). The original data is converted by PCA into a new collection of variables known as principal components. These variables are uncorrelated and arranged according to how much variance they are able to extract from the data.

The PCA process starts by calculating the dataset's covariance matrix, highlighting relationships between variables. Using eigenvectors and eigenvalues, PCA identifies directions (principal components) with the most variance, with the first few components often capturing the majority. The optimal number of components is usually decided by examining the cumulative explained variance, often using an "elbow" in the variance plot to pinpoint where additional components add little extra value [13].

2.2.2 K-Means Clustering

K-means is the chosen clustering algorithm for this project, grouping data points into K clusters based on their proximity to a central point (centroid). The optimal number of clusters, K, will be determined using the Elbow Method and Silhouette Method.[24].

- **Elbow Method:** Identifies the optimal number of clusters by finding the "elbow point" in a plot of Sum of Squared Errors (SSE). The elbow point indicates when adding more clusters yields minimal improvement in SSE.

- **Silhouette Method:** Evaluates cluster quality by measuring the cohesion within clusters and separation from other clusters. A higher silhouette score suggests well-defined clusters.

While both methods aim to determine the best number of clusters, the Silhouette Method is often preferred when results differ. It provides a more comprehensive evaluation by balancing cohesion and separation, making it effective in cases where the Elbow Method does not exhibit a clear "elbow" or when the clusters are not well-separated. Thus, in situations where the Elbow Method's results are ambiguous, the Silhouette Method offers a more reliable choice for clustering decisions. [22]

2.2.3 Clustering Insights with LLM Analysis

After clustering, Large Language Models (LLMs) like ChatGPT-4 will analyze the characteristics of each customer group. LLMs are highly effective at processing structured data and identifying complex behavior patterns [12]. These models will provide deeper insights into customer habits, spending behaviors, and engagement levels within each cluster.

By using LLMs, the system can refine customer segmentation and offer more personalized recommendations. For example, clusters such as "budget-conscious shoppers" could receive targeted promotions for lower-priced items, while "premium buyers" may be offered exclusive product launches or loyalty programs. This integrated approach enhances personalization and improves recommendation accuracy across different user groups.

2.3 Collaborative-Based Filtering

A collaborative filtering algorithm is implemented to recommend items based on the preferences of similar users or users with similar purchase histories [25].This approach operates under the assumption that users who have shared interests in the past are likely to have similar preferences in the future.

There are two main types of collaborative filtering:

- **User-based collaborative filtering:** Recommends items to a user by identifying other users with similar preferences.
- **Item-based collaborative filtering:** Suggests items that are similar to those the user has already interacted with.

However, Collaborative filtering faces several challenges:

- **Cold start problem:** This occurs when there is insufficient data for new users or new items. Since collaborative filtering relies on user

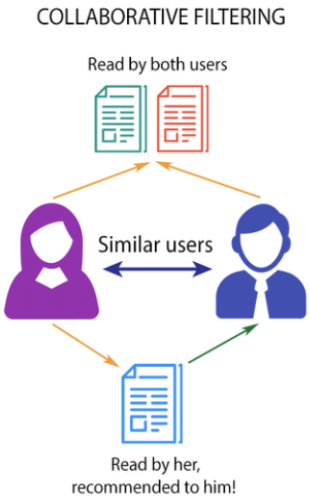

Figure 2.1: Collaborative-Based filtering explained.

Source: https://doi.org/10.69554/AMHI2323

interaction history, it struggles to provide accurate recommendations when a user has no prior interactions or when new items have not yet been rated by many users.

- **Data sparsity:** E-commerce datasets often have large user-item matrices with many missing values, as most users interact with only a small subset of the total available items.

To address the issue of data sparsity, Singular Value Decomposition (SVD) is employed which is a method to perform PCA. SVD reduces the dimensionality of the user-item matrix by uncovering latent factors that influence user preferences, allowing the system to make more accurate recommendations even when interaction data is sparse [9]. By transforming the user-item matrix into a lower-dimensional space, SVD helps mitigate issues such as overfitting and improves the scalability of the collaborative filtering model.

2.4 Content-Based Filtering

In parallel with collaborative filtering, content-based filtering is implemented to recommend items based on the features of the items themselves, rather than relying on user behavior. This method suggests products that are similar to items a user has previously interacted with, leveraging characteristics such as item descriptions, categories, or brand attributes [25].

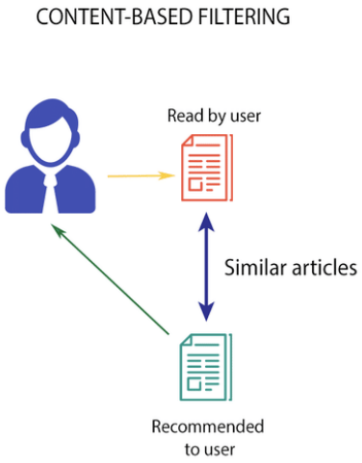

Figure 2.2: Content-based filtering explained.

Source: https://doi.org/10.69554/AMHI2323

Content-based filtering requires detailed information about the items. For each item, features such as keywords, tags, and categories are extracted to build a profile. When recommending new items to a user, the system compares the features of items the user has liked or interacted with to other available items, identifying those with similar characteristics.

To improve the accuracy of keyword matching and feature extraction, Natural Language Processing (NLP) techniques will be employed. These techniques include:

- **Tokenization** and **Stemming/Lemmatization** to break down product descriptions and reviews into meaningful terms.
- **Synonym expansion** to include variations of product names or attributes, ensuring that related terms are captured in the search.
- **Named Entity Recognition (NER)** to identify and prioritize key entities (such as brands, product types, or materials) within product descriptions.

These NLP techniques will enhance the quality of item feature extraction, enabling the system to generate more accurate and relevant content-based recommendations.

To calculate item similarity:

- **Text-based features** (e.g., product descriptions) are vectorized using techniques such as Term Frequency-Inverse Document Frequency

(TF-IDF), which measures the importance of terms within the item descriptions.

- **Cosine Similarity** is then used to compute the similarity score between the vectorized representations of different items, allowing the system to recommend items with similar characteristics [25].

Content-based filtering helps solve the cold start problem faced by collaborative filtering because it relies on item characteristics rather than user interaction history. Even for new users who have no prior interactions or new items with few ratings, recommendations can still be made based on the features of the items.

While content-based filtering handles the cold start problem well and provides personalized recommendations based on item features, it can struggle with offering diverse recommendations. It does not adapt well to changes in user preferences over time as it does not rely on past interactions but only item features

2.5 Hybrid Based Filtering

To address the limitations of both individual methods, a hybrid recommendation system is developed. This approach offers more varied and accurate recommendations by combining content-based filtering with collaborative filtering. The results from both models are combined and weighted to generate the final recommendation list [5].

- For users with sufficient interaction history, collaborative filtering takes precedence by recommending items liked by similar users.
- For new users or items, the system leans on content-based filtering, recommending items based on their characteristics.

. By doing this, the hybrid system ensures that:

- Cold-start problems are reduced, as new users and items can still receive recommendations based on item features.
- Improves personalization by combining content similarity with user-based recommendations.
- Recommendation diversity is improved, by introducing similar products based on user behavior and preferences.

2.6 Result Analysis and Discussion

These models will be evaluated using key offline metrics to measure its performance, following the framework by Shani and Gunawardana [8].

2.6.1 Content-Based Filtering Evaluation

Content-based filtering relies on several key metrics to measure the accuracy and relevance of recommendations:

- **Precision**: The ratio of relevant items recommended to the total number of recommended items, indicating the accuracy of the recommendations.
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$
- **Recall**: The ratio of relevant items recommended to the total number of relevant items available, reflecting the system's ability to retrieve all relevant items.
$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$
- **F1-score**: The harmonic mean of Precision and Recall, providing a balanced measure when there is an uneven distribution between relevant and irrelevant recommendations.
$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$
- **NDCG (Normalized Discounted Cumulative Gain)**: Evaluates the ranking of recommendations by considering both relevance and the order of recommended items. $NDCG$ (Discounted Cumulative Gain) measures the accumulated gain of relevant items at each rank position$i$, while $IDCG$ (Ideal Discounted Cumulative Gain) represents the ideal ranking. p is the number of items.
$$NDCG = \frac{DCG}{IDCG}, \quad DCG = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$
- **MAP (Mean Average Precision)**: Measures the precision across all queries, averaged across users. For each query q in the set of queries Q, MAP calculates the average precision by evaluating both relevance and ranking, providing an overall measure of recommendation accuracy.
$$MAP = \frac{1}{|Q|} \sum_{q \in Q} \text{Average Precision}(q)$$

2.6.2 Collaborative-Based Filtering Evaluation

In collaborative filtering, accuracy is commonly measured by how well predicted user preferences match actual ratings. Key metrics used include:

- **RMSE (Root Mean Square Error)**: Measures the square root of the average squared differences between predicted ($\hat{r}_i$) and actual (r_i) ratings across N items. Lower RMSE values indicate better accuracy.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{r}_i - r_i)^2}$$

- **MAE (Mean Absolute Error)**: Averages the absolute differences between predicted and actual ratings, providing another measure of prediction accuracy.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{r}_i - r_i|$$

2.6.3 Clustering Evaluation

Clustering techniques are assessed using specific metrics to determine the quality of the cluster formations [18]:

- **Silhouette Score**: This metric measures how well a data point fits within its cluster compared to other clusters. A higher score indicates that clusters are well-separated and that points are close to their respective cluster centers.

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

- **Calinski-Harabasz Score**: Also known as the Variance Ratio Criterion, this score measures how well the clusters are separated and compact.

$$CH = \frac{\text{trace}(B_k)/(k-1)}{\text{trace}(W_k)/(n-k)}$$

- **Davies-Bouldin Score**: This metric assesses the average similarity between each cluster and the cluster that is most similar to it.

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

2.7 Related Work

Research in recommendation systems has advanced significantly, particularly in content-based and collaborative filtering techniques. Collaborative filtering, as shown by Sarwar et al. (2001), personalizes recommendations based on user behavior but faces challenges with new users or items due to limited data availability [23]. In contrast, content-based filtering, used

by Lops et al. (2011), recommends items by analyzing attributes like descriptions and tags, making it effective for suggesting new items but less adaptable to changing user preferences [19]. This research combines both methods into a hybrid system, drawing from Koren et al. (2009)'s work on hybrid models. Adapting this approach for e-commerce, the study incorporates real-time interactions and diverse product attributes to create a system that better meets the dynamic needs of online retail [16].

Summary

As a reader, you should now have a clear understanding of the main techniques employed in recommendation systems and how all these methods can be combined into a hybrid model to improve accuracy and relevance. The following chapter will dive into data collection and preprocessing, focusing on preparing the dataset for effective analysis and model building. This step is essential to ensure the recommendation system can deliver reliable, high-quality suggestions.

Chapter 3

Data Collection and Preprocessing

3.1 Dataset Overview

Three datasets are used in this project to develop and evaluate the hybrid recommendation system:

- **Public Beauty E-commerce Dataset**: This publicly available dataset consists of beauty product information, user rating, and sales data. While the dataset provides a rich source of customer behavior and product descriptions, it also suffers from inconsistencies such as missing product descriptions, incomplete user interactions, and noisy data. This dataset will be used for the in-depth analysis, as it provides a variety of real-world challenges that need to be addressed through data cleaning and feature engineering.
- **Movie Dataset**: Unlike the e-commerce datasets, this dataset is relatively complete and well-structured, containing detailed metadata about movies, user ratings, and genres. The completeness of this dataset allows for a smoother implementation of collaborative and content-based filtering methods without the extensive need for preprocessing. This data will only be used on the hybrid based filtering

Each of these datasets poses unique challenges and opportunities for building a robust recommendation system. While the movie dataset is ideal for testing the core recommendation algorithms, the e-commerce datasets, particularly the Public Beauty dataset, introduce real-world challenges such as data sparsity and inconsistencies, which will be addressed in the following sections.

Most of the steps below will be focus on the Public Beauty E-commerce Dataset. The same methodology applies to the other datasets with minor adaptations based on the dataset's specific features and challenges.

