`

PRODUCT RECOMMENDATION SYSTEM FOR SUPERMARKET

Pranavi Satheesan

(IT17095136)

Bachelor of Science (Hons) in Information Technology

Specializing in Data Science

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

`

PRODUCT RECOMMENDATION SYSTEM FOR SUPERMARKET

Pranavi Satheesan

(IT17095136)

This dissertation was submitted in partial fulfillment of the requirements for the

Bachelor of Science (Honors) degree in Information Technology Specializing in Data Science

Bachelor of Science (Hons) in Information Technology

Specializing in Data Science

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

September 2019

`

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name: Pranavi Satheesan

Signature: Date: 2019-09-26

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Supervisor: Mr. Jesuthasan Alosius

Signature: Date: 2019-09-26

`

Abstract

Customers who seek the services at supermarkets are subjected to inconsistencies & ambiguities over choosing their desired products from a wide range of products with the closest quality. Meanwhile, supermarkets find it very difficult to satiate the customers' demand. Therefore, proposing a method to analyze customers' need plays an important role in attracting new and regular customers. The main goal of this study is to formulate a product recommendation system, which analyze customers' needs and thus recommend the best products. This system recommends products to the regular customers and to the new customers as well. New customers mean obviously the customers with no purchasing history at the supermarket in question. The system referred to recommends the products to the new customers using up two methods. One method recommends the most popular products while the other method solely focuses on product description for recommendation. The system recommends the products to the regular customers using up user-based collaborative filtering; item based collaborative filtering and association rule mining. It recommends products to regular customers based on purchasing history and priority ratings given by other users who bought the products. Initially, the recommendation algorithm finds a set of customers who purchased and rated the products that overlap with the user who purchased and rated the products. The algorithm aggregates products from the customers with similar preference and eliminates the products the user has already purchased or rated. The proposed methodology improves the shopping experience of customers by recommending accurately and efficiently the products that are personalized to the need of the customers.

KEYWORDS: recommendation system, collaborative filtering, cosine similarity, association rule mining, correlation

`

Acknowledgement

First and foremost, it is a genuine pleasure to express my deep sense of gratitude to my Research Supervisor, Mr. Jesuthasan Alosius for his fullest support, dedication, enthusiasm, and overwhelming desire he expressed throughout the project to make this research a success. I gratefully acknowledge his patience at the time of long piece of writing this report.

My sincere thanks go to Ms. Janani Tharmaseelan, my Co-Supervisor who enlightened me with her vast amount of knowledge and for her support and motivation during the monitoring of progress. I would also like to extend my heartfelt appreciation to Mr. Prasanna S. Haddela, Data Science Program Coordinator, for deserving his valuable time to guide and encourage me and I would like to thank Dr. Janaka Wijekoon, Lecturer-In-Charge of Comprehensive Design & Analysis Project for giving precise account on this module during the lecture hours.

I must also express my very profound gratitude to all staff at the SLIIT for their tireless support and encouragement given by them from the beginning and up to the end of my tenure with the duration of four years at SLIIT and with their fullest commitment I was able to make this project a success. Their art of teaching made me to have a strong impression on the course and I reminisce such experience forever in my life.

This work would have been a mere dream without the timely advice and necessary guidance from my parents, especially my mother, Mrs. Vanitha Satheesan, who is always with me throughout my ups and downs.

I would like to record my sincere gratitude to Mr.Rajeenthiran Thangenthiran for his support and motivation.

Finally, I would like to thank everyone who helped and encouraged me in various forms to make this project into the state of reality. Meanwhile, I owe an apology to all people whom I might have failed to mention in gratitude of their service at this juncture. My tenure at the SLIIT could be considered as a golden period as I have learnt a lot of things in my academic career and gained professional experience in life.

`

Table of Contents

`

List of Table

Table of Equation

Table of Figure

`

`

`

Acronyms/Abbreviation

Table 1-1:Acronyms/Abbreviation

Term	Definition
BFM	Basket sensitive Factorization Machine
CBFM	Constrained Basket sensitive Factorization Machine
SMACA	Single Cycle Multiple Attractor Cellular Automata
COREL	CustOmer purchase pREdiction model
SDLC	Software Development Life Cycle
WBS	Work Breakdown Structure
UML	Unified Modeling Language
TS	TypeScript

`

1. Introduction

This chapter provides a detailed description of the project overview, research background, research problems, aims and objectives.

