PRODUCT RECOMMENDATION SYSTEM

by

Md.Ashiq Ul Islam Sajid
20201225
Raihan Romeo
19301055
Sheikh Farid
20201221
Mohammed Shahrier Tasin
19101255

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
September 2023

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md.Ashiq Ul Islam Sazid
20201225

Raihan Romeo
19301055

Sheikh Farid
20201221

Mohammad Shahrier Tasin
19101255

Approval

The thesis/project titled "PRODUCT RECOMMENDATION SYSTEM" submitted by

1. Md.Ashiq Ul Islam Sazid (20201225)
2. Raihan Romeo(19301055)
3. Sheikh Farid (20201221)
4. Mohammad Shahrier Tasin(19101255)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 21, 2023.

Examining Committee:

Supervisor:

Moin Mostakim
Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Associate Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Recommender systems play a role, in helping consumers by suggesting products. These systems rely on algorithms, many of which are based on machine learning from the intelligence field. However, choosing the algorithm for a recommender system can be pretty challenging due to the range of options available. Additionally developing systems often comes with obstacles. Raises questions.One major challenge in creating a recommendation system for an e-commerce business is the "cold start" problem. This occurs when there isn't data in the product dataset. Consequently accurately recommending products to customers becomes extremely difficult. The problem arises because there is no user purchase history or item ratings, for users making it impossible for the system to provide recommendations based on their preferences.In our research, we focus on addressing this cold start problem in recommender systems. Our goal is to find solutions using multiple approaches, including similarity methods and clustering algorithms like Agglomerative Hierarchical and K-means clustering, and also create a merged recommendation system from all the approaches. By doing we aim to overcome the cold start issue and assist businesses in generating accurate recommendations.To conduct our research we use an unsupervised learning dataset since the cold start problem primarily occurs when there is no user data available. Our investigation aims to provide insights and suggestions to address the challenges related to the cold start problem, in recommender systems. This will benefit businesses. Improve the user experience.

Keywords: Machine Learning, Product Recommendation, K-means, Clustering, Jaccard similarity, Agglomerative clustering, cosine similarity

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis has been completed without any significant interruption.
Secondly, to our advisor Mr. Moin Mostakim sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their support, it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ICD The inter-cluster dissimilarity

AS Agglomerative clustering

CS Cosine similarity

DP Data Preprocessing

EM Evaluation Metrics

ICS The intra-cluster similarity

JS Jaccard similarity

MLT Machine learning techniques

PRS Product recommender system

Chapter 1

Introduction

1.1 Thoughts Behind The Product Recommendation Engine

The ever-increasing amount of data worldwide makes locating relevant info in this data-driven society tough. A recommender system can assist the user in locating relevant and tailored information or their preferred product. Existence of numerous types of recommender systems. Collaborative filtering, content-based filtering, and hybrid recommender systems are common recommender systems. In the real world, Facebook, Google, and YouTube, Amazon use recommendation systems to recommend specific types of information. Different applications implement recommender systems using distinct models. Creating a recommendation system face a different type of obstacles. such as cold start, shilling attack, latency, sparsity, and grey sheep. The application of the Recommender system is seen in E-commerce. E-commerce websites utilize this Recommender system to determine consumer preferences. Big E-commerce uses collaborative filtering or hybrid filtering recommendation systems on their website. They can use this recommendation system because they already have a big database where they can use the rating system and user purchase history or click system and based on this they can build a recommendation system but how can a new E-commerce site recommend a product if there is no user data or product rating data or any product purchase data. For this, we can use a content-based recommendation system. Based on the product description we can recommend other product that has a similar type of description or a similar type of category product they might use. This allows the recommendation engine to predict user preferences from the beginning of the E-commerce site without the user preference. This way we can improve the cold start problem. Cold start is a problem in a recommendation system where there is not yet optimal data available for the engine to recommend the best possible recommendation. For this issue, we tried to solve this problem. At first, we collect product datasets from the Kaggle website where we collect the BigBusket dataset. It is an unsupervised dataset. So we have to use machine learning clustering algorithms. For the clustering algorithm, we used the K-means clustering algorithm and Hierarchical agglomerative clustering. But first, we have to convert the product description text data into numerical representations. Then we use the machine learning algorithm to train and predict the data. Expanding the use of recommender systems in product-based applications. User happiness can play a key part in a product-based business's success. A product-based recommender system's

primary the objective is to recommend a product that the user or consumer desires effectively. With the help of machine learning, our recommendation engine can develop effective suggestion methods in this manner. For some product recommendations, we need product data, and by training that dataset, we can create a recommendation engine that can present the user with helpful suggestions.

1.2 Motivation

There are a variety of challenges involved in making effective use of this system. And one of the goals of this study is to make things easier or get rid of the obstacles for initial businesses. Developing a powerful product recommendation engine needs both time and financial investment. You will have to pay money for data scientists and other professionals, as well as for discovery and analysis, which may include a feasibility study, in order to ensure that this is the best path for your company to take. The key aim of this research is, therefore, to come up with a solution that is consistent with these types of problems and is also practicable.

1.3 Problem Statement

The wholesale and retail sectors are both undergoing rapid change. Many brick-and-mortar stores are going out of business, and they are being quickly replaced by online retailers, brands that sell directly to consumers, and subscription and membership services. Customers go to websites because of the wide range of products available, but many E-Commerce platforms don't make enough sales to clear out a significant portion of their stock. This is frequently attributable to an unsatisfactory browsing experience for the user. Customers might spend hours looking through hundreds or even thousands of different products before finding something they want to buy, but they might not find anything they like. It is necessary to make suggestions to customers based on their preferences and requirements in order to improve the overall shopping experience and increase the amount of time spent on a website. Although it is obvious that a lot of online retailers, such as Amazon, have been utilizing recommender algorithms for quite some time now, a lot of newer or smaller websites still have a requirement for them. So here comes our biggest problem which is new businesses won't be able to compete with big giants as new businesses have less data or no new customer data at all. This problem is known as the cold start problem in the recommendation system. When initially there are not enough data we face this cold start problem. The recommendation system struggles to recommend products because there is not enough data to suggest to the user. Machine learning algorithms also struggle to run their algorithm when there are cold start problems. Collaborative filtering is a common approach for developing recommender systems that provide recommendations to users based on their interests and behavior. However, dealing with new users or things with little or no ratings or interactions is a significant difficulty. This is referred to as the cold start problem, and it can have an impact on the accuracy and diversity of the suggestions. A cold-start issue occurs when the recommender system cannot produce suggestions for a new user with no history. When the recommender system lacks adequate information to generate credible forecasts or suggestions for a person

or an item, the cold start problem develops. Depending on the source of the missing information, the cold start problem may be divided into three types: user cold start, item cold start, and system cold start. User cold start happens when the system is uninformed of a new or existing user's preferences or profile, whereas item cold start occurs when the system is unaware of the characteristics or quality of a new or existing item. In contrast, a system cold start occurs when the system is deployed for the first time with no ratings or interactions from any users or objects. Each sort of cold start has its own set of issues that necessitate particular solutions for the recommender system.

1.4 Research Objective

The aim is to prepare a multiple model that will predict the result of a recommendation system using machine learning algorithms to solve the cold start problem that recommendation systems usually face when there is no initial data. Our main objective is to take the product description information from a dataset and build recommendation systems to recommend similar types of products or goods based on user search products. We want to develop a recommendation system based on a content-based recommendation system using data from BigBasket data for this thesis. The objective of a product recommender system is to provide recommendations for items or goods that may be of interest to a group of users and to do so in a way that is both relevant and contextually appropriate. Typical product recommendation systems include a large amount of user data that they may utilize to train their models, such as collaborative filtering, nearest neighbor, naive Bayes, and other machine learning techniques. However, if you do not have past data, it is difficult for e-commerce to compete with other major giants who have a lot of data from which they can develop efficient recommendation systems. It is extremely improbable that new businesses would rely on recommendation systems when they have no prior data or history of users in any form other than what they are selling. Building a recommendation system is therefore difficult. So the primary goal or objective of this study is to solve the cold start problem that new e-commerce sites face when they don't have any user information by developing a system that will enable new users to become acquainted with its basic recommendation system. The main goal of this study is to design, build, and test an effective product recommendation system that uses unsupervised learning techniques to improve user engagement, satisfaction, and online retail environments. The findings of this research have uncovered a lot of noteworthy facts, which will be helpful to researchers of RS in the past, present, and future when it comes to analyzing and planning their research roadmaps. This research aims to address the following key aspects,Investigate innovative data preprocessing methods for handling noisy, sparse data sources to enhance the quality and reliability of recommendations. Create unsupervised learning algorithms, such as K-means clustering and Hierarchical Agglomerative Clustering to recommend products. Also, build a recommendation system using cosine similarity and Jaccard similarity. Establish comprehensive evaluation metrics, including similarity scores for cosine similarity and Jaccard similarity, and compare these two similarity scores. silhouette score for the clustering algorithm to check the effectiveness of the recommendation system.