ID	ProdID	TimeStamp	Rating	Reviews	Category	Brand	Name	Price
37e+09	2.0	2020-09-24	0.0	0	Premium,Makeup,Nail	OPI	OPI Infinite Shine	8.95
00e+01	76.0	2020-10-30	0.0	0	Beauty,Hair,Color	Nice'n Easy	Nice n Easy Color	29.86
00e+00	8.0	2020-08-06	4.5	29221	Hair,Color,Permanent	Clairol	Clairol Color 7/106A	7.99
00e+00	3.0	2020-07-15	0.0	0	Makeup,Lip	Kokie	Kokie Matte Lipstick	5.16
00e+02	3.0	2020-11-26	0.0	131	Stock,Personal Care	Gillette	Gillette Razor Blades	19.97

Table 3.1: Product Data Table

In the Public Beauty E-commerce Dataset, only a subset of columns is crucial for building the recommendation system. The image highlights the key columns selected.

3.2 Data Cleaning and Preprocessing

Before applying data to a model, data scientists and analysts spend a significant portion of their time organizing, cleaning, and preparing it for analysis. It is often estimated that 80% of their work involves these tasks due to the iterative nature of the process [6]. Data cleaning is essential for building an effective recommendation system, as high-quality data leads to more accurate recommendations and reliable insights [2]. In this section, the Public Beauty E-commerce Dataset undergoes cleaning and preprocessing to resolve issues like missing values, inconsistencies, and noise. The following steps outline the process:

3.2.1 Data Cleaning

Handling Missing Values

E-commerce datasets typically contain alot of missing values due to a wide variety of products and inconsistent data collection across platforms.

- Missing numerical fields such as `Product Rating` and `Product Reviews Count` were filled with 0, as these fields still hold valuable information even when data is not available. A rating of 0 implies no user rating is available.
- Categorical fields like `Product Category`, `Product Brand`, and `Product Description` were filled with empty strings to retain the structure of the dataset while handling the lack of information.

Instead of removing rows with missing values, which would result in significant data loss (especially with only around 5000 rows in total for this dataset), missing values in important fields such as product ratings, reviews count, and product details were filled with appropriate placeholders such as null for numerical fields and empty strings for categorical fields. These

placeholders ensure that the dataset remains intact for analysis without introducing errors or biases from missing data.

Removing Duplicates

Duplicate entries in a dataset can skew analysis results and impact the accuracy of the recommendation system. The dataset was checked for duplicate entries and none were found, indicating that the data is unique in terms of product entries. Keeping the data free from duplicates ensures that recommendations are based on distinct and accurate information.

Outliers

Outliers in the dataset are not removed because they offer valuable insights into the diverse behaviors of customers and product interactions. E-commerce data often captures a wide range of shopping patterns, and these outliers can highlight unique trends, preferences, and niche customer segments, all of which are essential for a robust recommendation system.

Anomaly Detection

Anomalies were examined to maintain data integrity for building a reliable recommendation system. Here's a concise breakdown:

- **Stock Code Analysis**: The dataset comprises 1697 unique stock codes, with the top 10 most frequent codes collectively accounting for a small percentage of the dataset. These frequent codes align with popular, high-demand products, showing expected behavior in e-commerce data.
- **Numeric Character Distribution**: Stock codes typically include 4 to 5 number characters, indicating a consistent and predictable pattern with no inconsistencies. There were no surprising trends that would indicate data errors.
- **Price Analysis**: The `Price` column includes a range of values, with some entries listed as 0. These are assumed to represent valid special cases, such as promotions or free samples, rather than errors.
- **Dataset Diversity**: The dataset covers 1721 unique users and 1697 unique items, confirming a broad and diverse range of products and user interactions, with no irregularities in data consistency.

3.2.2 Data Preprocessing

To enhance the dataset's utility for building a recommendation system, a text preprocessing step was conducted, focusing on extracting and cleaning relevant product information. This preprocessing utilized natural language processing (NLP) techniques with the spaCy library to clean and extract tags from key product attributes.

Tag Extraction and Text Cleaning

The following key attributes were targeted: `Category`, `Brand`, `Description`, and `Tags`. Each of these fields contained valuable textual data that could assist in capturing product similarities for content-based filtering.

The preprocessing involved:

- Converting all text to lowercase for uniformity
- Removing non-alphanumeric characters.
- Filtering out stop words, which are common words (e.g., "the", "and") that do not contribute to the meaning or context of the product
- **Lemmatization**: Converting words to their dictionary forms (e.g, "Showering" to "Shower") to retain grammatical context
- **Stemming**: Cutting down words to their shortest forms by removing endings (e.g, "connections" , "connected" , and "connecting" were shortened to connect) which makes the description easier and faster to compare

The cleaned data was compiled into a simplified string making it more consistent and relevant for better analysis.

3.3 Feature Engineering

Now the next step is Feature Engineering which is a crucial step as it allows the extraction of meaningful insights from raw data. Refining the data through feature engineering can significantly improve the performance and quality of a recommendation system[2]. In this context, most of the feature engineering was based on timestamp data, which provided a wealth of information about customer purchasing habits, frequency of transactions, and spending behavior which are mostly relevant for the clustering part of this project.

3.3.1 RFM

In this project, Recency, Frequency, and Monetary (RFM) analysis was performed to gain insights into customer behavior. RFM analysis is widely used because it effectively segments customers based on their purchasing patterns and is relatively simple to implement while providing significant business value. It is particularly beneficial for predicting future buying behavior and identifying loyal customers [17].Moreover, RFM-based segmentation has been proven to be a robust approach for detecting potential churn, nurturing customer loyalty, and driving personalized marketing strategies[7].

- **Recency (R)**: Measures how recently a customer made a purchase. The `Days Since Last Purchase` feature was calculated as the difference between the current date and the most recent purchase date for each customer.
- **Frequency (F)**: Represents the number of transactions a customer has made. This was captured using `Total Transactions Unique` for distinct transactions and `Total Products Purchased` for the overall quantity of items bought.
- **Monetary (M)**: Reflects the customer's total spending behavior. Features like `Total Spend` and `Average Transaction Value` were created by summing and averaging the spending data linked to each customer's transaction history.

3.3.2 Product Diversity

Understanding the diversity of products purchased by each customer is critical for developing personalized marketing strategies and accurate product recommendations. Research indicates that diverse purchase behaviors can significantly enhance the precision of recommendations by revealing customer preferences across various product categories [2].

- **Unique Products Purchased**: This feature shows how many different things a customer has purchased. A larger number indicates that the buyer has a wide range of interests and experiments with different things. On the other hand, a lesser count can indicate a targeted preference for particular product categories. By identifying these trends in product diversity, we can better segment our consumer base and provide tailored recommendations that suit their individual purchasing preferences.

3.3.3 Behavioral Features

Behavioral features are crucial for capturing the unique shopping habits of customers, which can guide effective marketing strategies. These metrics,

highlighted below, provide a detailed view of how and when customers engage with products [4]:

- **Average Days Between Purchases**: This feature calculates the average number of days a customer waits before making another purchase. Understanding this timing helps us anticipate when a customer may buy next.
- **Favorite Shopping Day**: Identifying the preferred day of the week for shopping helps align promotional efforts with peak customer activity periods, thereby increasing engagement.
- **Favorite Shopping Hour**: Pinpointing the preferred shopping hour provides insights into the times customers are most active which is useful for optimizing the timing for targeted marketing and promotional campaigns.

Gaining insight into these behavioral features in our dataset can help us see our clients more clearly, which will improve the clustering algorithm's performance and produce more insightful customer groups.

3.3.4 Seasonality Trends

Seasonal trends offer an additional dimension in understanding customer behavior. Examining how spending patterns change over time can guide strategic marketing decisions aligned with individual customer preferences. This approach not only aids in product promotion timing but also enhances customer retention strategies[26].

- **Monthly Spending Mean**: This is the average amount a customer spends each month, giving us a sense of their typical spending level. Higher monthly averages may suggest an interest in premium products, while lower averages might reflect budget-conscious choices, allowing us to shape recommendations accordingly.
- **Monthly Spending Variability (Std)**: This measures how much a customer's spending changes from month to month. Higher variability suggests occasional bigger purchases, while lower variability points to steadier spending. This understanding helps us time promotions for months when higher spending is expected.
- **Spending Trend(Slope of Linear Regression)**: To capture the directional pattern of a customer's spending over time, a linear regression model is fitted to each customer's monthly spending data. Specifically, the linear trend component is derived by treating each

month as an independent variable (x-axis) and the monthly expenditure as the dependent variable (y-axis). An upward trend indicates growing interest or satisfaction, while a downward trend may show declining engagement. A stable trend shows consistent spending making us easier to spot customers who may benefit from re-engagement.

With these seasonality and trend insights, we can build precise customer segments and create marketing strategies that align with each customer's spending habits.

Lastly, The `PurchaseCount` feature was integrated into the existing customer data to provide a clearer view of buying behavior. This metric, which sums the total items bought by each customer, offers valuable insights when combined with RFM, Product Diversity, Behavioral Features, and Seasonality Trends, giving a comprehensive perspective on customer preferences.The resulting customer data is shown in the figure below:

ID	PurchaseCount	Days Since Last Purchase	Total Transactions	Total Unique Products	Total Spend	Average Transaction Value
0.0	204	1	172	172	3899.92	22.19047
1.0	119	1	156	156	4195.50	26.891246
2.0	218	0	175	175	4267.75	24.381429
3.0	197	1	155	155	3681.43	23.751016
4.0	105	1	154	154	3540.64	16.648923
5.0	103	1	154	154	3359.56	20.445122
6.0	179	1	187	187	3744.04	20.01925
7.0	224	0	187	187	4240.02	22.673804
8.0	203	1	183	183	3968.30	20.964304
9.0	197	1	160	160	3794.94	22.589029

Table 3.2: Customer Purchase Data

ID	Unique Products Purchased	Avg. Days Between Purchases	Day of Week	Hour	Monthly Spending Mean	Monthly Spending Std	Spending Trend
0.0	110	1.0	2	20	757.786	268.43194	182.175
1.0	100	1.0	2	19	819.010	402.17515	-27.317
2.0	101	1.0	2	22	853.930	348.24743	-278.403
3.0	101	1.0	2	17	918.200	403.84137	78.341
4.0	97	1.0	2	15	671.102	251.18032	-160.202
5.0	112	1.0	2	16	744.004	411.54295	-204.833
6.0	102	1.0	2	14	848.104	414.32542	-203.208
7.0	113	1.0	2	25	721.116	315.27514	157.950
8.0	98	1.0	2	4	754.683	415.89410	192.701

Table 3.3: Customer Purchase Data

3.4 Exploratory Data Analysis

After collecting and organizing the data, assessing its quality, and enhancing its features, the next essential step is exploratory data analysis (EDA). EDA enables us to gain insights through visual exploration, helping us understand customer purchase patterns, spending trends, and product preferences. This phase often reveals key relationships within the data, such as correlations between spending habits, product diversity, and purchase timing. These

preliminary steps are crucial, as they lay a strong foundation for reliable analysis and actionable insights [6].

3.4.1 Item Analysis

Item analysis focuses on understanding product preferences, brand popularity, pricing distributions, and customer feedback. The visualizations presented below provide insights into what drives consumer choices and highlight key product trends within the dataset.

Rating Analysis

Understanding how products are rated can reveal customer satisfaction and influence product recommendations.