1.1. Background Literature

1.1.1. Background

Recommendation systems play an important role in supermarkets. Recommender Systems are intelligent engines that gather information relevant to which product a customer has seen or bought previously, with an objective of providing personalized suggestions on unobserved items that are may likely to be of interest [1]. It analyses the needs of customers and suggest the best possible shopping list. Most of the customers would like to have recommendation system because they can gain access to know about the feedback given by other customers. Many applications use the products that customer purchased and explicitly rated to reflect their interests. There are three popular algorithms used in recommendation systems. They are collaborative filtering, cluster models and association rule mining. Collaborative filtering algorithms recommend products based on the opinions of other like-minded customers. There are two types of collaborative filtering. They are item-based collaborative filtering and user-based collaborative filtering. In item-based collaborative filtering, similarities between items are calculated using up cosine similarities. In user-based collaborative filtering, similarities between users are calculated using up cosine similarities. User-based method uses historical information to identify the neighborhood for the active customer. In consequent to this, the products are recommended according to their similarities to this neighborhood. User-based recommender systems use the customer profile data and they can incorporate demographics of customers along with the purchasing data history [2]. Recommendations for customers are computed by finding products that are similar to other products preferred by customers. Cluster methods are used either in case where a supermarket system is to recommend products to a new customer or in a new supermarket that is obviously lacking purchase history of customers. This recommendation is based on textual clustering analysis given in product description.

Nowadays, with the using up of data mining and applying the rules in this field, we can create models on data that reveal this implicit knowledge and pass on information to us [3]. We use data mining in supermarket because customers can have easy access to their desired products without consuming much time for search. Monitoring commercial transactions can mine

`

patterns as association rules to discover the potential relationship between the products in the store and subsequently suggest the products with similar quality to customers. In addition, product recommendation systems can monitor the history of purchasing behavior of customers, their preferences and predict the needs of customers and the products that are closely related with products that proposed to customers. Customers can also recognize products based on the recommendations that associate with their priorities, and make a final decision for the purchase. Hence, the recommendation system can be helpful to users to identify suitable products for their needs and preferences in an effective manner.

1.1.2. Literature Survey

I. A Real-Time Targeted Recommender System for Supermarkets

The paper [1] introduces an approach that combines Entropy-based algorithm, a Hard k-modes clustering and Bayesian Interference in order to minimize cold-start problem, data sparsity and other scalability issues. The approach captures the dynamic environment of a supermarket as users change their preferences time to time. The approach used traditional collaborative filtering technique to achieve this.

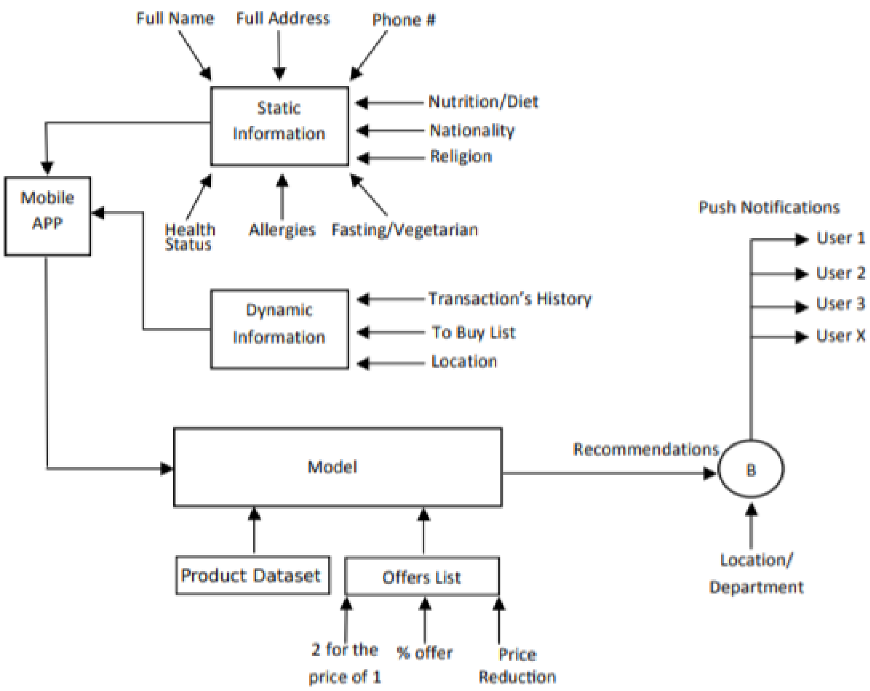


Figure 1:1: Overview of the Recommender System [1]

`

Figure 1.1 shows that static information of the customer such as full name, full address, phone number nationality nutrition etc. and dynamic information such as transaction history, buy list etc. fed in to the mobile app. Subsequently, a model was built using the product dataset and offers list to provide recommendation. When customers navigate into store, iBeacons push personalized notifications to the customer's mobile phone.