Chapter 2

Literature Review

2.1 Related Work

1. The phrase "Product Recommendation System based on User Trustworthiness amp; Sentiment Analysis" The author has explored the reviews mining from e-commerce websites, which includes sentiment analysis and reviewer credibility verification, in Gunjeet Kaur Soor, Amey Morje, Rohit Dalal, and Deepali Vora .[8] Sentiment Analysis underwent several changes and trials, and the over-fitting difference was reduced to roughly 4. Clustering and association mining is used in a unique approach to hybrid recommendation systems by A. A. Kardan and M. Ebrahimi for content suggestions in asynchronous discussion groups. The authors' goal was to create a solution to the CBF problem, often known as the cold start issue.[2] The given algorithm functions as follows: User clusters are established during the early stage.These clusters are created by collaborative filtering using cosine similarity as the similarity measure. In the second step, each cluster is converted into a transactional database. This transactional database uses extended FP trees to identify frequently repeated sets of objects.
2. A recommender system for automated vending machines employing GA, k-means, DT, and Bayesian networks (BN) was proposed by Lin et al. To recommend locally sourced commodities, it used meta-heuristics with statistical, classification, and clustering approaches.[10] Making decisions to coordinate product allocation to storage compartments, product replenishment points and limitations at vending machines, and vehicle routes for inventory replenishment are all necessary in smart vending machine systems with vendor-managed stock since they all have an impact on system profit. This article looks into intelligent vending machine systems that switch out items when there is a shortage of supplies.
3. The authors of "Proceedings of the International Conference on Recent Advances in Computational Techniques" by Parvattikar, Suhasini, and Parasar [6] the proposed a solution in the paper is an attempt to resolve problems with existing recommender systems, such as the prevalence of the first-rater or gray sheep problems and their inability to scale. The usefulness of the fuzzy logic and association-based categorization algorithms used in the method is shown by the case study and deployment of the strategy in a tourism system.
4. Collaborative filtering based on workflow space by L. Zhen et al. emphasized that standard Collaborative filtering-based recommender systems often have modifications aimed toward daily situations.[1] In cooperative team settings, team mem-

bers frequently originate from different disciplines, each bringing unique expertise and skills to the table. As a result, their informational needs are also different. A method for efficiently recommending relevant information to the right people is also necessary for a collaborative recommendation.

5. Image-based service recommendation system: A JPEG-coefficient RFs approach.Ullah F, Zhang B, Khan RU.[4] The authors made an effort to showcase a two-phase image-based product recommendation system. Phase 1 teaches the fourth type of product classification. Phase 2 searches for comparable goods. Machine learning product class learning used Random Forest classifiers. They used JPEG coefficients to get picture properties. The Phase I model has a 75 percent accuracy rate. The Random Forest model boosts performance in the DL configuration with accurate predictions of 84 percent.

6." Building a trust-based doctor recommendation system on top of multilayer graph database"[5] the research focuses on two essential features of a doctor recommendation system: modeling patient–doctor links for rapid data access and modeling the database's trust factor. This study modeled the relationships between patients and physicians as multilayer graphs. The fast NoSQL graph database Neo4j verifies the multilayer patient–doctor graph data model. A relational model is contrasted with our multilayer graph model. Due to the complexity and irregularity of patient–doctor relationships, graph data architecture enables quicker data access than relational systems. Significant in this scenario are interactions between individual entities, not associations between similar types of entities as displayed in a table when a relational data model is transferred to a table. Consequently, joint operations are required to locate a patient-doctor relationship (such as a treatment session) in a relational data structure, increasing access time.[18] Graph models reduce access time by representing relationships between nodes as edges. The second feature of this study introduces the doctor recommendation trust element. Database state determines trust factor. Trust is measured without external data. Additionally, database updates update the trust factor. Real-world data implemented the multilayer graph model and trust factor.

7. As suggested by their recommendation model, Salina et al. [11] worked on gathering sentiment information from social user evaluations. Additionally, they created a new link called interpersonal feeling, which expresses users' friends' sentimental influence on them. They quantify the sentiment of the user and determine the reputation of the item by analyzing the sentiment distribution across users. Their research's findings indicate that the three emotional components considerably influence the prediction of rating. Additionally, it shows significant adjustments to current real-world dataset techniques.

8." Improving the Product Recommendation System based on Customer Interest for Online Shopping [9]Using Deep Reinforcement Learning." Using reinforcement learning and a learning-based technique, the main goal of this work is to recommend a product to the user based on the user's click data in 2021. They discovered that the proposed result has a relatively higher output than other state-of-the-art when comparing the achieved outcomes to other existing strategies in the recommendation system. They want to improve the performance of the system by adding more approaches for the next projects. also, make the system more efficient.

9." Toward Improving the Prediction Accuracy of Product Recommendation " by Shahbazi, Zeinab and Hazra, Debapriya and Park, Sejoon and Byun, Yung CheoSys-

tem Using Extreme Gradient" This study evaluated XGBoost-based cooperation filtering product recommendations based on user profile and click data.[7] The approach filters past purchases' suggestions before predicting. The recommended approach retrieved user purchases to improve prediction and categorization. Using the data of users it was an efficient recommendation system. Word2vec, which normalizes data, is vital to data preparation and processing. Their model performs well, as shown by comparative studies and the XGBoost recommendation model. Based on the expected weight, the system suggests neighboring items.

10. Flipkart Product Recommendation System." In this study, the author examined multiple studies' methodology, approach, and algorithmic aspects.[9]After that, they carried out the same studies. Consumers and purchasers are more focused on the "things" and "quality" of search engine suggestions, said the study. Recommendation engines will benefit from cognitive computing. building recommendation systems using popular things. This shouldn't be the end of it. Variety is crucial since repetition runs the risk of boring clients. Precision may be improved by expanding the regions the report suggests.

11. The authors of "Facing the cold start problem in recommender systems" by Lika,Kolomvatsos, and Hadjiefthymiades proposed a solution for RSs using CF to relieve the new user cold start problem. To give forecasts for new users, the suggested system uses a three-phase technique.[3] We employ a method that considers their demographic data and, using similarity approaches, discovers the user's "neighbors." We define 'neighbours' as users who have comparable traits with the new user.The theory is that persons with comparable backgrounds and qualities are more likely to share similar tastes. As a consequence, each new user is assigned to a group, and a rating prediction system is in charge of generating ratings for things. The final scores are derived using a weighted method in which developers can prioritize certain qualities or choose a more 'fair' approach.Our experimental findings demonstrate the effectiveness of the offered strategies. The dataset given by the GroupLens research team is used. When a large number of users are already enrolled in the system, the suggested system operates better. In such instances, the algorithm achieves lower MAE values, boosting the forecast accuracy of ratings.

12. "A survey on solving cold start problem in Recommender Systems" According to Gope and Jain, the cold start problem expresses itself in two ways: new user cold start problem and new user cold start problem.[12] All solutions to the cold start problem for new users acquire missing user information. Existing solutions are categorized into two groups based on whether they obtain missing knowledge explicitly or indirectly, according to the literature study. The techniques are contrasted, and their benefits and drawbacks are listed. The purpose of this research is to inform readers about the cold start problem in recommender systems and the different solutions. The problem is still open for a better solution, and interested researchers may begin to research it. [15]

Chapter 3

Description of the model

3.1 Methodology

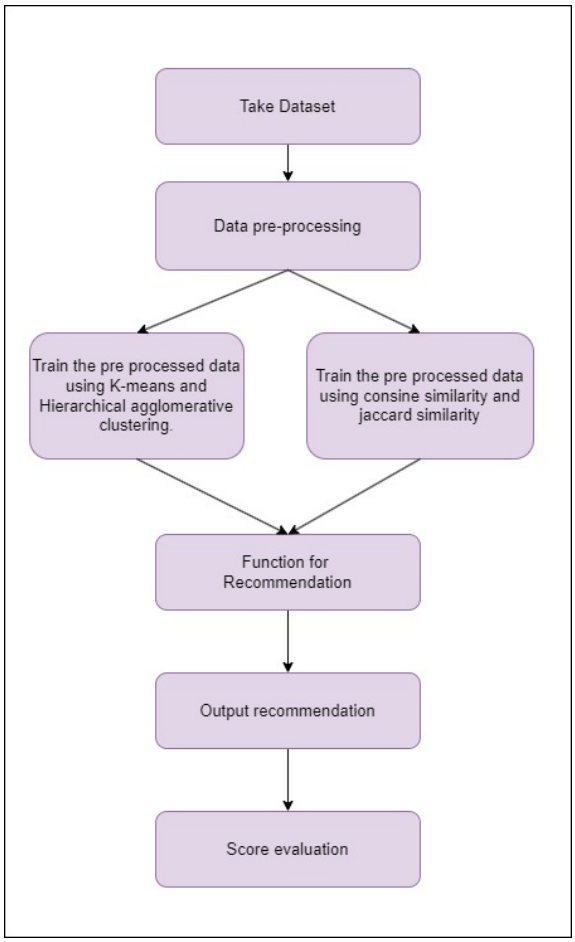

Figure 3.1: Methodology

At first, we created a data frame using the Pandas Python library. We used the product description column and used Tf-idf to transform text to the numerical value. At first, we used cosine similarity and Jaccard similarity to build a recommendation system using the value we got from the vectorization. Then we used the K-means clustering algorithm and hierarchical agglomerative clustering algorithm and gave the numerical data as input. This way we trained the K-means model and created

clusters from the model. We build a function that predicts the closest cluster from the clusters created for the recommendation. We use a target product and find the similarity within that closest cluster. For agglomerative we have treated each data point as its own group and then merged the two most similar groups together. Based on that we've found the most similar recommendation using the most resonated cluster. Then we evaluated the score from the models that we've built.