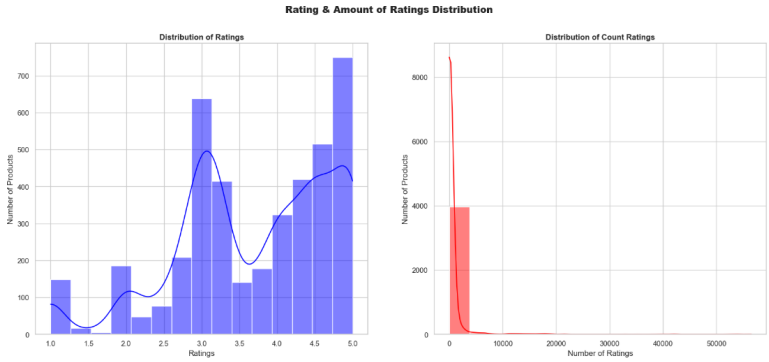

Figure 3.1: Rating and Amount of Rating Distribution

- **Distribution of Ratings**: A visualization of the rating distribution shows a high concentration of products rated around 3 to 5 stars, indicating overall customer satisfaction. The blue line represents the smoothed density of ratings, highlighting the distribution shape, while the red line in the count distribution indicates a threshold or notable point in the number of ratings. Most ratings are positive, suggesting general approval of product quality.

- **Rating Count**: The analysis of the top-rated products highlighted certain brands, such as *Clairol Nice N Easy*, which have received substantial customer feedback. Products with high rating counts are typically more visible and trusted by potential buyers, which could serve as a basis for driving sales.

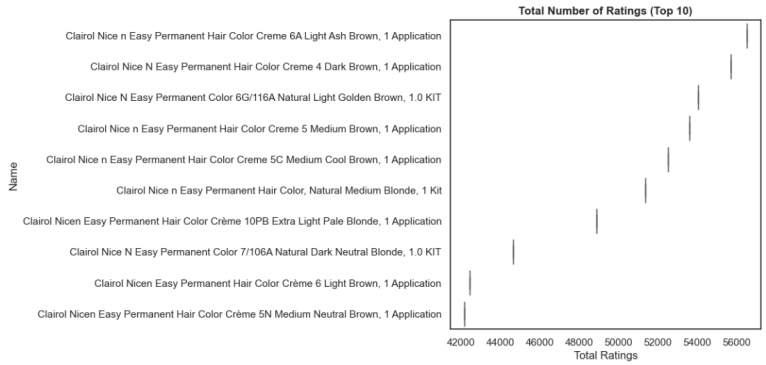

Figure 3.2: Top 10 Most Ratings by Item Name

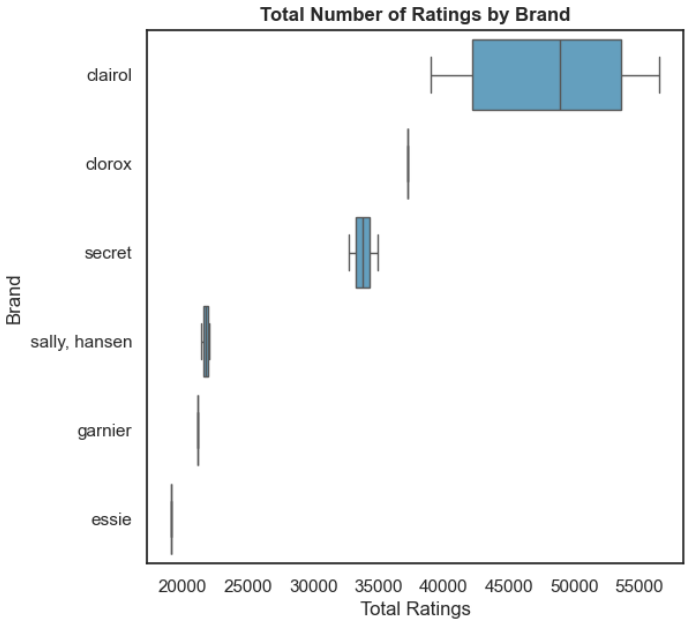

Figure 3.3: Top 10 Most Ratings by Brand

Price Distribution Analysis

The Price Distribution analysis offers insights into the dataset's pricing landscape which is crucial for understanding consumer preferences and market positioning.

- **Top 5 Most Expensive and Least Expensive Products**: The dataset's pricing analysis covers both ends of the spectrum, from budget-friendly to premium items. The least expensive products range from €0.10 to €0.87, including affordable items like low-cost hand sanitizers. On the higher end, the most expensive products range from

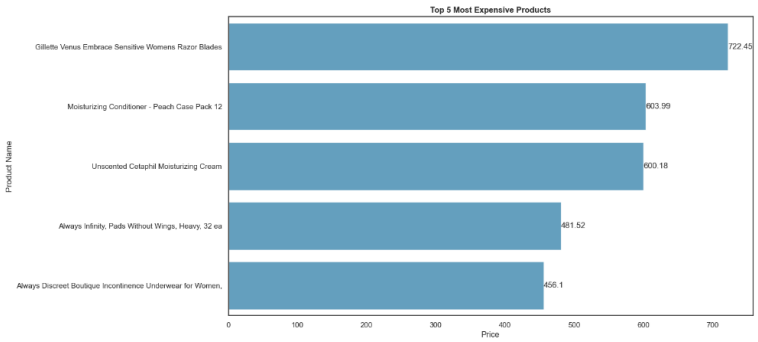

Figure 3.4: Top 5 Most Expensive Products

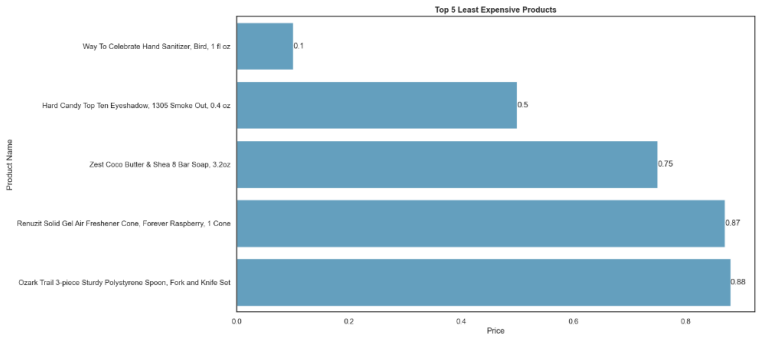

Figure 3.5: Top 5 Least Expensive Products

€456.10 to €722.45, featuring premium products like high-quality razor blades.

- **Price Binning**: A frequency distribution of product prices was created using predefined price bins, helping categorize products based on their price range. These bins include:
  - **Very Cheap**: €0 - €5
  - **Cheap**: €5 - €10
  - **Somewhat Cheap**: €10 - €20
  - **Normal Pricing**: €20 - €50
  - **Moderately Pricey**: €50 - €100
  - **Expensive**: €100 - €200
  - **Very Expensive**: €200 - €400
  - **Premium**: €400 - €800

  This binning reveals that most products fall within the "Cheap" (€5 - €10) to "Somewhat Cheap" (€10 - €20) categories, indicating that

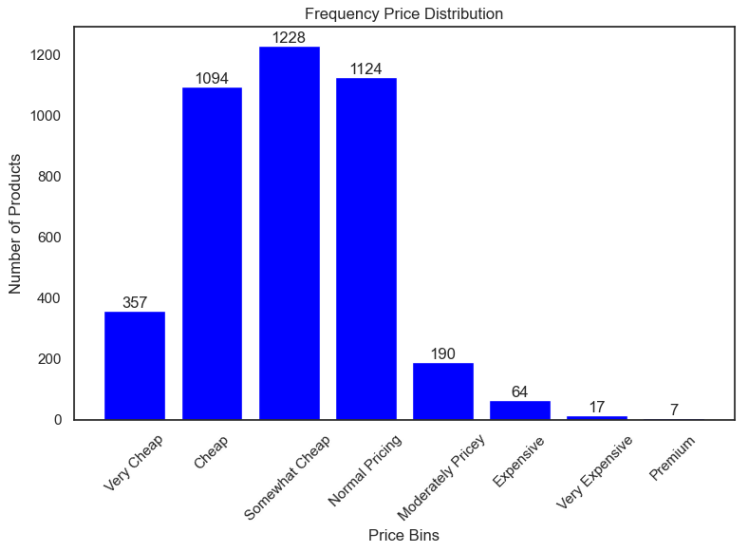

Figure 3.6: Frequency Price Distribution

the dataset largely consists of affordable items. To enhance the dataset for further analysis, the price categories were added back into the item data as a new feature.

Textual Analysis: Word Frequency

In addition to numerical data, textual data from product descriptions, categories, and tags were analyzed using word frequency.

Figure 3.7: Word Frequencies

- **Word Cloud Analysis**: The word cloud generated from product categories, tags, and descriptions offers a quick visual representation of frequently used words. Keywords such as "premium," "beauty,"

"essential," and "walmart" dominate the dataset, indicating a strong focus on beauty and personal care items. This visualization helps in understanding the core focus areas of the product inventory.

3.4.2 User Analysis

This section explores user spending and purchasing habits to uncover trends, segment customers, and predict behaviors. The visualizations below provide key insights into user behavior.

Monthly Spending Trend

The Monthly Spending Trend visualization examines how overall spending varies over time, derived from the `TimeStamp` data in the dataset. This time series analysis utilizes monthly aggregated spending data to observe fluctuations.

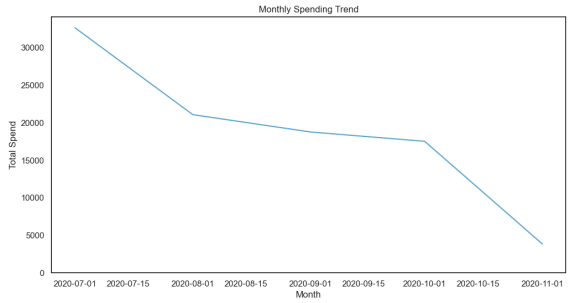

Figure 3.8: Monthly Spending Trend

The graph reveals a clear decline in spending over the analyzed period which drops significantly during October to November. This might indicate off-peak periods or evolving customer interests.

Spender Categorization and Frequency

In the analysis, `Spender Categorization` is derived based on the average price bins of items that each customer purchased. Each product in the dataset is assigned a price bin ranging from "Very Cheap" to "Premium." Using this information, an average price bin score is calculated for each customer.

This score allows for categorizing customers into different spending levels:

- **Very Low Spender**: Score below 2.5
- **Low Spender**: Score between 2.5 and 3.5

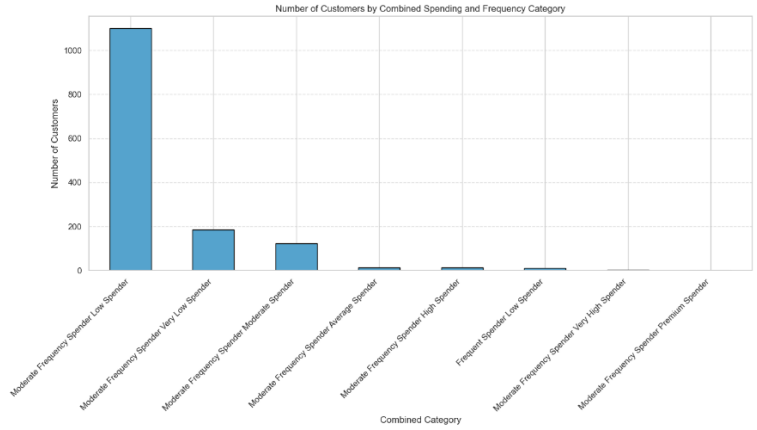

Figure 3.9: Spending and Frequency Category

- **Moderate Spender**: Score between 3.5 and 4.5
- **Average Spender**: Score between 4.5 and 5.5
- **High Spender**: Score between 5.5 and 6.5
- **Very High Spender**: Score between 6.5 and 7.5
- **Premium Spender**: Score above 7.5

The frequency of spending is then analyzed using the `Purchase Count`. Each customer's purchase frequency is categorized into three distinct groups:

- **Infrequent Spender**: Less than or equal to the difference of the average purchase count and one standard deviation.
- **Moderate Frequency Spender**: Between the difference of average purchase count minus and plus one standard deviation.
- **Frequent Spender**: Greater than the sum of the average purchase count and one standard deviation.

With this information, the categorization is then added into the dataset since it will be useful for clustering and identifying customer groups in the next steps.