II. Enhancing Product Recommender Systems on Sparse Binary Data

The study [2] proposes an association rule mining-based recommender tool called e-VZpro which includes two-phase approach. In first phase, customer historical data are analyzed by using association rule mining. In the second phase, scoring algorithm is used to rank the recommendations for the customer. Two measures such as Pearson's correlation coefficient and cosine are used to find similarity between two customers.

III. Dihedral Product Recommendation System for E-commerce Using Data Mining Applications

The study [3] cluster the products and create groups that have similar characteristics. C-Means algorithm is used to cluster the products. Another technique called association rule mining is used to find the association rules through the customer behavior and purchase history. Figure 1.2 shows the relationship between products that was found after generating the association rules. When a customer selects particular product, the products that are linked with that particular products are recommended to the customer.

Figure 1:2: Relations between products [3]

IV. Basket-Sensitive Personalized Item Recommendation

The study [4] recommends items to be added to the basket. Two approaches are proposed here. First one exploring a factorization-based model called Basket sensitive Factorization Machine (BFM) that incorporates various types of associations involving the user, the target item to be recommended, and the items currently in the basket. Second one is based on the observation that various recommendations towards constructing the same basket with similar likelihoods, proposing another model called Constrained Basket sensitive Factorization Machine (CBFM) that further incorporates basket-level constraints.

V. A Product Recommendation System using Vector Space Model and Association Rule

In the research paper [5] system recommends the products to a new user. Recommendation of products depend on the purchase pattern of previous users whose purchase pattern are close to that of a new user. To find out the closest user profile among the profiles of all users the system uses vector space model and also use Single

`

Cycle Multiple Attractor Cellular Automata (SMACA) in implementing association rule. Figure 1.3 shows the FP-tree constructed to generate association rules.

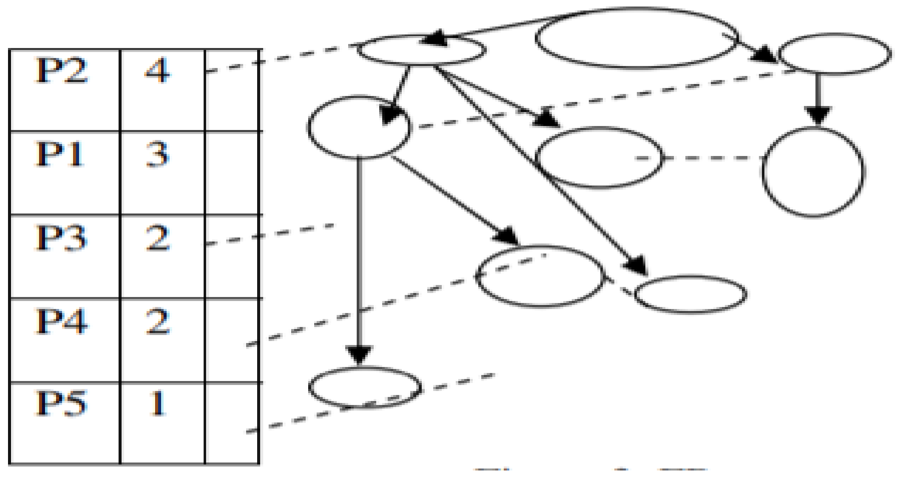


Figure 1:3: FP-tree [5]

VI. Large Scale Product Recommendation of Supermarket Ware Based on Customer Behavior Analysis

The article [6] proposes prediction model based on the behavior of each customer using data mining techniques. The model analyses the data in order to classify customers as well as products. The model classifies customers according to their consuming behavior and proposes new products more likely to be purchased by them. The prediction model is intended to be utilized as a tool for marketers to provide a targeted and specified consumer behavior. The figure 1.4 shows the input of dataset and then subsequent analysis of customer product behavior based on the ID of the customer. Product sampling and product clustering are used for analyzing the behavior of the customer. Distance clustering is done for the shops where the customer had purchased products. The prediction is done by using the classification technique.