3.2 Description of the data:

Attempt to direct us in the direction of a dataset that will be useful in this case for resolving cold start concerns and many more recommendation systems as per our problem definition. Therefore, we expected to work on a specific dataset with information about product descriptions, which will be a key feature to address cold start, as well as other features to develop later a more effective recommendation engine. [13] Therefore, we choose a data collection that has some notable data that

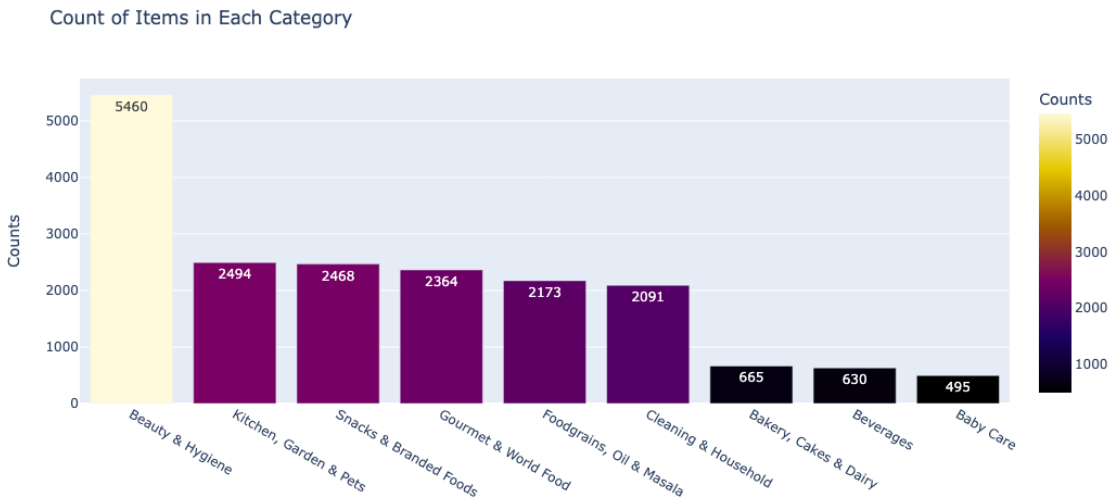

Figure 3.2: Category

we found on Kaggle. The biggest online grocery store in India, BigBasket, also has a huge variety of data to address these kinds of problems. The data sets are organized into categories and subcategories with product information. For example, "Brass Angle Deep Plain, No.2" is a product that falls under the "Cleaning and Household" category and the "pooja needs" subcategory. This product's listing also includes the brand name, sales price, market price, type, rating, and description. Some important columns to use when determining a recommendation are the product's attributes like rating and price. If we provide more information about the dataset that includes: [13] **This dataset includes 10 simple attributes, each of which is explained as follows:**
index - The Index product alone - The product's name (as it appears on the listing)
category - The category a product has been assigned to.
sub-category - The category that the product has been placed in.
brand - The product's brand sale-price - The product's price on the website.

Market price - The product's market price.
type- Select the category under which the product falls.
rating- The rating the product received from its customers' descriptions
description- A detailed explanation of the dataset If we examine the data in the bar chart below, we can see that the chart incorporates some of the categorical data from the dataset, such as beauty and hygiene items (5460 units), kitchen, garden, and pets (2468 units), and so on. However, the lowest of the supplied categories is baby care (495 units), the greatest is beauty and hygiene items (5460 units), and the remainder is typical. So the cosine similarity will most likely produce similarities in the area of beauty and hygiene and will be less likely to provide similar products in the category of baby care.

3.3 Data pre-processing

The initial step in the data preprocessing function involves breaking down the input text into its fundamental components, often referred to as tokens. Think of this process as disassembling a puzzle into its individual pieces.Next, we convert all the tokens to lowercase, like standardizing the puzzle pieces to have the same color, ensuring consistency and removing any potential biases related to capitalization.Following this, we filter out common and insignificant words known as stopwords. These words are akin to common puzzle pieces that don't contribute much to the overall picture and can be removed to focus on the more meaningful ones.Then comes lemmatization, a process that further simplifies the words to their root form, essentially reducing variations. This is like organizing similar puzzle pieces together, creating a more compact and organized set.To maintain efficiency and avoid unnecessary repetition, we ensure each type of puzzle piece is unique, just like eliminating duplicate tokens, so we don't have extra identical puzzle pieces lying around. In summary, this preprocessing is like preparing puzzle pieces for assembly, making sure they fit well and represent the picture accurately, without unnecessary or redundant pieces. It's all about organizing and simplifying the pieces for effective analysis and understanding.

3.4 K-means clustering:

The K-means algorithm is a clustering method and takes input data points and divides them into k clusters. As a consequence, the model would accept a data sample as input and return the cluster to which the new data point belongs based on the training that the model went through.[16]
Step 1: we need to specify the value of k which is the number of clusters we want to create. At first, this means algorithms randomly initialize the positions of k centroids in the feature space. Suppose we take k=3 so there will be 3 distinct clusters.

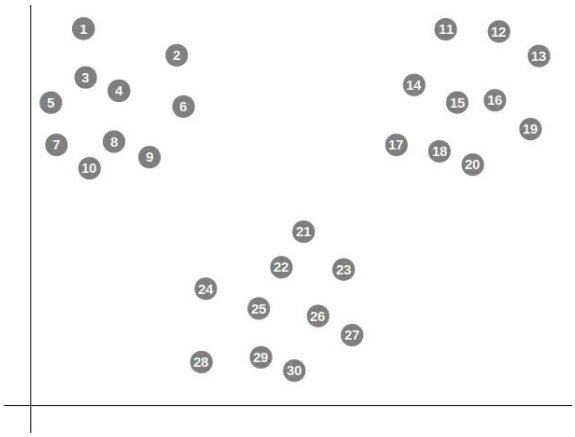

Figure 3.3: Datasets

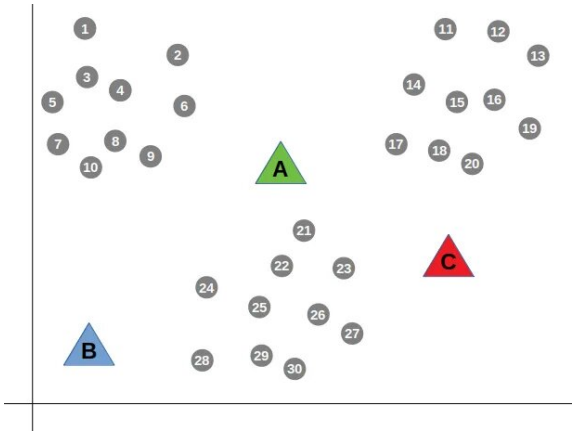

Figure 3.4: Data Points

Step 2: Now we measure the distance between the points and the three initial clusters. From this distance, the nearest data points will go to the nearest cluster. this way data points distance are measured and assign the point to the nearest cluster.
Step 3: Recalculate the centroids' locations based on the average of the data points inside each cluster once all the data points have been assigned to clusters. By averaging the coordinates of all the data points allocated to that centroid, the new centroid positions are calculated.
Step 4: Repeat the step until each data point is assigned to the nearest centroid from the updated positions.
Step 5: Steps 3 and 4 are repeated iteratively until the positions of the centroids no longer change significantly or when a maximum number of iterations is reached.
Step 6: The algorithm terminates, and the final clustering result is obtained. Each data point now belongs to one of the k clusters based on distinct features and its proximity to the centroids.

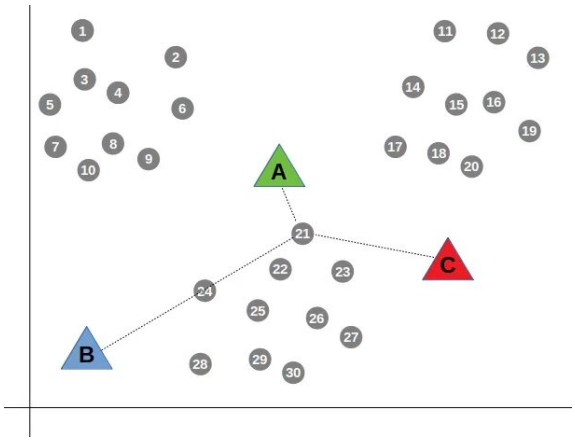

Figure 3.5: In-Corporating Clusters Centroids

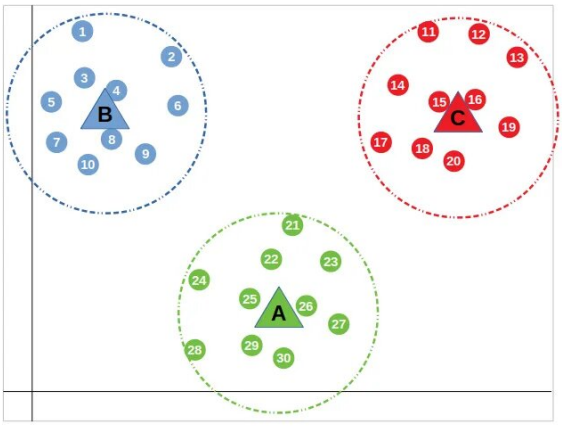

Figure 3.6: Centroid Distance Measuring

3.5 Hierarchical Agglomerative Clustering:

In this study Bottom-up hierarchical agglomerative clustering is used. Initially, each item is considered a single-element cluster. At the start of the method, algorithms treat each data as a singleton cluster and then begin a loop that creates comparable pairings of clusters until all clusters have been merged into a single cluster that contains all data. The two clusters that are the most comparable are joined into a new larger cluster at each phase of the method. This method is repeated until all points belong to a single large cluster. Step 1: Consider each number to be a separate cluster and compute the distance between one cluster and all the others.
Step 2: Comparable clusters are combined to produce a single cluster in the second phase. Let's assume clusters (2) and (3) are highly similar, so we merge them in the second phase, just like clusters (4) and (5), and at the end of this process, we obtain the clusters [(1), (23), (45), (6)].
Step 3: We recalculate the closeness using the technique and combine the two nearest clusters ([(45), (6)]) to produce new clusters as [(1), (23), (456)].
Step 4: Repeat the process until clusters (456) and (23) are comparable and can be joined to produce a new cluster. We are now down to clusters [(1), (23456)].
Step 5: Finally, the two remaining clusters are combined to form a single cluster [(123456)].