Correlation Matrix

A quick detection of any connections between the variables is made possible by the Correlation Matrix, which visualizes the links between important elements in the dataset. The key elements that have the biggest effects on comprehending consumer behavior are highlighted in this matrix.

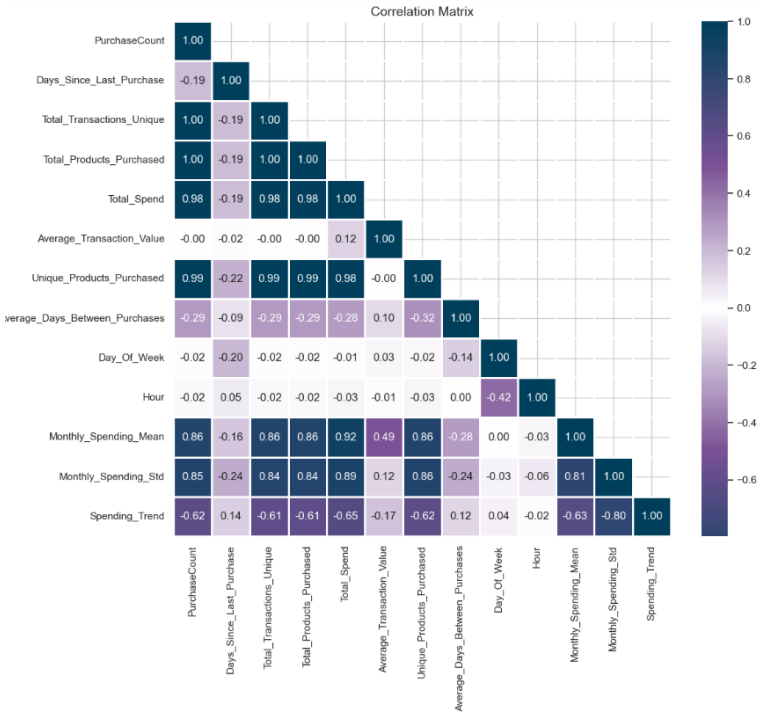

Figure 3.10: Correlation Matrix

- **Strong Positive Correlations**:
  - **Total Products Purchased vs. Total Spend (0.98)**: A near-perfect positive correlation suggests that as customers buy more items, their overall spending increases proportionally. This is expected, as more frequent purchases lead to higher spending totals.
  - **Unique Products Purchased vs. Total Products Purchased (0.99)**: This high correlation indicates that customers who buy a variety of items also tend to purchase a larger quantity overall, emphasizing the diverse buying habits among active shoppers.
  - **Monthly Spending Mean vs. Monthly Spending Std (0.84)**: The strong correlation suggests that customers who tend to spend more on average each month also display greater variability in their monthly spending. This might indicate that higher spenders occasionally make large purchases, influencing spending fluctuations.
- **Moderate Positive Correlations**:

  - **Average Transaction Value vs. Monthly Spending Mean (0.49)**: A moderate positive correlation implies that customers who spend more per transaction often have higher average monthly expenditures.
  - **Purchase Count vs. Unique Products Purchased (0.99)**: A high correlation between these variables suggests that customers making frequent purchases often explore a broader range of products.

- **Weak or Negative Correlations**:
  - **Day of Week vs. Spending Variables**: A weak negative correlation with day-based variables suggests that the specific day of the week has minimal impact on how much customers spend or the volume of purchases they make.
  - **Spending Trend vs. Monthly Spending Std (-0.63)**: The negative correlation between spending trends and monthly variability implies that customers with a consistent upward spending trend tend to have steadier monthly expenses.

The figures from the EDA give a clear overview of the dataset's key patterns. Most products have ratings between 3 and 5 stars, showing strong customer satisfaction. A few products stand out with many reviews, likely due to their popularity. Spending patterns vary, with some customers spending more than others. The data also shows how often customers buy and the variety of products they purchase. The insights gained from these visualizations offer a solid basis for crafting personalized recommendations and shaping marketing strategies that reflect actual consumer preferences and habits.

Chapter 4

Clustering

4.1 Feature Scaling and Dimensionality Reduction

The relevant numerical columns are scaled using StandardScaler, which adjusts the data to have a mean of 0 and a standard deviation of 1. This means the data is centered around 0 with a standard deviation of 1.

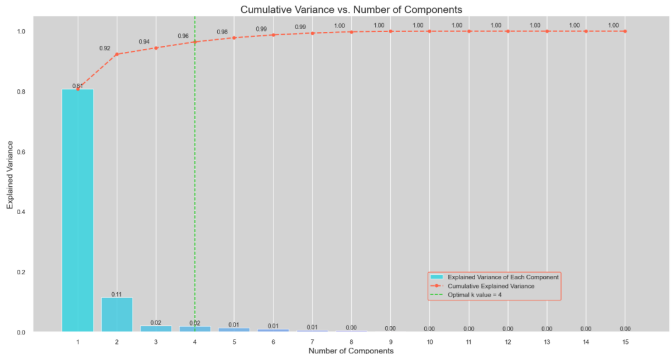

Figure 4.1: Cumulative Variance vs Number of Components

Dimensionality Reduction is applied next with PCA. In this analysis, the threshold is set to 0.95 which means that the goal is to capture at least 95% of the datasets variance. The green dashed line marks the "elbow" point at four components, where the rate of gain in explained variance slows significantly. This suggests that four components are optimal for retaining relevant information while reducing dimensionality.

4.2 Elbow Method

Next, we proceed with analyzing the scaled data by applying the Elbow Method to find the optimal k for the clustering.

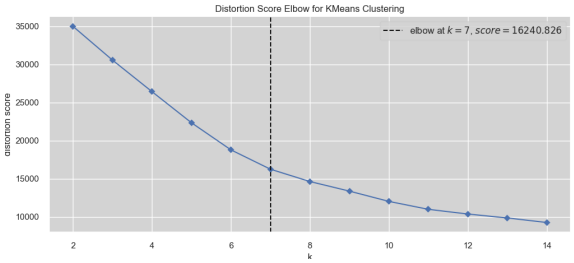

Figure 4.2: Elbow Method

The graph provided reveals that at k=7, the distortion score reaches a point where additional clusters provide limited improvement in reducing within-cluster variance. While the distortion score continues to decline beyond k=7, the reduction is minimal, indicating that seven clusters effectively capture the structure of the dataset without overfitting.

The score value at this point (16,240.826) provides a quantitative measure of the compactness of clusters when k=7. This aids in justifying the choice of clusters, although further validation might be needed, such as comparing with another metric like Silhouette scores, to confirm that the selected k provides meaningful segmentation.

4.3 Silhouette Method

Following the elbow method analysis, the Silhouette method was applied to validate and further refine the clustering process.

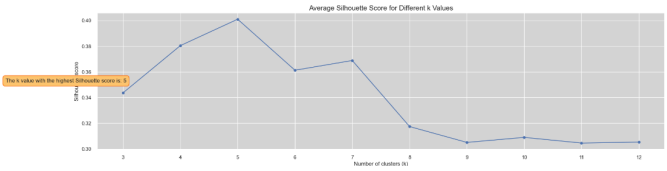

Figure 4.3: Silhouette Method

The plots reveal that the silhouette score peaks at 5 clusters with a score of 0.40, indicating this configuration offers the most distinct separation between clusters. This optimal k=5 suggests that the clustering structure achieves a strong balance between intra-cluster cohesion (points are similar within clusters) and inter-cluster separation (clusters are distinct from each other). A higher number of clusters, such as k=7, as suggested by the Elbow Method, does not provide the same level of clarity in cluster separation, which is why the silhouette score is particularly valuable here and thus will is choosen for over the Elbow Methods result

4.4 Clustering Analysis

In this phase, the K-means clustering algorithm is implemented to categorize customers into distinct segments according to their purchasing patterns and other relevant features. Using the previously identified optimal number of clusters (k=5), each customer is grouped based on behavioral similarities.

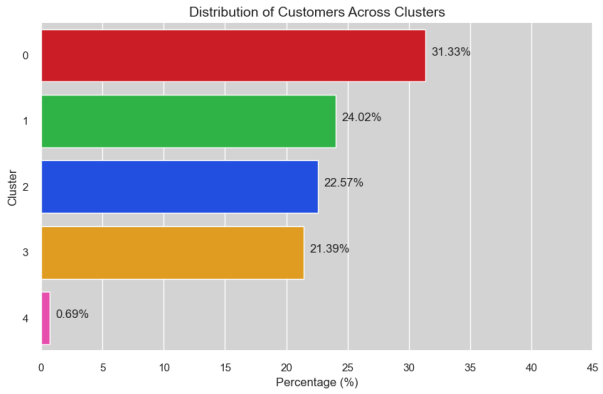

Figure 4.4: Distribution of Customers Across Clusters

The bar plot illustrating the Distribution of Customers Across Clusters shows that the customer base has been segmented into five distinct clusters. Here are the key observations:

- **Cluster 0** is the largest segment, comprising 31.33% of the total customers. This indicates that a significant portion of the customer base shares similar characteristics captured within this cluster.
- **Clusters 1, 2, and 3** have fairly balanced sizes, with **Cluster 1** making up 24.02%, **Cluster 2** at 22.57%, and **Cluster 3** at 21.39%. These clusters represent a substantial part of the customer base and highlight diverse yet significant purchasing patterns.
- **Cluster 4** stands out due to its minimal size, only 0.69% of the customers. This suggests that Cluster 4 may represent an outlier group with unique behaviors not shared by the majority.

The relatively even distribution across the first four clusters implies a well-balanced segmentation strategy, where each cluster reflects distinct customer behaviors without any single cluster overwhelmingly dominating the dataset. Despite its small size, Cluster 4's existence indicates the identification of a niche customer segment which could be particularly valuable for understanding outlier behaviors

4.4.1 Analysis with LLM

The final step of clustering involves analyzing the distinct characteristics of each cluster. As explained from the preliminaries, ChatGPT-4 will analyze

Figure 4.5: Cluster 0-4 Analysis

the characteristics of each customer group based on this figure.

Customer Description and Insights

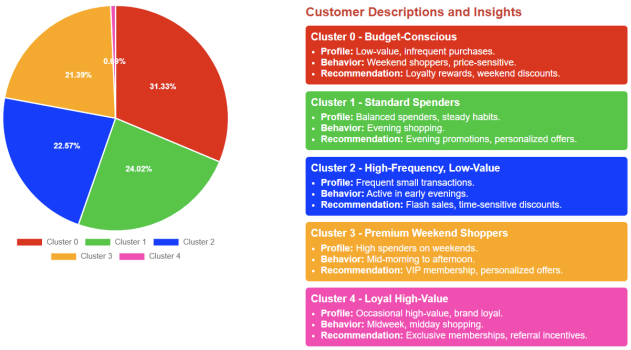

Figure 4.6: Customer Description and Recommendations

Based on the analysis of the five identified customer clusters, it is clear that each group exhibits unique shopping patterns, spending behavior, and engagement levels. By understanding these distinct characteristics, e-commerce platforms can adapt their strategies to meet the specific needs of each cluster, enhancing customer satisfaction, boosting retention, and driving revenue growth. Below are strategic recommendations designed to optimize engagement and sales across the clusters:

Chapter 5

Recommendation System

5.1 Collaborative Based Filtering Implementation

Collaborative-Based Filtering (CBF) was implemented as the first approach to identify user preferences through patterns in similar users' interactions. This approach is widely used in e-commerce, where user-item interaction data often reveals hidden trends that can predict user preferences without explicit feedback.

The model here utilizes user ratings on different products to uncover shared preferences. Each user's preferences are indirectly inferred by comparing them with other users. This enables the system to "learn" which items are relevant. This approach is particularly useful in scenarios where individual users may not have extensive histories but can be grouped with similar users based on their behaviors.

To address challenges commonly associated with collaborative filtering—namely, data sparsity and scalability, a model based on Singular Value Decomposition (SVD) was use since it effectively manages sparse data by capturing latent factors within the user-item matrix.