`

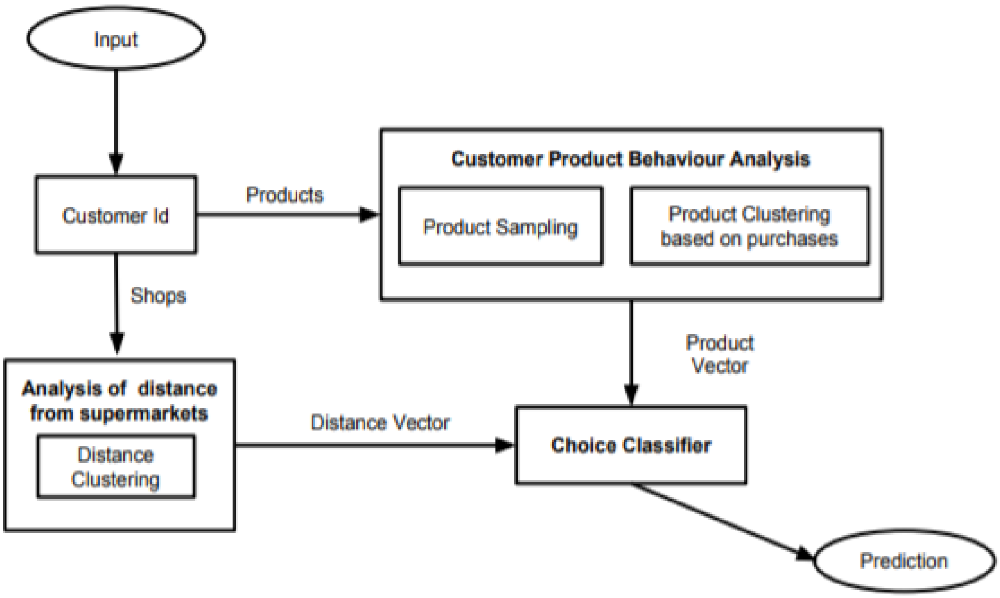


Figure 1:4: Supermarket model [6]

VII. Market Basket Analysis for a Supermarket based on Frequent Itemset Mining

In [7] K-Apriori algorithm is used to divide the customers into different segments initially. Then it finds the frequent item sets and association rules separately for those categories. K-Apriori algorithm attempts to find consumer behaviors as groups and those specific groups of people can be satisfied effectively. Related products are placed together in such a manner to increase the customer satisfaction and the customers can find items logically they to buy among the rest. The figure 1.5 shows the Apriori algorithm that counts the transaction. It uses an iterative approach known as level-wise search, in which n-item sets are used to explore (n+1) item sets.

Apriori algorithm for Frequent Itemset Mining

Cd_n : Candidate itemset of size n
L_n: frequent itemset of size n
L_1 = {frequent items};
For (n=1; L_n != ϕ; n++)
Do begin
Cd_{n+1} = candidates generated from L_n;
For each transaction T in database do
Increment the count of all candidates in Cd_{n+1} that are contained in T
 L_{n+1}= candidates in Cd_{n+1} with min_support
 End
Return $\cup_n L_n$

Figure 1:5: Apriori Algorithm [7]

VIII. Offering A Product Recommendation System in E-commerce

The paper [8] proposes a system that recommends products to a new user. Recommendation depends on the purchase behavior of previous users whose purchase behavior is closed to that of a user who asks for a recommendation. To find out the closest user profile among the profiles of all users in database, the system used weighted cosine similarity measure. The Association rule mining rule is also used in recommending products. The figure 1.6 shows the algorithm with a set of rules in 14 steps for the recommendation of products.

```
Input: D= {Pl1, Pl2,….Plk}  // Database of products
      Plr                    // The rating on the products of the User for whom the
                             recommendation is being carried out
      RatD                   // Rating of all users on the products of D
      s                      // Support
Output : Recommended products

Step 1: Use Vector Space model on D, Plr and RatD to find the users of the same
        clusters of the user for whom the recommendation is to be made.
Step 2: Find out the ratings of the products rated best for each of the 5 users for the
        products which are not present in Plr and store these into plr1 array of size
        5..
Step 3: for i=1 to 5
            Loop
Step 4: Take temp_reco= i.
Step 5: Find out whether the product has been bought after any of the products in plr
Step 6: then recommend the product and insert into reco[i]
Step 7: else go to Step 3 and execute Step 4 thru Step 7.
        End Loop
   Step 8: for i= 1 to 5
           Loop
   Step 9: Sort D on the frequency of the products in transactions
Step 10: Find all Association rule using FP-growth algorithm for the product in plr[i]
       and insert into Reclist1.
   Step 11:  Find out whether the products in Reclist1 bought after any of the products
in plr.
   Step 12:  If Step 11 is satisfied, then include that product in plr1[i].
   Step 13: Else go to Step 8.
             End Loop
   Step 14: End.
```