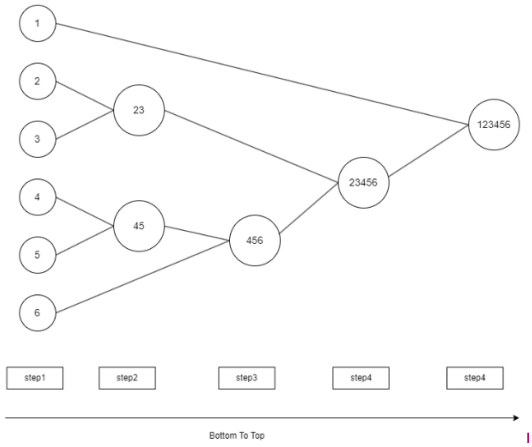

Figure 3.7: Hierarchical Agglomerative Clustering Model

3.6 Cosine similarities

Cosine similarity, in the landscape of text analytics and natural language processing, stands out as an indispensable metric. Irrespective of document size, this metric offers a nuanced approach to determining the similarity between two pieces of content. Conceptually, cosine similarity computes the cosine of the angle between two vectors positioned within a multidimensional space. These vectors are essentially numerical arrays representing word counts from distinct publications or descriptions. A pivotal characteristic of cosine similarity is its emphasis on orientation rather than magnitude. To discern magnitude, analysts often resort to the Euclidean distance.An illustrative scenario can be the word "cricket". If it appears 50 times in one document and 10 times in another, the Euclidean distance may show them as distinct due to sheer size differences. However, cosine similarity might reveal a smaller angle between them, signifying high content resemblance. As this angle diminishes, the similarity or resemblance escalates.[14]
Mathematically, cosine similarity is articulated as:

$$\text{Cos}\,\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}} \tag{3.1}$$

where, $\vec{a}\cdot\vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$
By weaving cosine similarity into our product recommendation system, particularly harnessing the product description attribute, we unlock an advanced method to quantify similarity amidst product descriptions. Our journey commenced with the TFIDF vectorizer, a technique that transformed raw descriptions into an organized matrix of vectors. Using these vectors, we delved into the cosine similarity function, generating scores that span from 0 (absolute dissimilarity) to 1 (utter similarity). To broaden our analytical horizon, we devised a similarity score matrix, juxtaposing each product description with every other in our dataset. This matrix not only becomes an analytical cornerstone but also paves the way for curating clusters of like products. As businesses strive for customer-centricity, such clusters can drastically refine recommendation engines, ensuring users find products that genuinely resonate with their preferences.

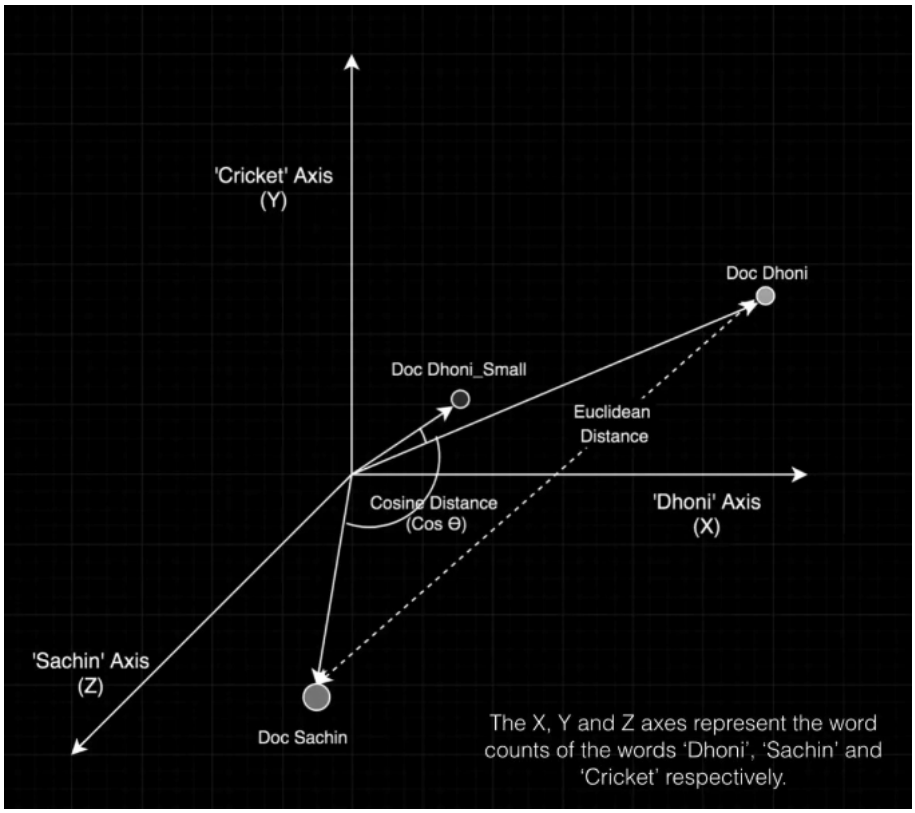

Figure 3.8: cosine similarities

3.7 Jaccard Similarity

Jaccard Similarity is a popular measure for determining the similarity of two things, such as two text texts. This metric is very useful for comparing collections of items. It is represented by the letter 'J' and is also known as the Jaccard Index, Jaccard Coefficient, Jaccard Dissimilarity, and Jaccard Distance. The intersection of two sets divided by their union is properly described as Jaccard Similarity, which particularly refers to the number of common words divided by the total number of words. In practice, we use word sets to discover this intersection and union.Jaccard Similarity is a useful method for determining the similarity of two asymmetric binary variables. In the context of binary variables, '0' represents the lack of a specific property, whereas '1' represents its existence. While both states are equally important for symmetric binary characteristics, this balance does not hold true for asymmetric binary variables.[17]

It is calculated by the formula: Jaccard Similarity = (number of observations in both sets) / (number in either set) or mathematically,

$$
\begin{aligned}
&\mathrm{J(d1,\ d2)} = |\mathrm{d1} \cap \mathrm{d2}|/|\mathrm{d1} \cup \mathrm{d2}| \\
&J\left(\mathrm{d_1}, \mathrm{d_2}\right) = \frac{\{\text{'data','is','the','new','oil','of','digital','economy'}\} \cap \{\text{'data','is','a','new','oil'}\}}{\{\text{'data','is','the','new','oil','of','digital','economy'}\} \bigcup \{\text{'data','is','a','new','oil'}\}} \\
&= \frac{\{\text{'data','is','new','oil'}\}}{\{\text{'data','a','of','is','economy','the','new','digital','oil'}\}} \\
&= \frac{4}{9} = 0.444
\end{aligned}
\tag{3.2}
$$

If two datasets d1 and d2 have exactly the same members, their Jaccard Similarity Index is 1, and if there are no shared members, the Jaccard Similarity Index is 0. Jaccard Similarity will tell us how many features in the dataset are comparable to each other.

Chapter 4

Result and Analysis

We've stratified our recommendation strategies into four distinct categories, each representing a unique method for deriving recommendations. Additionally, we've delineated between similarity-based comparisons and clustering-based analyses. In conclusion, we highlight the most optimal approach and provide the rationale underpinning our selection.

4.1 Data Quality and Analysis:

During our analysis, we often stumbled upon data quality issues. These ranged from gaps in data to inconsistent descriptions, not to mention varied ways products were depicted. Add to this the challenge of outliers, which you can see in the next image. Such disarray made clustering, especially with techniques like K-means and the Agglomerative Hierarchical algorithm, a tough nut to crack. Clearly, we can't overlook these hurdles. Fine-tuning our data be it through cleaning up text, filling in missing details, or crafting new features can significantly sharpen our similarity and clustering tools. While both Cosine and Jaccard similarities hold their own in this chaos, it seemed that Cosine Similarity took an extra leap forward when we polished the text, especially by pruning out stop words and applying lemmatization. An interesting observation was that descriptive or richer data seemed to gel better with similarity techniques than with clustering. But let's park that thought for now; we'll circle back to it later in our analysis.