Data Preparation and Preprocessing

The collaborative filtering model was trained on the purchase history dataset, which includes user-product interactions in the form of ratings. These ratings range from 1 to 5, where higher values indicate stronger preferences. To ensure generalizability, the data was split into an 80% training set and a 20% test set. This split allowed the model to learn from existing data while being validated on its ability to predict user preferences for unseen products.

Model Training with SVD

With the data prepared, the SVD model was trained to identify latent patterns across user ratings. A five-fold cross-validation was performed to eval-

uate the model's robustness. Cross-validation helps mitigate overfitting by training and testing the model on different data subsets, allowing a more reliable assessment of its predictive capability.

In this setup, the SVD model processes user ratings to identify hidden "factors" that influence preferences. For example, one factor might capture a user's inclination toward premium products, while another could reflect a preference for specific product categories.

Hyperparameter Tuning

Hyperparameter tuning was conducted to optimize the performance of the SVD-based collaborative filtering model. The following parameters were fine-tuned:

Parameter	Tested Values	Best Value
`n_factors`	{50, 100, 150}	150
`n_epochs`	{10, 20, 30}	20
`lr_all`	{0.005, 0.01, 0.02}	0.01
`reg_all`	{0.02, 0.05, 0.1}	0.05

Table 5.1: Hyperparameter Tuning Results for SVD Model

Top N Recommendations

After training, the model was set up to provide top-N recommendations for individual users based on their predicted preferences. The collaborative filtering process sorts items by predicted rating which allows us to present each user with their top-N recommended items. This approach also helps mitigate the cold-start problem by leveraging similar user profiles within the latent factor space, even for users with minimal interaction data.

To retrieve recommendations for a specific user, the model ranks all unviewed items by predicted rating scores, highlighting items with high relevance based on user similarity. For example, if User 1 has shown a preference for beauty products, the system can suggest similar items highly rated by users with comparable interests.

5.2 Content Based Filtering Implementation

The next step is to implement Content Based Filtering.This method leverages textual data and product features to identify similar items based on their. `Name`, `Description`, `Tags`, `Categories`, and `Brand`.

In this context, a search system was implemented to help new users or those with little interaction history get relevant recommendations.Unlike

collaborative filtering, which relies on user history, content-based filtering allows us to make relevant recommendations based on product features alone.

Users rarely type full sentences in search queries; they typically input inferred keywords instead. This search behavior guided the design of each search method used in the system. By focusing on individual keywords or phrases, our approach allows for flexibility in recognizing partial matches and related concepts, which enhances the likelihood of connecting users to relevant products based on minimal input.

Exact Match

The Exact Match search serves as a foundation by directly matching user-provided keywords to product names. Given that users often input specific products or well-known terms, Exact Match ensures the highest possible relevance by selecting items that explicitly contain the searched keywords.

AI Vector Search

To capture a broader range of related items, an AI Vector Search using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization with n-grams (unigrams and bigrams) was implemented. This search method analyzes multiple text fields, including `Name`, `Description`, `Tags`, `Brands` and `Categories`, by calculating cosine similarity between vectorized representations. Since users often provide brief, inferred keywords, the AI Vector Search identifies products with similar language and themes, expanding recommendation options even when there isn't an exact match.

Category Match

The Category Match method filters results by focusing on items within the same category as the search query. This approach allows us to interpret keywords at a higher level, linking related products within broader categories. For example, if a user searches for "skincare," the system prioritizes skincare-related items, providing recommendations that align with the general area of interest rather than unrelated products.

Word Search

Word Search enhances flexibility by expanding the search across multiple product fields, such as `Name`, `Description`, `Tags`, `Brand`, and `Categories`. It allows partial matches because it understands that consumers might only type a portion of the product name or a few descriptive words, increasing the possibility of finding relevant goods. This method guarantees that suggestions include a greater range of products related to the term, even if it shows up in other fields.

Fuzzy Search

To handle variations in input, such as typos or slight misspellings, a Fuzzy Search mechanism was integrated. Fuzzy Search tolerates minor differences between the search term and product data by applying a similarity threshold. This method is especially useful because users may input inferred keywords that are close to, but not exactly, the product name. Fuzzy Search captures such near-matches, allowing us to deliver relevant results even when the input is imperfect.

Ranking Adjustments

Finally, all search results undergo a ranking adjustment to prioritize the most relevant items. Each recommended product receives a combined relevance score based on weighted factors such as similarity score from AI Vector Search, exact match confidence, and popularity metrics, including `Rating`, `RatingCount`, and `ReviewCount`. This step ensures that users are presented with the best options at the top of their recommendations.

5.3 Hybrid Based Filtering Implementation

This system ends with hybrid-based filtering. To improve the accuracy of recommendations, it integrates the advantages of collaborative and content-based filtering techniques. This hybrid strategy uses both item attributes and user behavior patterns to increase the recommendations' relevancy.

To better understand the effectiveness of hybrid models, a movie dataset could be utilized in parallel with the e-commerce dataset, as it is often easier to discern similarities in items like movies due to genre classifications and established user preferences. However, the focus of is still solely on implementing the hybrid model within the e-commerce dataset. Movie data, while useful for illustrative purposes, is omitted in favor of applying the model to real-world e-commerce data.

Content Based

Using the previously developed cosine similarity function, the content-based filtering approach identifies products that are similar based on attributes like product name, category, tags, brands, and description. This initial recommendation list provides a set of products that are contextually relevant based on item features. It is important to note that while the content-based filtering model has undergone evaluation, this evaluation focused on the search system aspect rather than the recommendation quality alone

In this hybrid approach, content-based recommendations are used as an initial filtering layer, after which collaborative filtering is applied to rank these items based on personalized user preferences.

Collaborative Based

We employ the collaborative filtering model, previously implemented using Singular Value Decomposition (SVD), to identify patterns in user-item interactions. This model has been trained using optimized parameters to minimize RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error), ensuring better prediction accuracy.

Combination

In the hybrid recommendation system, content-based filtering generates an initial list of similar items, and collaborative filtering then ranks these items based on predicted user ratings. By applying collaborative filtering to the items selected by content similarity, we prioritize products that not only match the user's historical interests but are also highly rated by similar users. This process improves personalization which make it possible for the model to make more accurate recommendations even when explicit user-item interactions are sparse.

Chapter 6

Evaluations

The ideal way to evaluate a recommender and search system is through real-world testing, such as A/B testing or usability studies, to observe how users interact with the recommendations. However, when this is not possible, offline evaluation with pre-defined metrics offers a practical alternative. This chapter focuses on such offline evaluation. For all the evaluations below, the Public Beauty E-commerce Dataset is used.

6.1 Clustering Evaluation

In evaluating the effectiveness of our clustering model, we applied key metrics—Silhouette Score, Calinski-Harabasz Score, and Davies-Bouldin Score—to gauge cluster cohesion and separation.

Metric	Value
Number of Observations	1449
Silhouette Score	0.4597367838037207
Calinski Harabasz Score	2939.7800322495577
Davies Bouldin Score	0.644620911374911

Table 6.1: Clustering Evaluation Metrics

- **Silhouette Score:** A score of approximately 0.46 suggests a moderate level of cohesion within clusters, indicating that while distinct clusters exist, there may be some overlap or similarity between adjacent clusters. Ideally, higher values signify better-defined clusters; however, this score is reasonable given the inherent diversity in consumer behavior patterns. Thus, the model demonstrates an acceptable clustering performance for this dataset.
- **Calinski-Harabasz Score:** The Calinski-Harabasz score is calculated to be 2939.78, reflecting a favorable balance between the between-

cluster and within-cluster dispersion. This high value implies that the clustering model effectively differentiates between clusters and contains variance within each cluster, further validating the model's adequacy.

- **Davies-Bouldin Score:** The Davies-Bouldin score of 0.64 indicates a good level of separation between clusters. Lower values suggest better-defined clusters with minimal overlap, and this score implies that the clusters are relatively distinct.

6.2 Collaborative Based Filtering Evaluation

To evaluate the performance of the Collaborative-Based Filtering model implemented using Singular Value Decomposition (SVD), two widely recognized metrics were used: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). In e-commerce applications, achieving an result close to 1.0 is considered strong performance given the inherent data challenges. For instance, in the Netflix Prize competition, the winning team's algorithm achieved an RMSE of approximately 0.8567, improving upon Netflix's own algorithm, which had an RMSE of about 0.9525. This improvement was significant enough to win the competition, highlighting that an RMSE close to 1.0 is indeed indicative of strong performance in this domain [16].

Fold	RMSE	MAE
1	1.2598	0.9992
2	1.2721	0.9642
3	1.3363	1.0648
4	1.3181	1.0121
5	1.2029	0.8743
Std	**0.0071**	**0.057**
Mean	**1.2778**	**0.9829**

Table 6.2: Cross-Validation Results for SVD Model

The results indicate that the SVD model achieved an average RMSE of 1.2778 and an average MAE of 0.9829 across the five folds. These values suggest that, on average, the predicted ratings deviate from the actual ratings by approximately 1.28 (for RMSE) and 0.98 (for MAE).

6.2.1 Comparison with Existing Studies

While our dataset differs in nature from widely used benchmarks, the characteristics of the datasets are very similar, particularly in terms of sparsity, diversity of user interactions, and the inclusion of temporal features. These shared characteristics make the comparisons valid and relevant.

- A study by Koren et al. [16] evaluated collaborative filtering using SVD and achieved an RMSE of 1.31 and an MAE of 1.04. Their approach incorporated additional features, such as user demographics and temporal patterns, to improve performance. Despite these enhancements, their results are comparable to our RMSE of 1.2778 and MAE of 0.9829.
- Sarwar et al. [23] explored collaborative filtering on e-commerce datasets and achieved an average RMSE of 1.29 using item-based association rule mining techniques. This aligns closely with our findings.

Study	**RMSE**	**MAE**
Our Model	1.2778	0.9829
Koren et al. [16]	1.31	1.04
Sarwar et al. [23]	1.29	N/A

Table 6.3: Comparison of RMSE and MAE Across Studies

These comparisons indicate that our results are on par with or better than those in similar studies. The shared characteristics between our dataset and those in previous studies, such as sparsity and diversity in user interactions, ensure the validity of these comparisons. Our dataset's unique challenges were effectively mitigated by optimizing preprocessing steps and performing hyperparameter tuning for SVD.

6.3 Content Based Filtering Evaluation

The evaluation of the Content-Based Filtering Search System utilized several widely recognized metrics: Precision, Recall, F1-Score, NDCG (Normalized Discounted Cumulative Gain), and MAP (Mean Average Precision). In e-commerce, precision and recall are particularly critical due to the prevalence of missing data and the need to cater to diverse customer needs. NDCG was also included to assess ranking quality, ensuring that the most relevant results are prioritized at the top.

A manual ground truth was constructed due to the inherent challenges in objectively evaluating a search-based recommendation system, particularly for new users. The manual ground truth was developed for 3 specific product categories: Hair Products, Household Products, and Beauty Products where each are assessed using search queries to emulate a realistic user searches within these domains.

In this evaluation, we assume that the chosen search queries represent common or popular terms that users might typically use when searching for these products. These queries reflect common, concise search terms, as users typically rely on specific keywords rather than lengthy descriptions.

6.3.1 Hair Product Evaluation

Search Query	Precision	Recall	F1-Score	MAP	NDCG
Hair Product	0.5582	0.9648	0.7072	0.5471	0.7603
Hair Mask	0.5566	0.9648	0.7059	0.5456	0.5805
Shampoo	0.6575	0.9648	0.7146	0.5561	0.7603
Conditioner	0.5863	0.9497	0.7016	0.5405	0.6454
Combined Average	0.5596	0.9610	0.7073	0.5473	0.6866

Table 6.4: Hair Products Query Evaluation Metrics

The Hair Products category comprises four queries: "Hair Product," "Hair Mask," "Shampoo," and "Conditioner." Across these queries, the recall remains consistently high, averaging around 0.96, indicating that the system effectively retrieves a comprehensive set of relevant items. However, precision varies, with "Shampoo" achieving the highest precision (0.6575), suggesting it yields the most relevant items without excessive noise, whereas "Hair Mask" has slightly lower precision.