Figure 1:6: Recommendation Algorithm [8]

IX. Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites

In the article [9] Collaborative filtering- based recommender system is developed for e-commerce sites. The proposed approach analyses the data captured from the behavioral patterns of customers, finds the preference levels of a customer for the products, which are clicked but not purchased, and CF is conducted using the preference levels for making recommendations. The figure 1.7 shows the possible actions that can be taken by customers in e-commerce sites from the point of logging-in to the website to the point of actual purchase of a product.

`

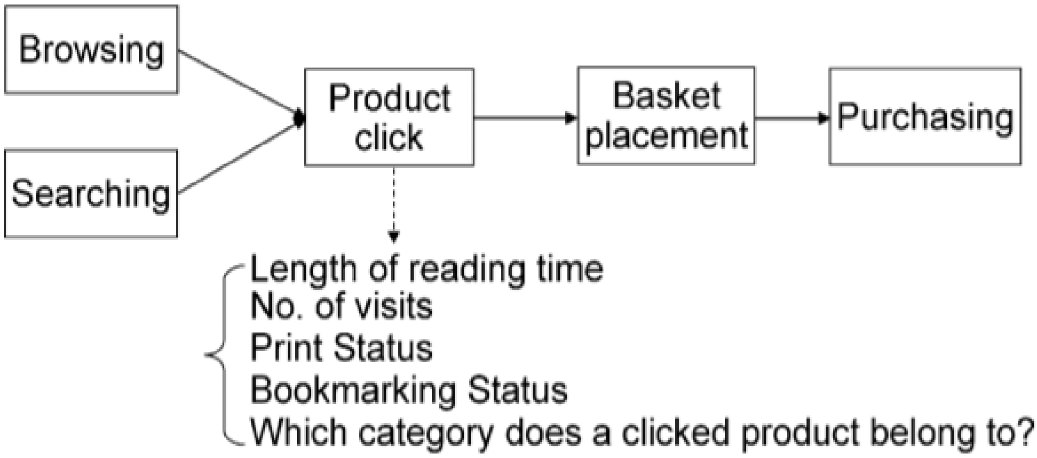


Figure 1:7: Possible actions taken by customers in e-commerce sites [9]

X. Digitalized Recommendation Engine for Supermarket Customers Using Frequency of Purchase

The paper [10] recommends products to the customers by using the purchase history of the user and the user's behavior of repetition during purchases in a supermarket. The proposed algorithm comprises of three main steps. They are calculating the difference in number of days between current date and last purchase date, calculating the mean days of purchase and determining product to be recommended. Figure 1.8 shows the steps involved in recommending products to the customers.

```
Step 1: Registration of the user in the supermarket database
Step 2: Add purchase details to the database whenever new purchase occurs
Step 3: Request user id if the user wants recommendation
Step 4: Calcualte the difference in number of days between purchase of each product by that user.
Step 5: Calculate the mean days of purchase from the above value.
Step 6: Check the below condition for each product to be recommended.
            if (mean value lies between specified range)
                    Check the item as recommended
            Else
                Do not recommend the item
      End
Step 7: Display all the recommended items in the UI
Step 8: End
```

Figure 1:8: Pseudo code of recommendation system [10]

XI. Opinionated Product Recommendation

The paper [11] recommends products by extracting the sentiment scores mined from the user- generated product reviews. Suitable cases are retrieved and ranked based on similarity and sentiment during recommendation. Sentiment analysis and feature extraction are the main techniques used to recommend products. Figure 1.9 shows how user-generated reviews are mined to create experimental product case bases for sentiment-based recommendation.