4.2 Evaluating cosine similarity:

Cosine similarity works well with a text-based recommendation system as it measures the cosine angle between two vectors in high dimensional space. Furthermore, It effectively captures the orientation of vectors, making it sensitive to the direction of term frequencies in text. So in our case, as we have intended to find a break in solving cold start we needed to use only the description part of products. As a result, cosine similarity made a great impact. Now at first, we have created a basic similarity score model which is considered in the picture as an old recommendation, and also compared it with a new recommendation which is based on cosine similarity. And for the old recommendation, we can see that our search was chocolate yet we got Cashews and Nuts recommended. We need to optimize this based on category,

subcategory, and brand. So we have merged this and based on these merged data points we have calculated cosine similarity which shows prominent similarity and diversity. which is visible in the below picture.

```
old_rec = get_recommendations_1('Cadbury Perk - Chocolate Bar').values
new_rec = get_recommendations_2('Cadbury Perk - Chocolate Bar', cosine_sim2).values
pd.DataFrame({'Old Recommendor': old_rec,'New Recommendor':new_rec})
```

	Old Recommendor	New Recommendor
0	Cadbury Perk - Chocolate Bar	Nutties Chocolate Pack
1	Choco Stick - Hexagon Pack	5 Star Chocolate Bar
2	Luvit Chocwich White Home Delights 187 g	Dairy Milk Silk - Hazelnut Chocolate Bar
3	Luvit Chocwich Home Delights 187 g	Perk - Chocolate, Home Treats, 175.5 g, 27 Units
4	Wafer Biscuits - Chocolate Flavor	Dark Milk Chocolate Bar
5	Drinking Chocolate - Original	Dairy Milk Silk Mousse - Chocolate Bar
6	Drinking Chocolate - Original	Dark Milk Chocolate Bar
7	Biscuit - Bourbon Creams	Chocolate Bar - Fuse
8	Wafers With Hazelnut Cream	Choclairs Gold Coffee
9	Choco Stick - Chocolate	5 Star Chocolate Home Pack, 200 g, 20 units

Figure 4.1: Evaluating cosine similarity

4.3 Evaluating Jaccard similarity

Jaccard similarity is particularly effective when dealing with data represented as sets, such as text data where you treat each document as a set of terms. It measures the similarity based on the intersection and union of sets, making it a valuable choice for text-based recommendation. Like cosine similarity, Jaccard similarity is scale-invariant. It focuses on shared elements between sets rather than their sizes, making it robust to variations in document length.

Jaccard similarity can effectively capture document similarity when considering the presence or absence of terms in product descriptions. It's particularly useful when term frequency information isn't as important as the mere existence of terms. Now in our case, Jaccard similarity has shown a protruding result but not as good as

```
Evening Primrose Oil - Vegetarian Capsule (500 mg)
Natural Moisturising Lotion, Enriched with Echinacea & Aloevera
Natural Moisturising Lotion, Enriched with Echinacea & Aloevera
Coconut Oil - 100 % Pure
Bathing Soap (Lavender & Milk Cream)
Moisturise Lotion - Body Cocoon
Laboratory Reagent CH3, CO, CH3
Oil - Gingelly
Cotton Balls
Fruity Soap Enriched with Narural Grape Extract
```

Figure 4.2: Evaluating Jaccard similarity

cosine similarity. We have selected a target index which is a product named “Garlic Oil - Vegetarian Capsule 500 mg” and after using Jaccard we have got results similar to it but also some nearer products that are not directly related to garlic oil but showing nearer diversified words as visible in the figure 4.2 .

4.4 Comparing Cosine and Jaccard similarity

For the comparison part we have divided the comparison into multiple segments. which are below:
1. The similarity scores between the two methods.
2. The unique recommendations count for each method.
3. The overlapping recommendations count.
4. Coverage for each method.

4.5 The similarity scores between the two methods

For each target product, We have calculated recommendations using both cosine similarity and Jaccard similarity.To calculate the similarity score between the two methods, we find the number of overlapping recommendations,the recommendations that both methods have in common and divide it by the total number of recommendations from the cosine similarity method. This ratio provides a measure of how similar the recommendations are between the two methods. In this case, we have a bit more similarity score for Jaccard than cosine which is visible in the below plot:

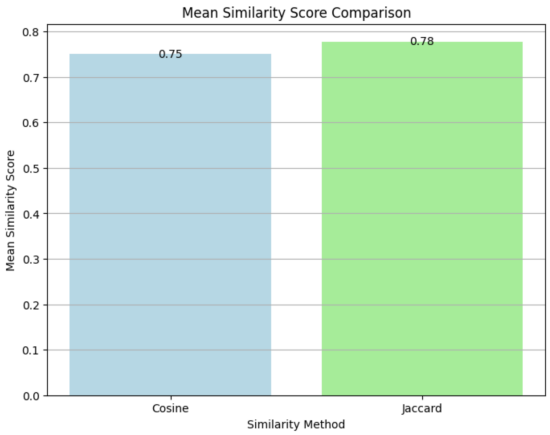

Figure 4.3: Mean similarity score comparison

4.6 Unique Recommendations Count for Each Method

For each target product, we have calculated the number of unique recommendations generated by both cosine similarity and Jaccard similarity methods. Moreover,

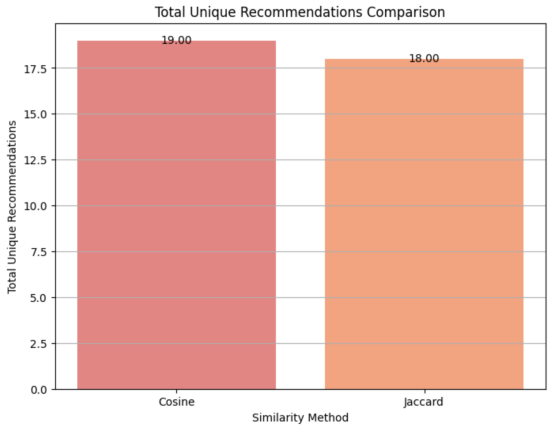

Figure 4.4: Total unique recommendation comparison

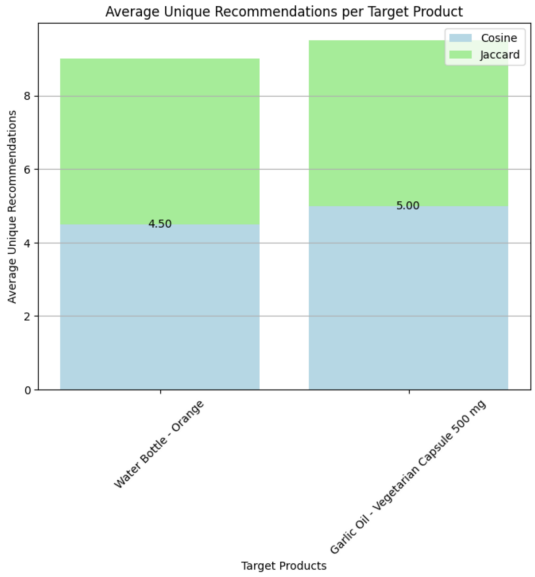

Figure 4.5: Average unique recommendation per target product

We have summed up these unique counts for all target products to get the total unique recommendations count for each method. The results we got according to it is that cosine similarity promises to show more unique recommendations which is about 19 recommendations in a set and 18 for jaccard. This tells us that cosine similarity provides a more diversified recommendation. but overall the average tells us differently although it varies from product to product.

4.7 Overlapping Recommendations Count

We have tried to find the recommendations generated by both cosine similarity and Jaccard similarity methods for each target product. Then, calculate the number of recommendations that are common (overlapping) between the two methods. This provides insights into how many recommendations both methods agree upon. which is clearly visible in the below picture.