For NDCG , "Shampoo" and "Hair Product" rank the highest, achieving 0.7603 and 0.7063, respectively. This indicates that for these terms, the most relevant items are positioned closer to the top of the search results. The F1-score is also relatively stable across all hair-related queries, averaging around 0.71 . Overall, "Shampoo" emerges as the most effective query for hair products, providing a well-ranked and relevant selection.

6.3.2 Household Product Evaluation

Search Query	Precision	Recall	F1-Score	MAP	NDCG
Households	0.5675	0.9609	0.7136	0.5549	0.7254
Detergent	0.6349	0.9348	0.7562	0.6094	0.8256
Cleaning	0.5482	0.9590	0.6953	0.5331	0.7055
Home	0.5804	0.9451	0.7062	0.5549	0.7219
Combined Average	0.5942	0.9476	0.7261	0.5631	0.7446

Table 6.5: Household Products Query Evaluation Metrics

For Household Products, we used queries like "Households," "Detergent," "Cleaning," and "Home." Among these, "Detergent" achieved the highest precision (0.6349). Recall remained high across all household-related queries at approximately 0.95, showing the system's comprehensive retrieval for each term.

In terms of ranking, "Detergent" and "Cleaning" performed best with NDCG scores of 0.8256 and 0.7655, respectively. This indicates that for

these queries, the system was more effective at positioning the most relevant items at the top. While F1-scores averaged around 0.72, "Detergent" demonstrated superior balance between recall and ranking precision, suggesting that specific product terms narrow down results more effectively than broader queries like "Households."

6.3.3 Beauty Products Evaluation

Search Query	Precision	Recall	F1-Score	MAP	NDCG
Beauty	0.4659	0.9648	0.6372	0.4659	0.6313
Makeup	0.4560	0.9545	0.6189	0.4560	0.6568
Lotion	0.4657	0.9455	0.6240	0.4657	0.6977
Lipstick	0.4858	0.9503	0.6451	0.4858	0.6451
Combined Average	0.4682	0.9537	0.6313	0.4682	0.6582

Table 6.6: Beauty Products Query Evaluation Metrics

The Beauty Products category was evaluated with "Beauty," "Makeup," "Lotion," and "Lipstick." Recall remained strong across these queries, averaging around 0.94, which speaks to the system's robust retrieval capabilities in this category. However, precision was lower, with "Makeup" achieving the highest precision at 0.4663, indicating that broader terms like "Beauty" tend to retrieve a larger variety of items, some of which may be less relevant.

In terms of ranking, "Makeup" and "Lipstick" scored highest in NDCG at 0.6651 and 0.6451, respectively, suggesting effective placement of relevant results for these queries. F1-scores for beauty-related terms were consistent, averaging 0.63. Although "Makeup" and "Lipstick" produced more focused results, broader terms like "Beauty" slightly diluted the relevance of retrieved items, highlighting an area for further optimization.

6.3.4 Combined Evaluation

Metric	Precision	Recall	F1-Score	MAP	NDCG
Combined Average	0.5306	0.9466	0.6782	0.5144	0.6507

Table 6.7: Final Combined Evaluation Metrics Across All Categories

When combining results from all categories, the overall performance shows encouraging outcomes for an e-commerce setting. Precision averaged 0.5306, which is good for e-commerce, where missing data is common and achieving over 50% precision indicates reliable specificity. Recall was consistently high at 0.9466, reflecting the system's strong ability to retrieve a broad range of relevant items. With an NDCG of 0.6507, the system ranks relevant items fairly well, ensuring that at least 5 out of the top 10 results

are relevant, which aligns with practical e-commerce needs. The MAP of 0.5144 further suggests effective retrieval, with room to improve ranking for even better customer satisfaction.

6.3.5 Comparison with Existing Studies

While our dataset is proprietary, its characteristics closely align with publicly available datasets used in similar studies, particularly in terms of sparsity, diversity of product attributes, and the inclusion of hierarchical categories. These shared features validate the comparison with existing research.

- A study by Lops et al. [19] evaluated content-based filtering techniques using hierarchical product categories and achieved an average Precision of 0.54 and NDCG of 0.65, comparable to our Precision of 0.5306 and NDCG of 0.6507.
- Another study by Pazzani and Billsus [21] on personalized content-based recommendations achieved a MAP of 0.51 and F1-Score of 0.68, which aligns with our MAP of 0.5144 and F1-Score of 0.6782.

Study	**Precision**	**NDCG**	**MAP**	**F1-Score**
Our Model	0.5306	0.6507	0.5144	0.6782
Lops et al. [19]	0.54	0.65	N/A	N/A
Pazzani and Billsus [21]	N/A	N/A	0.51	0.68

Table 6.8: Comparison of Content-Based Filtering Metrics Across Studies

These results indicate that our model performs competitively with established benchmarks, demonstrating its ability to deliver precise and actionable recommendations.

Chapter 7

Conclusion

This study explores the state-of-the-art methods in user intent prediction and recommendation systems within an e-commerce context, integrating advanced techniques like clustering, content-based filtering, collaborative filtering, and hybrid filtering. Through this multi-method approach, the study addresses key limitations in traditional recommendation models, including cold-start issues, data sparsity, and user intent accuracy.

7.1 State-of-the-Art Knowledge and Awareness

Recognizing the limitations of existing recommendation systems, this study integrates clustering to group users based on purchasing behavior, providing a foundation that enhances both collaborative and hybrid filtering models. By analyzing these clusters with insights from Large Language Models (LLMs), we can better interpret hidden factors influencing user preferences, leading to refined personalization in recommendations.

In addition to clustering, the system leverages content-based filtering using product attributes such as `Name`, `Description`, `Tags`, `Category`, `Brands`. This enables the system to recommend relevant items for new users who may have limited interaction history, addressing the common cold-start problem in recommendation systems.

Furthermore, the model includes a specialized search query system designed to capture typical e-commerce user behavior, where users often provide partial or inferred keywords instead of complete phrases. This search system incorporates techniques such as Exact Match, AI Vector Search, Category Matching, Word Search, and Fuzzy Search to accommodate diverse search patterns, achieving a flexible yet targeted approach for matching user needs even when historical interaction data is sparse. Together, these methods create a comprehensive recommendation and search solution that adapts to user behaviors and preferences effectively.

7.2 Novel Solution and Hybrid Approach

A core advancement in this study is the development of a hybrid recommendation system that combines content-based filtering with collaborative filtering for enhanced relevance. This hybrid model first applies content-based filtering to shortlist relevant items based on product features, capturing inferred user interests from search queries and item attributes. Subsequently, collaborative filtering ranks these items based on predicted user preferences from historical interactions making it achieve a personalized ranking. This two-step process enables recommendations to adapt effectively for both new and returning users by balancing relevance with personalization.

7.3 Results and Practical Implications

The evaluation metrics which can be seen from the results on the evaluation section demonstrate the effectiveness of our hybrid model in providing relevant recommendations for diverse user profiles. The clustering, coupled with LLM insights, successfully manages the diversity in user interests, while content-based filtering enables the system to identify relevant items even for new users. The search query-enhanced content-based filtering model further improves this flexibility, aligning results with inferred user intent based on common e-commerce search behavior.

Comparative analyses of users across e-commerce and movie datasets further validate the model's versatility, showing that the hybrid filtering effectively captures nuanced user preferences regardless of the product domain. This comprehensive recommendation system, by balancing accuracy with adaptability, offers practical implications for e-commerce—showing promise for improved user engagement, satisfaction, and conversion rates.

7.4 Future Work

While this study has made strides in refining e-commerce recommendation systems with methods like clustering, content-based, collaborative, and hybrid filtering, there are several promising paths forward to expand its impact.

Real-World Testing with A/B Testing

While offline evaluation provided useful insights into model performance, real-world testing, such as A/B testing, is essential for a practical assessment. Through A/B testing, user interactions with recommendations in a live setting—such as click-through rates and purchases—could be observed, providing feedback on how accurately the system captures user intent. This iterative feedback loop would allow for continuous refinement and improvement of the recommendation system [15].

Integrating Deep Learning in Collaborative Filtering

Incorporating deep learning techniques, such as transformers (e.g., BERT-based models) or Deep Collaborative Filtering (DCF), into the collaborative filtering component could significantly enhance the model's ability to understand more subtle user preferences and behavioral patterns. These models offer the advantage of fine-tuning instead of full retraining, enabling incremental updates as new data becomes available, thereby adapting to changing user behaviors more efficiently [11].

Prioritizing Inferred Keywords and Product Interaction Data

Enhancing the model's focus on inferred keywords, product views, and other implicit interaction data could improve its relevance and prioritization of recommended products. Observations from BigBridge data show that many user search terms are inferred rather than explicitly stated. By analyzing these inferred keywords, the model could better prioritize relevant results when explicit interaction history is limited. Implementing this feature requires a dataset that includes such interaction data for training and testing [14].

7.5 Closing Thoughts

In conclusion, this study not only confirms the benefits of combining clustering, content-based, and collaborative filtering but also introduces a robust framework for more accurate user intent prediction in recommendation systems. By refining the methodology in personalized recommendation technology, this work establishes a strong foundation for future developments in e-commerce, enhancing digital shopping experiences and fostering effective interactions in online commerce.

Bibliography

[1] Huda Alharthi and Saif Nurmohamed. A survey on challenges in recommendation systems. *Journal of Big Data*, 9(1), 2022.

[2] Sebastian Schelter Barrie Kersbergen. Learnings from a retail recommendation system on billions of interactions at bol.com. *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021.

[3] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping Multidimensional Data: Recent Advances in Clustering*, pages 25–71. Springer, 2006.

[4] Jacopo Tagliabue Ciro Greco Lucas Lacasa Borja Requena, Giovanni Cassani. Shopper intent prediction from clickstream e-commerce data with minimal browsing information. *Scientific Reports*, page 16983, 2020.

[5] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, pages 331–370, 2002.

[6] Gábor Békés and Gábor Kézdi. *Data Analysis for Business, Economics, and Policy*. Cambridge University Press, Cambridge, UK, 2021.

[7] B.A. Ojokoh F.O. Isinkaye, Y.O. Folajimi. Recommendation systems: Principles, methods, and evaluation. *Egyptian Informatics Journal*, pages 261–273, 2015.

[8] Asela Gunawardana Guy Shani. Evaluating recommendation systems. *SpringerLink*, pages 257–297, 2011.

[9] Ali Vardasbi Evangelos Kanoulas James Allan Hamed Bonab, Mohammad Aliannejadi. Cross-market product recommendation. *CIKM '21: Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 110–119, 2021.

[10] Kyung Jin Cha Hyunwoo Hwangbo, Yang Sok Kim. Recommendation system development for fashion retail e-commerce. *Electronic Commerce Research and Applications*, pages 94–101, 2018.

[11] Justin Shamoun Muriel Marable Ying Cui JH (Janghyun) Baek, John Tsai. Amazon recommender system. *University Project*, pages 1–15, 2021.

[12] Ruixiang Tang Xiaotian Han Qizhang Feng Haoming Jiang Shaochen Zhong Bing Yin Xia Hu Jingfeng Yang, Hongye Jin. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 18(6):Article 160, 2024.

[13] Ian T. Jolliffe. *Principal Component Analysis*. Springer, New York, NY, 2nd edition, 2002.

[14] Jure Leskovec Justin Cheng, Caroline Lo. Predicting intent using activity logs: How goal specificity and temporal range affect user behavior. *WWW '17 Companion: Proceedings of the 26th International Conference on World Wide Web Companion*, pages 593–601, 2017.

[15] Ron Kohavi and Roger Longbotham. Online controlled experiments and a/b testing. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 922–929. Springer, 2017.

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[17] Ka Lok Lee, Peter S. Fader, and Bruce G. S. Hardie. Using iso-value curves for customer base analysis. *Journal of Marketing Research*, pages 415–430, 2005.

[18] Gangli Liu. A new index for clustering evaluation based on density estimation. *Tsinghua University, Master Thesis*, 2023.

[19] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook*, pages 73–105, 2011.