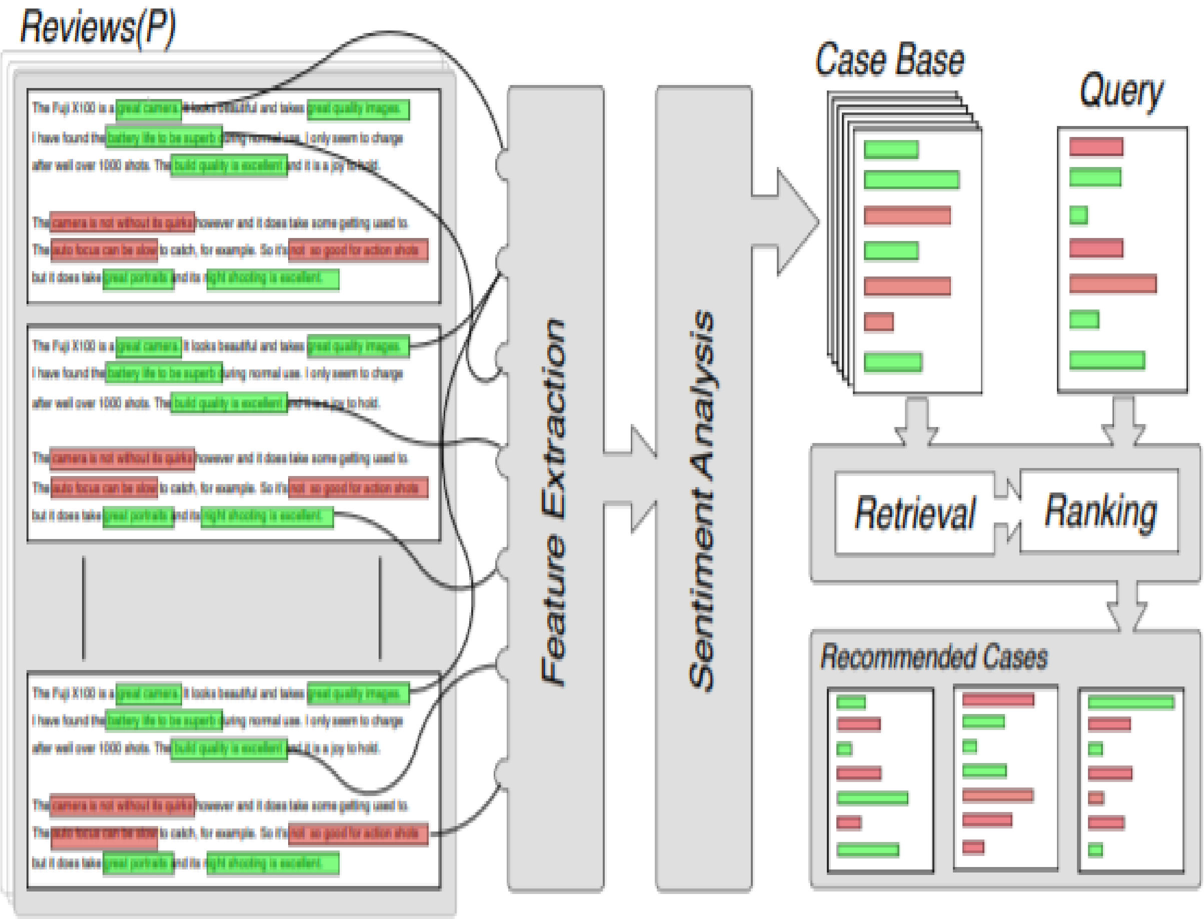


Figure 1:9: Overview of mining user-generated review [11]

XII. Product Recommendation system-based user purchase criteria and product reviews

The paper [12] recommends products to the customer based on the user purchase criteria and product reviews. When a user searches for a product to purchase, he/she selects one of the purchase criteria that is considered as top priority. The product retrieved by the user and selected purchase criterion are used as inputs. Python Web driver package to crawl the product's URL, product name, and product reviews. Figure 1.10 shows the system architecture of the product recommendation system.