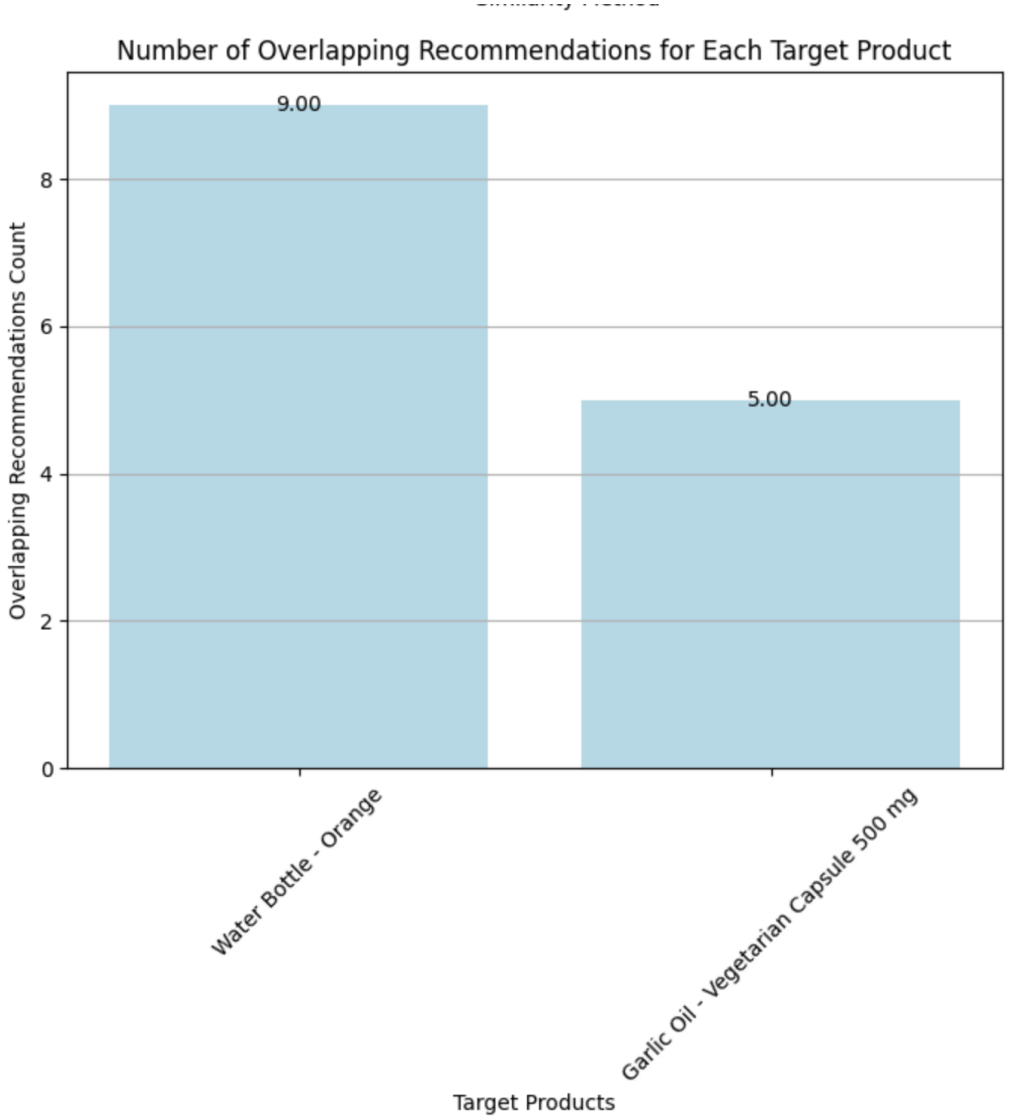

Figure 4.6: Overlapping Recommendations Count

4.8 Coverage for Each Method

Coverage is calculated as the percentage of unique recommendations generated by a method relative to the total number of unique products in the dataset. For each method (cosine and Jaccard), we have calculated the coverage by dividing the number of unique recommendations by the total number of unique products and then in percentage. This metric helps us to understand what proportion of the product catalog is covered by the recommendations of each method. which is 6 percent for both.

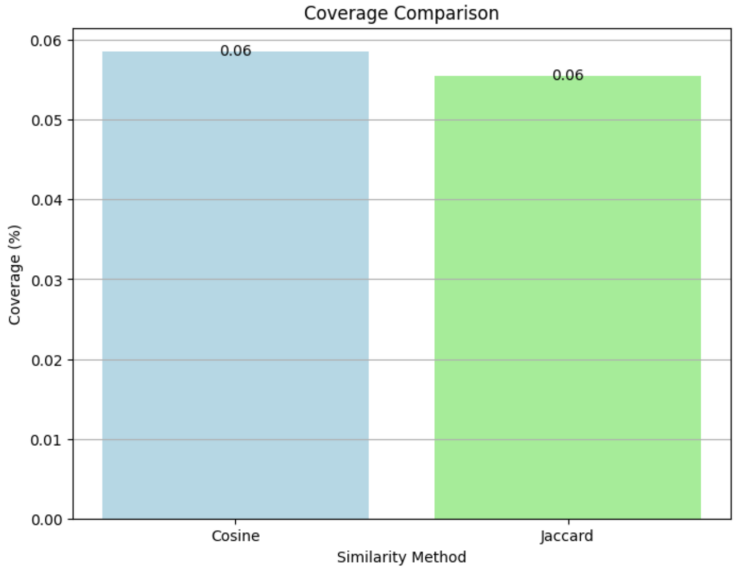

Figure 4.7: Coverage comparison

4.9 Evaluating K-means clustering

We have evaluated K-means clustering in three ways. First use only the product description without outliers to get optimal silhouette score second use only the product description with outliers to get recommendations better third merge the product description, category, and subcategories so that we get the clusters more refined than the other two. Silhouette scores provide a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). In other words, the silhouette score helps you evaluate the compactness and separation of clusters. In the below box plot, you can see 30 clusters by k-means and the silhouette score in this case. Here for noise cancelation (eps=5, min samples=5). Now the first method removes outliers to get better silhouette scores In order to

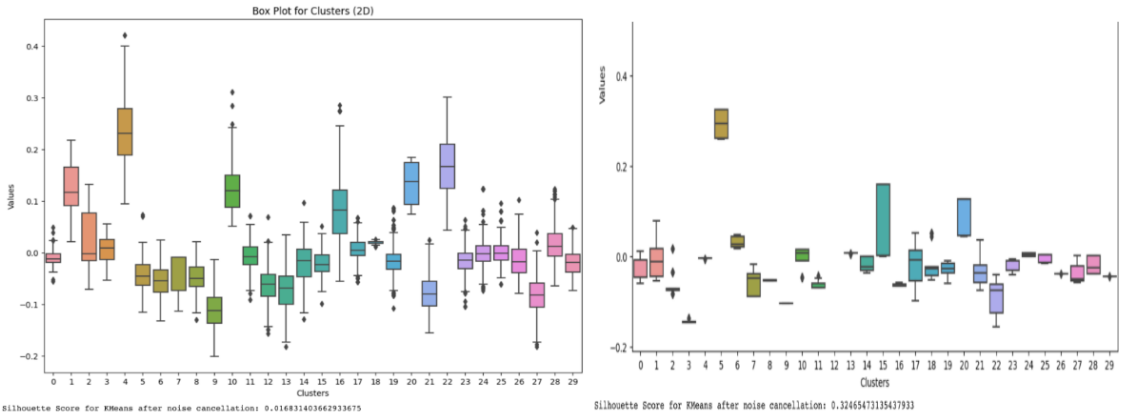

Figure 4.8: Box plot for cluster 2D

do that we can use DBScan which is also a clustering algorithm. Now there are two parameters we need to keep in check EPS value and min samples if we want to implement noise cancelation on data. So if we tweak these two values according to the dataset and what goals we intend to achieve we'll get the desired clusters with

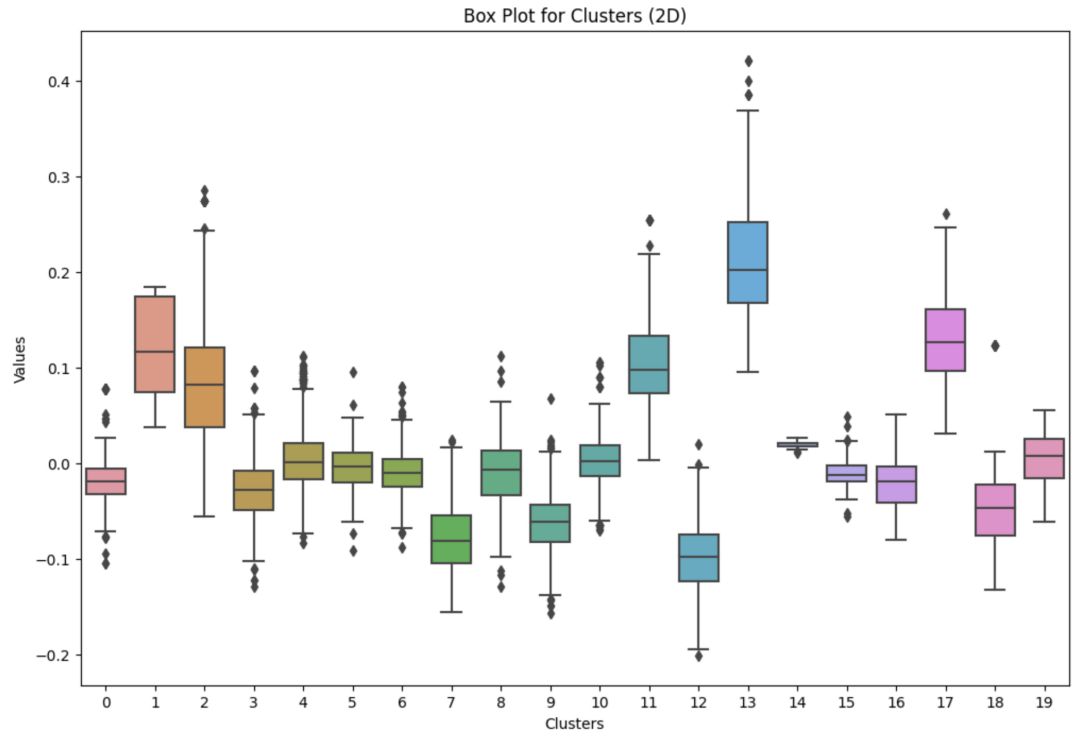

Figure 4.9: Box plot for cluster 2D

fewer outliers with better silhouette scores and also average Intra-cluster Similarity and average Inter-cluster dis-similarity.In this case, we have calculated silhouette scores of [0.01683] which vary on the noise of the data and also how many clusters were made, and what the optimal clustering is. Also after manipulating the noise parameters(eps=0.5, min samples=5,n clusters = 30), we get clusters like figure 4.8 and a silhouette score of [0.324654].
Now In our second approach, which is visible in Figure 4.9, we applied K-means clustering to the entire dataset and then used the recommendation function to find out, if the recommendations were based on the clustering structure created by K-means. However, this approach may not always capture fine scores but provides more accurate results when we test out using a target product leading to comparatively optimal recommendations than the first. We have used "Water Bottle - Orange" a product from our dataset and the recommendation we have got according to the product is visible below picture:

```
Target Product:
Water Bottle - Orange
Top 10 closest products to the target product by K-means:
Rectangular Plastic Container - With Lid, Multicolour
Jar - With Lid, Yellow
Premium Round Plastic Container With Lid - Yellow
Premium Rectangular Plastic Container With Lid - Multicolour
Premium Rectangular Plastic Container With Lid - Multicolour
Premium Plastic Jar With Lid - Green
Premium Plastic Jar With Lid - Green
Plastic Round Glass With Lid - Yellow
Plastic Container - Square, Pink
Plastic Round Glass With Lid - Pink
```

Figure 4.10: Recommendation results Second approach

Cluster 0:	Cluster 7:	Cluster 3:	Cluster 10:
teeth	taste	hair	baby
toothpaste	snack	shampoo	diaper
toothbrush	delicious	scalp	pants
oral	sweet	oil	skin
colgate	tasty	colour	soft
bristles	flavours	conditioner	pampers
gums	cookies	dandruff	diapers
brushing	healthy	natural	delicate
brush	enjoy	fall	wetness
plaque	crunchy	shine	comfortable

Figure 4.11: K-means third approach clusters

In the third approach, we used data with outliers to get adequate clusters and accurate recommendation results rather than having a good score. Our goal is to achieve a minimal silhouette score which is used to assess the quality of clusters formed by a clustering algorithm but with a near-accurate recommendation.