[20] Pim Nauts Sebastian Schelter Maarten de Rijke Mariya Hendriksen, Ernst Kuiper. Analyzing and predicting purchase intent in e-commerce: Anonymous vs. identified customers. *SIGIR eCom'20: Proceedings of the 2020 SIGIR Workshop on eCommerce*, pages 1–10, 2020.

[21] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[22] Noviyanti T. M. Sagala and Alexander Agung Santoso Gunawan. Discovering the optimal number of crime cluster using elbow, silhouette,

gap statistics, and nbclust methods. *ComTech: Computer, Mathematics and Engineering Applications*, 13(1):1–10, 2022.

[23] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[24] Darshika Nigam Tanupriya Choudhury, Vivek Kumar. Customer segmentation using k-means clustering. *2018 International Conference on Computational Techniques*, pages 1–7, 2018.

[25] Shivendu Bhushan Tejashri Sharad Phalle. Content based filtering and collaborative filtering: A comparative study. *Journal of Advanced Zoology*, pages 96–100, 2024.

[26] Karthik Subbian Thanh V. Nguyen, Nikhil Rao. Learning robust models for e-commerce product search. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6861–6869, 2020.

Appendix A

Appendix

A.1 Data and Code Availability

All data and code needed to replicate and build upon the findings presented in this thesis are available at:

`https://github.com/d4nielm7/User-Intent-Prediction.git`

A.2 Anecdotal Evidence

This analysis focuses on comparisons between 2 users across different queries to highlight subtle variations in their interests and purchasing patterns

A.2.1 E-commerce Results

	ProdID	Name	Rating	Price	est
3653	7.0	LOreal Paris Excellence Creme Permanent Triple...	4.0	7.91	5.000000
2966	7.0	Loreal Loreal Healthy Look Sublime Mousse Perm...	3.1	15.00	5.000000
3711	35.0	LOreal Superior Preference - 6 Light Brown (Na...	3.3	33.36	4.646874
2786	5.0	LOreal Superior Preference Les Blondissimes, L...	3.0	24.76	3.999205
3950	95.0	Garnier Nutrisse Nourishing Color Creme, Extra...	2.8	34.37	3.681590
59	6418.0	Creme of Nature Colors Hair Color, 1 each	4.0	9.99	3.398189
2666	77.0	Garnier Nutrisse Nourishing Hair Color Creme (...	4.0	6.63	3.251630
1703	61.0	LOreal Excellence Creme - 9-1/2NB Lightest Nat...	5.0	34.24	3.127970
1226	8.0	LOreal Paris Excellence Creme Triple Protectio...	5.0	62.81	3.064031
643	8.0	LOreal Excellence Creme, Medium Brown [5] 1 Ea...	2.9	23.47	3.064031

Figure A.1: User 1: Recommendation results for 'L'Oreal Paris Excellence Creme Triple Protection Color'.

User 1 received a recommendation list focused exclusively on various L'Oreal and Garnier hair color products. This suggests that User 1 has a strong interest in beauty and hair care, particularly in products offering long-lasting color and nourishment. The hidden factors influencing these

recommendations include a preference for high-quality, brand-specific, and nourishing hair colors.

	ProdID	Name	Rating	Price	est
2755	2.0	JUST FOR MEN Color Gel Mustache & Beard M-35 M...	3.2	40.68	4.995725
2	1.0	Garnier Nutrisse Nourishing Hair Color Creme (...	3.9	4.44	4.751803
2811	1.0	Garnier Nutrisse Nourishing Hair Color Creme, ...	4.0	6.97	4.751803
2666	77.0	Garnier Nutrisse Nourishing Hair Color Creme (...	4.0	6.63	4.720700
1903	6.0	LOreal Paris Superior Preference Fade-Defying ...	3.7	8.97	4.088991
2786	5.0	LOreal Superior Preference Les Blondissimes, L...	3.0	24.76	3.996128
731	4.0	LOreal Superior Preference - 9-1/2A Lightest A...	3.2	41.96	3.971195
4077	561.0	LOreal Superior Preference - 8G Golden Blonde ...	4.0	41.96	3.192287
59	6418.0	Creme of Nature Colors Hair Color, 1 each	4.0	9.99	3.156404
3711	35.0	LOreal Superior Preference - 6 Light Brown (Na...	3.3	33.36	3.016505

Figure A.2: User 2: Recommendation results for 'L'Oreal Paris Excellence Creme Triple Protection Color'.

On the other hand, User 2's recommendations, while still centered around hair care products, display a slightly broader range. The list includes products like JUST FOR MEN Mustache and Beard Gel, alongside the expected L'Oreal and Garnier items. This diversity indicates a more general interest in hair care products, without strong brand loyalty. The hidden factor here suggests that User 2 is more open to exploring various brands and product types within the hair care category.

A.2.2 Movie Results

For this analysis, we compare recommendations generated for 2 users as well, focusing on well-known action and superhero films, to examine the differences in hidden factors, such as preference for specific franchises, interest in character-driven narratives, and a tendency toward intense action genres.

Comparison

User 2 received recommendations that heavily feature superhero and action-thriller movies, such as Avengers: Age of Ultron and Batman Begins. This indicates a hidden factor of interest in high-intensity, conflict-driven narratives, and possibly a preference for franchise-based films with iconic characters and dynamic action sequences.

On the other hand, User 300's recommendations, while still containing superhero movies, show a more diverse blend with dramatic action films, including titles Return from Witch Mountains and Hostage. This suggests that User 300 may have a broader interest in both superhero films and action movies rooted in real-world or historical contexts. The hidden factor for User 300, therefore, spans both fictional heroes and more grounded action-drama films, indicating an appreciation for varied genres.

	title	year	id	est
3977	Empire of the Sun	1987	10110	3.685633
2136	Married to the Mob	1988	2321	3.601884
26558	Avengers: Age of Ultron	2015	99861	3.598528
19270	Brake	2012	85414	3.578899
2078	Stage Fright	1950	1978	3.560975
10122	Batman Begins	2005	272	3.529752
8335	Scarface	1932	877	3.515429
4314	The Last Dragon	1985	13938	3.498238
4834	Baran	2001	43774	3.454130
2782	The Dark Half	1993	10349	3.437295

Figure A.3: User 2: Recommendation results for 'Iron Man'.

	title	year	id	est
3977	Empire of the Sun	1987	10110	4.507974
10122	Batman Begins	2005	272	4.222041
9770	Hostage	2005	2026	4.180057
26558	Avengers: Age of Ultron	2015	99861	4.173909
2078	Stage Fright	1950	1978	4.090372
4834	Baran	2001	43774	4.039609
15153	Iron Man 2	2010	10138	4.024074
1982	Return from Witch Mountain	1978	14822	3.998264
2782	The Dark Half	1993	10349	3.997214
8003	To End All Wars	2001	1783	3.997214

Figure A.4: User 300: Recommendation results for 'Iron Man'.

A.3 Customer Funnel Analysis

In addition to developing the recommendation system, I got the chance on using Bigbridge data to conduct a detailed funnel analysis. This analysis tracks each step of the funnel, from the initial session start to purchase completion. The stages are recorded along with the drop-off rates, helping to identify points where users are most likely to abandon their shopping journey.

Key observations from this analysis include:

- **High Drop-Off Points:** The transition from `view_search_results` to `add_to_cart` has a high drop-off rate of 98.31%. This indicates a need for optimization. This could be an area to improve by simplifying

	Stage	Current Stage Count	Next Stage Count	Drop-Off Rate (%)
0	session_start → first_visit	13699	8631	37.00
1	first_visit → page_view	8631	49259	-470.72
2	page_view → view_item_list	49259	23742	51.80
3	view_item_list → view_item	23742	16552	30.28
4	view_item → view_search_results	16552	13473	18.60
5	view_search_results → add_to_cart	13473	228	98.31
6	add_to_cart → view_cart	228	2006	-779.82
7	view_cart → begin_checkout	2006	958	52.24
8	begin_checkout → add_shipping_info	958	1558	-62.63
9	add_shipping_info → add_payment_info	1558	813	47.82
10	add_payment_info → onestepcheckout	813	569	30.01
11	onestepcheckout → purchase	569	568	0.18

Figure A.5: Customer Funnel

the process or offering incentives to encourage users to add items to their cart.

- **Negative Drop-Off Rates:** Some transitions, like `first_visit → page_view` (-470.72%), show negative drop-off rates, indicating possible user backtracking or confusion. This may signal navigation issues that could be resolved by simplifying the layout or enhancing search functions.
- **Friction Points:** A drop-off rate of 47.82% between `add_shipping_info` and `add_payment_info` suggests potential user hesitation or obstacles at this stage. Addressing concerns like unexpected costs or offering diverse payment options could improve conversion rates here.

These points suggest key areas for improvement in BigBridge's user journey.

A.4 System Design

During my internship, I also focused on creating system designs and diagrams to support the recommendation system's workflow. I developed my own designs to show how the recommendation system could fit into BigBridge's current infrastructure and integrate smoothly with their existing components.

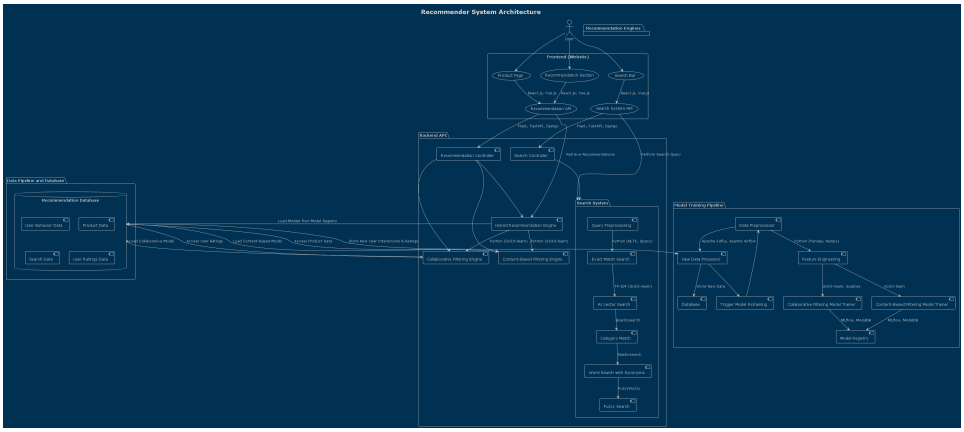

Figure A.6: Recommender System Architecture

A.5 Flow Diagrams

I will include some of the original diagrams provided by BigBridge at the start of my internship, which helped guide my work and showed me where my contribution would fit within their system. I will also include some diagrams created for the implemented model to better understand the process

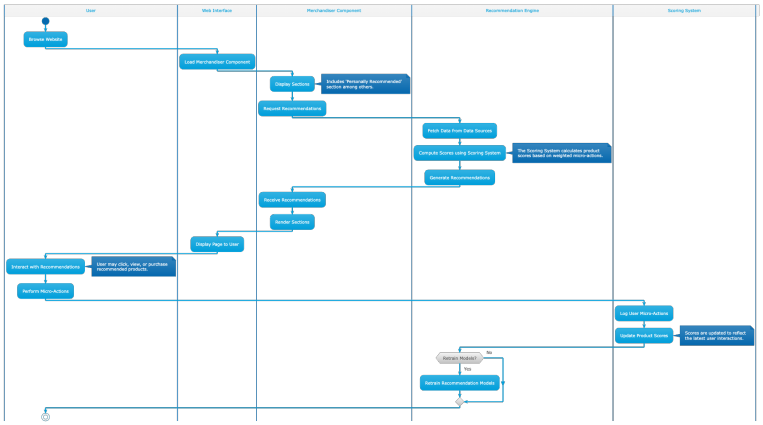

Figure A.7: E-commerce User Journey: Stages from Awareness to Advocacy

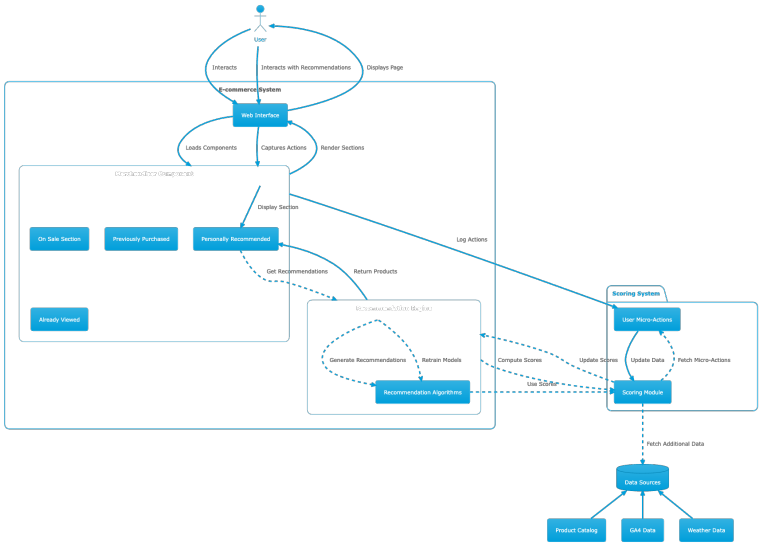

Figure A.8: Recommendation System Architecture: Interaction Flow between Components