`

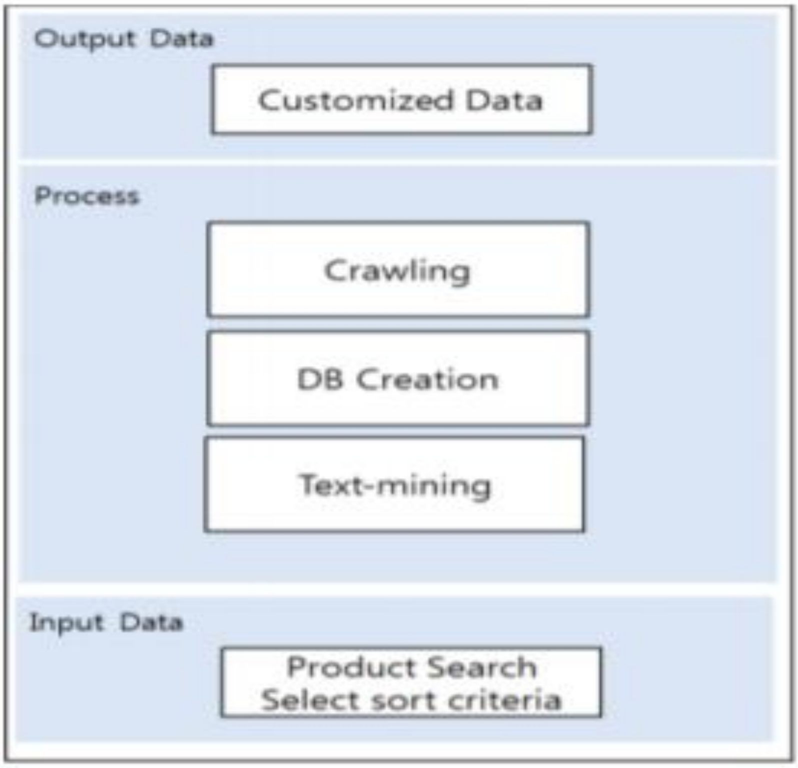


Figure 1:10: System architecture of recommendation system [12]

XIII. Products Recommendation for Mobile Devices

The study [13] proposes an algorithm called COREL (CustOmer purchase pREdiction model). The algorithm comprises of four main steps such as categorizing the products rated by the user, categorize the user’s preferences, generating product’s candidate list, calculating the probability and finally find the list of recommended products. Figure 1.11 shows the workflow of COREL.

`

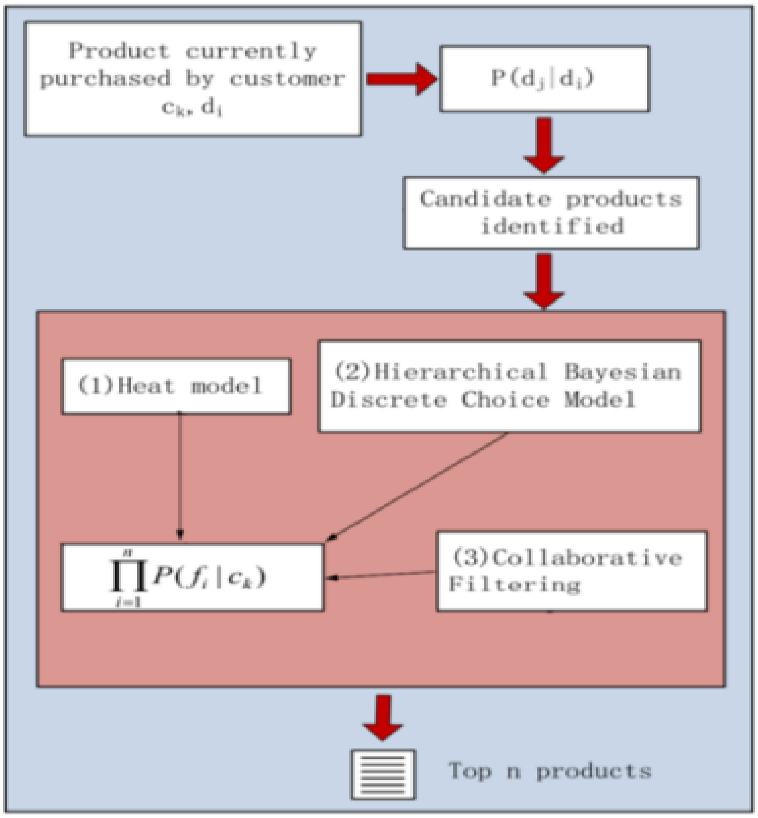


Figure 1:11: Workflow of COREL [13]

XIV. Recommendation system for grocery store considering data sparsity

The paper [14] proposes two recommendation systems. One is direct recommendation while the other is two-step recommendation of product items based on stored POS data considering the sparsity of data. Five methods such as user-based collaborative filtering, approximation by singular value decomposition, similarity by singular value decomposition, non-linear principle component analysis and combination of collaborative filtering and singular value decomposition were chosen to identify the best recommendation algorithm. As per the results, reconstructed evaluation value by SVD is appropriate for direct recommendation and user-based collaborative filtering is appropriate for two step recommendation. The figure 1.12 shows the way of using two recommendation systems.