Furthermore, In the third approach where we have selected a specific product

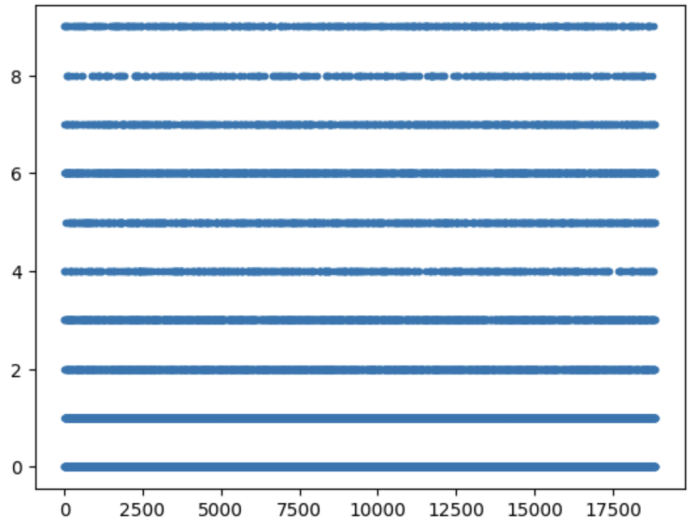

Figure 4.12: cluster third approach

and identified its cluster using K-Means, we directly retrieved the top terms within that cluster. This approach is more focused on finding items that are most similar to the given product within the same cluster, which can lead to more relevant recommendations. The below picture explains the cluster condition and how well they are separated and also if we use an input we are getting the closest cluster of that input which is basically the recommendation in this case.

```
show_recommendations("steel")
```

```
Cluster 18:
 steel
 stainless
 quality
 high
 durable
 food
 easy
 kitchen
 home
 grade
```

Figure 4.13: recommendation result third approach

4.10 Cluster and recommendation results

Furthermore, intra-cluster similarity and Inter-cluster Dissimilarity is measured. High intra-cluster similarity indicates that products within a cluster are similar to each other, which is desirable for recommendation systems. Low inter-cluster similarity suggests that products from different clusters are dissimilar, which is also a desirable outcome. Overall judging by the recommendation results we have got, we are optimistic with the second approach of k-means as the recommendations results are very fine-grained. The intra-cluster and Inter-cluster Dissimilarity for this approach is visible in figure 4.14.

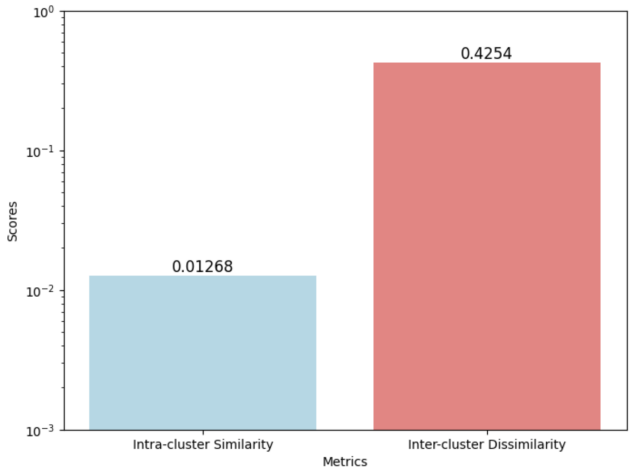

Figure 4.14: Cluster and recommendation results

The effectiveness of K-Means for recommendation depends on how you use it and whether you are interested in finding similar items within a specific cluster or across the entire dataset. All approaches have their merits, and you can choose the one that aligns better with your recommendation goals. Additionally, you can experiment with different clustering algorithms and similarity metrics to further improve recommendation accuracy.

4.11 Evaluating Agglomerative clustering

Agglomerative clustering is a hierarchical clustering method that starts with each data point as its cluster and successively merges clusters until only one cluster remains. Agglomerative clustering results are comparatively better than the second approach of K-means because we are getting diversified recommendations in agglomerative .So we have tested the recommendation by putting an input of a product which is 'Water Bottle - Orange' same as we have shown in K-means second approach and the recommendation we have got is in the below picture is up to the mark like cosine similarity and Jaccard similarity method but diversified results.

```
Target Product:
Water Bottle - Orange
Top closest products to the target product by Agglomerative Clustering:
Water Bottle - Orange
Cereal Flip Lid Container/Storage Jar - Assorted Colour
Pet Solitaire Container Set - Silver
Premium Square Plastic Container - Green
Solitaire Storage Transparent Pet Container Set - Green Lid
Cereal Flip Lid Container/Storage Jar - Assorted Colour
Premium Flat Square Plastic Container Set - Blue
Pet Selo Container Set - Silver
```

Figure 4.15: Recommendation results agglomerative clustering

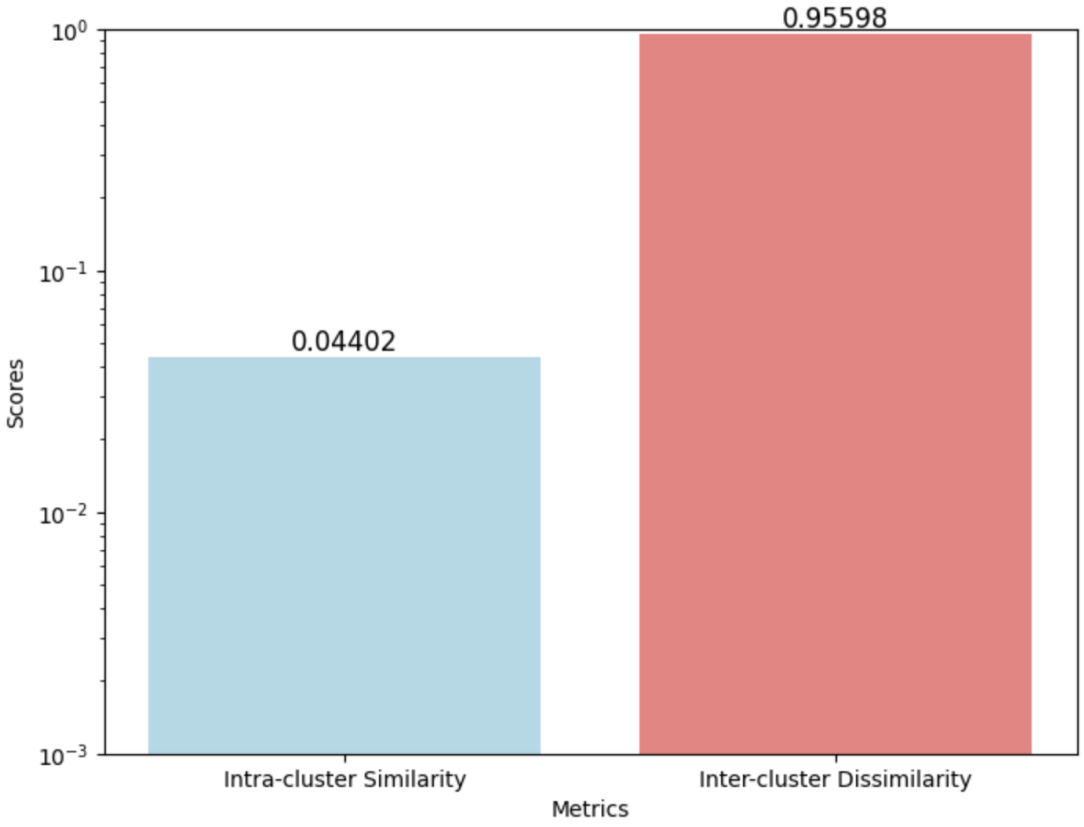

Figure 4.16: Clustering algorithm

For agglomerative clustering, we have also measured the intra-cluster similarity and Inter-cluster Dissimilarity which is Average intra-cluster similarity [0.04401] and

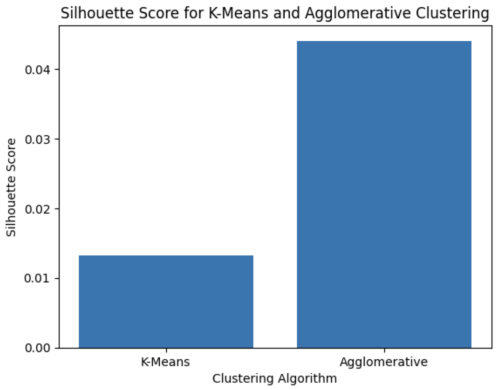

Figure 4.17: Clustering algorithm

Average inter-cluster dissimilarity [0.9559]. The diversified results are far more in this case as a result the Average intra-cluster similarity [0.04401] is comparatively lower.Now if we compare the silhouette score for both k-means and agglomerative clustering algorithm. agglomerative has an edge when it comes to silhouette score which is [0.044] on the other hand K-means has a score of [0.01328]. So overall we can say that any approach is applicable and it solely depends on user needs. If users want more diversified recommendations than agglomerative or if users want similar types of recommendations then K-means approaches can be used.