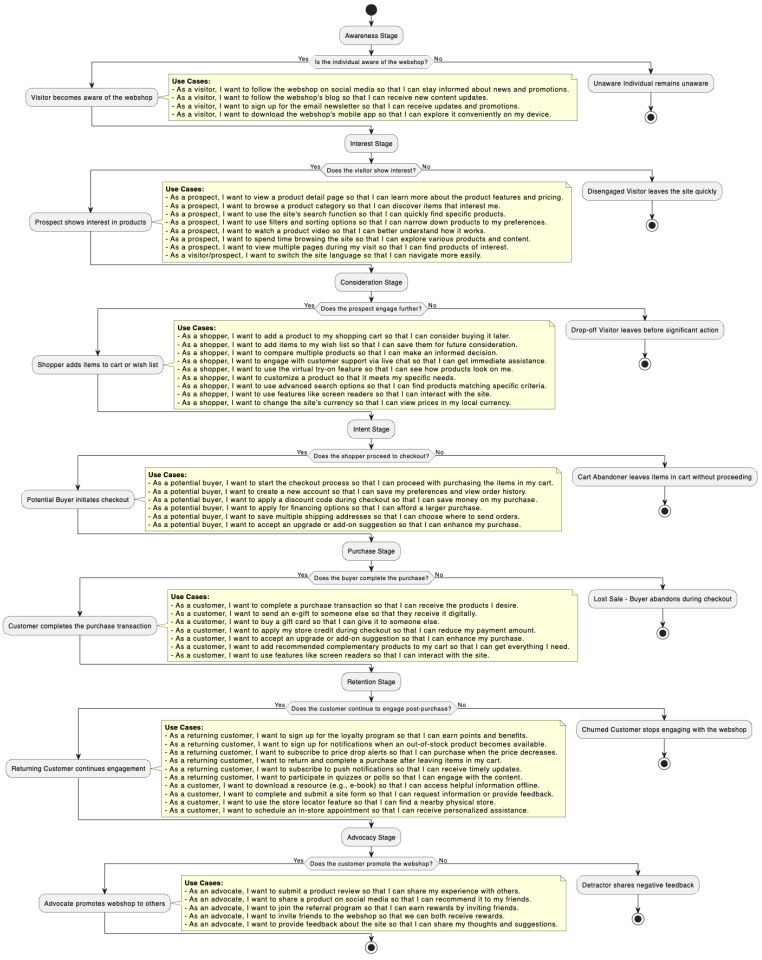

Figure A.9: System Interaction Timeline: Sequence of Events from User Interaction to Model Update

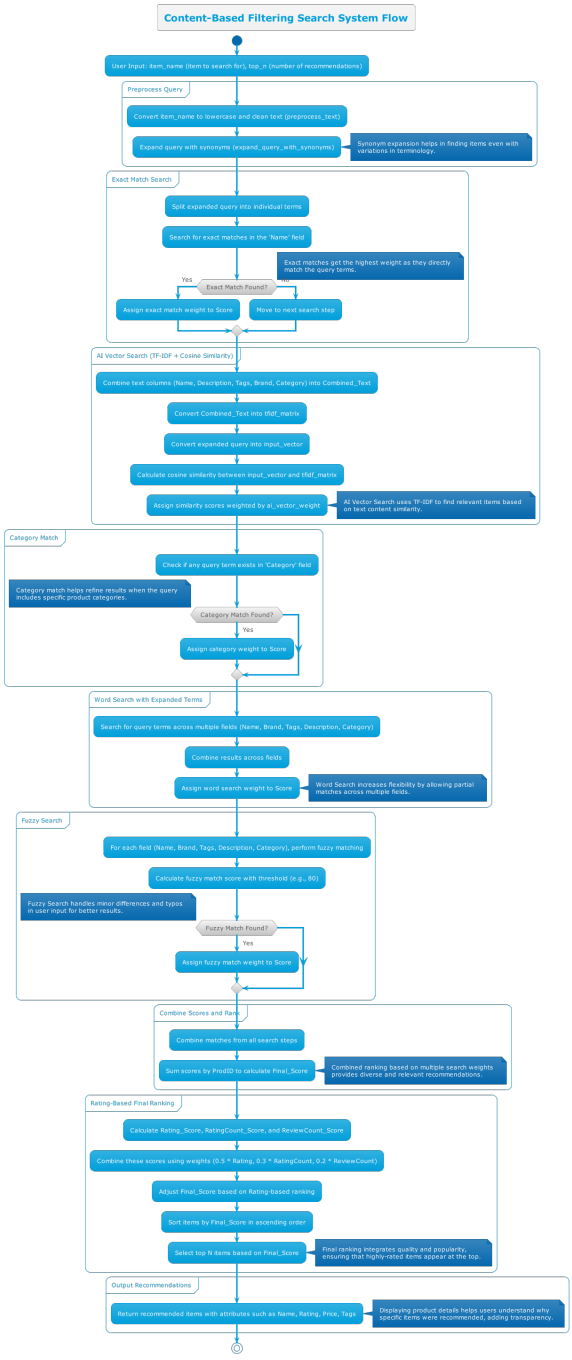

Figure A.10: Content-Based Filtering Search System Flow

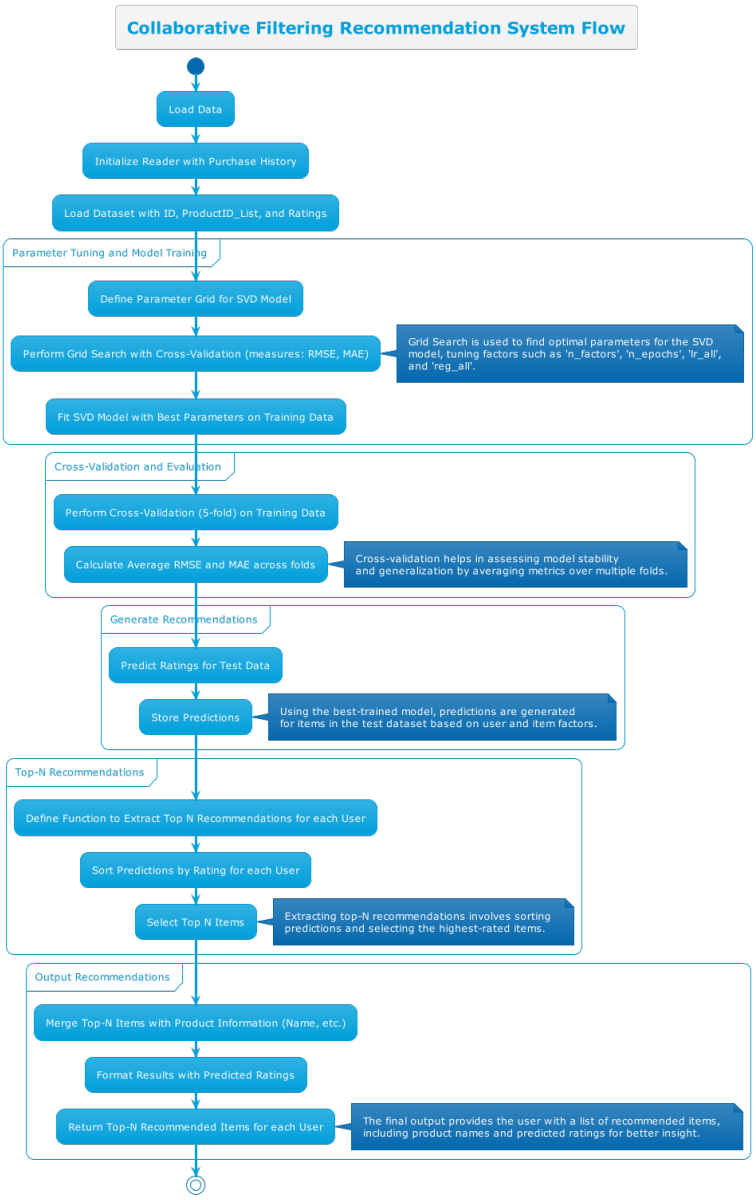

Figure A.11: Collaborative Filtering Recommendation System Flow

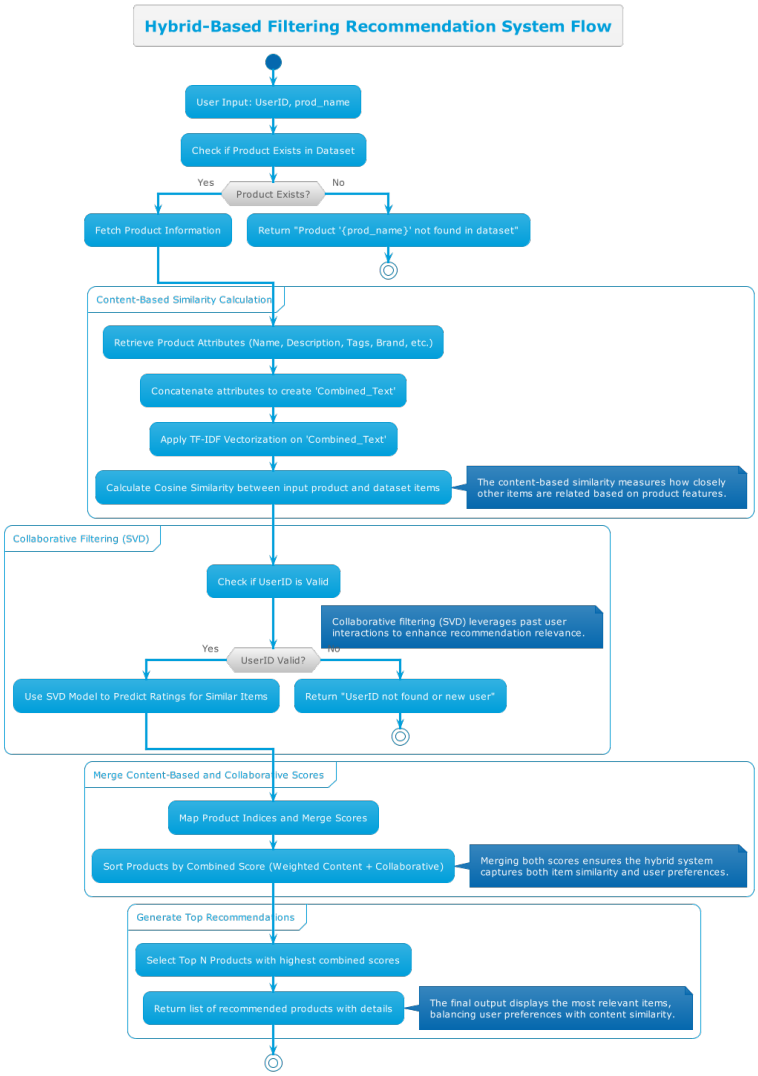

Figure A.12: Hybrid-Based Filtering Recommendation System Flow