4.12 Merged Recommendation

We have used multiple approaches to solve our objective but if we use a merged model we combine all the approaches together for a far better outcome of the model and get a result that will amplify the recommendation. It will not only increase the outcome of recommendation by having used multiple methods but also will provide a diversity of different models, by dropping the common recommendations of the different models. Overall it takes the recommendation to a whole new level. We have given the merged model 50 percent weight to cosine similarity to keep the model diversity in check furthermore 30 percent weight to the agglomerative model to have fine-grained results and also to keep a priority on similarity we have given 20 percent weight to our K-means model, making the overall merged model way more enhanced to solve our objective. The protruding results of the merged recommendation are visible in the below figure.

```
Merged Recommendations (50% Cosine, 30% agglo, 20%kmeans):
1. Glass Water Bottle - Aquaria Organic Purple - cosine
2. Glass Water Bottle With Round Base - Transparent, B1364 - cosine
3. H2O Unbreakable Water Bottle - Pink - cosine
4. Water Bottle H2O Purple - cosine
5. H2O Unbreakable Water Bottle - Green - cosine
6. Water Bottle - Orange - agglo
7. Premium Square Plastic Container - Green - agglo
8. Solitaire Storage Transparent Pet Container Set - Green Lid - agglo
9. Rectangular Plastic Container - With Lid, Multicolour - kmeans
10. Jar - With Lid, Yellow - kmeans
```

Figure 4.18: Merged Recommendation

Chapter 5

Conclusion

A recommender system is a tool that aids people in finding information that is pertinent to them. Collaborative filtering, content-based filtering, and hybrid recommender systems are only a few examples of the various types of recommender systems. However, some challenges might arise while developing a recommendation system, including cold start, shilling attack, latency, sparsity, and grey sheep. In our findings several distance similarity methods such as cosine similarity and jaccard clustering algorithms were used to train and forecast the data in order to overcome this issue. A product-based recommender system's main goal is to successfully suggest a product that the user or consumer wants. The wholesale and retail industries are changing quickly as internet merchants, brands that sell directly to customers, and subscription and membership services take the place of traditional brick-and-mortar storefronts. Recommender systems are gaining popularity as a way to enhance the entire purchasing experience and lengthen the time spent on a website. The primary goal of this research is to find multiple approaches in order to solve the cold start issue and provide a decision on which approach is feasible at certain times. The data collection is organized into categories and subcategories with product information. It also includes the brand name, sales price, market price, type, rating, and description. These attributes are important columns to use when determining a recommendation. We use the description column for the recommendation system and use text vectorization to transform the textual data into numerical representations and apply all the models to find out what is suitable for the recommendation system to perform better for recommending products.

After thorough experimentation with various approaches, this study arrives at a conclusion: when devising a recommendation system without relying on intricate user behavior analysis, employing similarity-based methods such as cosine similarity and Jaccard similarity stands out as a computationally lightweight yet highly effective strategy. These methods efficiently ascertain similarity and present recommendations that possess both striking resemblance and a tasteful variety.

Moreover, in cases where users seek nuanced recommendations with a balance of precision and diversity, the Agglomerative clustering approach emerges as the prime selection. It offers a fine-grained recommendation system, ensuring accuracy and a rich diversity of suggestions.

However, for users placing a premium on similarity with their own products as a basis for recommendations, the K-means clustering approach outshines others. It excels in providing recommendations closely aligned with a user's specific product,

demonstrating its superior aptitude in this context. In essence, this study underlines the adaptability and distinct advantages of each approach, catering to various user preferences and system requirements, ultimately paving the way for an enhanced recommendation system.

5.1 Future Work

Looking ahead it holds promise to focus on improving the silhouette score by adjusting our clustering techniques. In addition, using filtering methods can improve the accuracy and precision of our product suggestions significantly. Collaborative filtering uses aggregate user behavior to generate educated suggestions, with the capacity to accurately adapt recommendations. To ensure integration and usability a crucial step is the development of an API. This API would allow access, to our recommendation system enabling e-commerce platforms to implement and leverage our recommendation algorithms. Our objective is to develop an e-commerce system that enhances user experience and encourages customer participation. Combining algorithms with similarity-based techniques is a fascinating area of research. By integrating these approaches into a model we anticipate achieving superior results. This collaboration is meant to provide a comprehensive perspective on product recommendations, which aligns with our goal of giving ideas that not only react to client preferences but also exceed expectations in terms of variety and satisfaction.

Bibliography

[1] L. Zhen, G. Q. Huang, and Z. Jiang, "Collaborative filtering based on workflow space," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7873–7881, 2009. DOI: 10.1016/j.eswa.2008.11.047. [Online]. Available: https://doi.org/10.101/j.eswa.2008.11.047.

[2] A. A. Kardan and M. Ebrahimi, "A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups," *Information Sciences*, vol. 219, pp. 93–110, 2013. DOI: 10.1016/j.ins.2012.07.011. [Online]. Available: https://dx.doi.org/10.1016/j.ins.2012.07.011.

[3] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 2065–2073, 2014, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2013.09.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417413007240.

[4] F. Ullah, B. Zhang, and R. U. Khan, "Image-based service recommendation system: A jpeg-coefficient rfs approach," *IEEE Access*, vol. 8, pp. 3308–3318, 2019.

[5] S. Mondal, A. Basu, and N. Mukherjee, "Building a trust-based doctor recommendation system on top of multilayer graph database," *Journal of Biomedical Informatics*, vol. 110, p. 103 549, 2020. [Online]. Available: https://doi.org/10.1016/j.jbi.2020.103549.

[6] S. Parvattikar, D. Parasar, *et al.*, "Recommendation system using machine learning," in *Proceedings of the International Conference on Recent Advances in Computational Techniques (IC-RACT)*, 2020. DOI: 10.2139/ssrn.3702439. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3702439.

[7] Z. Shahbazi, D. Hazra, S. Park, and Y. C. Byun, "Toward improving the prediction accuracy of product recommendation system using extreme gradient boosting and encoding approaches," *Symmetry*, vol. 12, no. 9, p. 1566, 2020. [Online]. Available: https://www.mdpi.com/2073-8994/12/9/1566.

[8] G. K. Soor, A. Morje, R. Dalal, and D. Vora, "Product recommendation system based on user trustworthiness & sentiment analysis," in *ITM Web of Conferences*, EDP Sciences, vol. 32, 2020, p. 03 030. DOI: 10.1051/itmconf/20203203030. [Online]. Available: https://doi.org/10.1051/itmconf/20203203030.

[9] T. Keerthana, T. Bhavani, N. S. Priya, V. S. Prathyusha, and K. S. Sri, "Flipkart product recommendation system," *transactions*, vol. 33, p. 34, 2021. [Online]. Available: https://jespublication.com/upload/2020-110470.pdf.

[10] L. Liu, J. Cui, Y. Huan, Z. Zou, X. Hu, and L. Zheng, "A design of smart unmanned vending machine for new retail based on binocular camera and machine vision," *IEEE Consumer Electronics Magazine*, vol. 11, no. 4, pp. 21–31, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9359457.

[11] *(pdf) product recommendation system from users reviews using sentiment analysis*, https://www.researchgate.net/publication/318493578_Product_Recommendation_System_from_Users_Reviews_using_Sentiment_Analysis, (Accessed on 05/21/2023).

[12] *A survey on solving cold start problem in recommender systems — ieee conference publication — ieee xplore*, https://ieeexplore.ieee.org/abstract/document/8229786, (Accessed on 09/17/2023).

[13] *Bigbasket entire product list (˜28k datapoints) — kaggle*, https://www.kaggle.com/datasets/surajjha101/bigbasket-entire-product-list-28k-datapoints, (Accessed on 05/21/2023).

[14] *Cosine similarity - understanding the math and how it works? (with python)*, https://www.machinelearningplus.com/nlp/cosine-similarity/, (Accessed on 05/21/2023).

[15] *Facing the cold start problem in recommender systems - sciencedirect*, https://www.sciencedirect.com/science/article/abs/pii/S0957417413007240, (Accessed on 09/23/2023).

[16] *How does k-means clustering in machine learning work? — by anas al-masri — towards data science*, https://towardsdatascience.com/how-does-k-means-clustering-in-machine-learning-work-fdaaaf5acfa0, (Accessed on 05/21/2023).

[17] *Jaccard similarity – learndatasci*, https://www.learndatasci.com/glossary/jaccard-similarity/, (Accessed on 09/17/2023).

[18] *Recommendation of influenced products using association rule mining: Neo4j as a case study — springerlink*, https://link.springer.com/article/10.1007/s42979-021-00460-8, (Accessed on 05/21/2023).