`

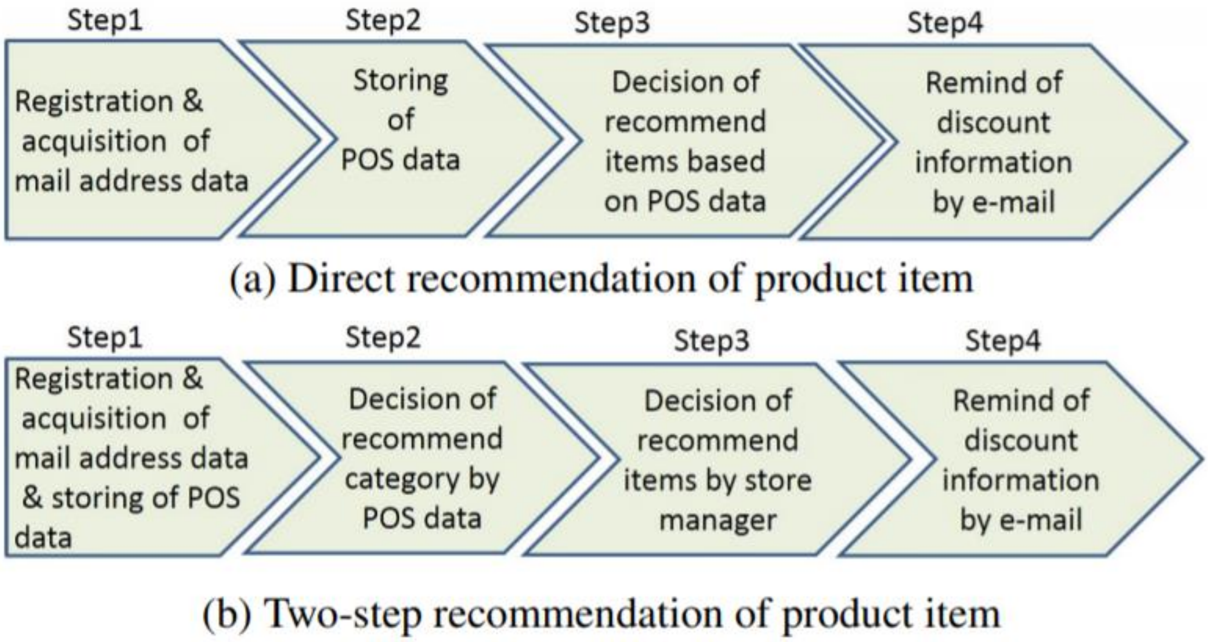


Figure 1:12: Recommendation system for grocery stores [14]

`

1.2. Research Gap

Table 1-1: Comparison with existing system

	Recommendation for new customers	Collaborative filtering	Association rule mining
A Real-Time Targeted Recommender System for Supermarkets [1]	✘	✔	✘
Enhancing Product Recommender Systems on Sparse Binary Data [2]	✘	✔	✔
Dihedral Product Recommendation System for E-commerce Using Data Mining Applications [3]	✘	✘	✔
Basket-Sensitive Personalized Item Recommendation [4]	✘	✘	✔
A Product Recommendation System using vector space model and Association Rule [5]	✔	✘	✔

`

Large Scale Product Recommendation of Supermarket Ware Based on Customer Behavior Analysis [6]	✘	✘	✓
Market Basket Analysis for a Supermarket based on Frequent Item set Mining [7]	✘	✘	✓
Offering A Product Recommendation System in Ecommerce [8]	✓	✘	✓
Development of a recommendation system based on navigational and behavioral patterns of customers in ecommerce-sites [9]	✘	✓	✘
Digitalized Recommendation Engine for Supermarket Customers Using Frequency of Purchase [10]	✘	✘	✘

`

Opinionated Product Recommendation [11]	✘	✘	✘
Product Recommendation system-based user purchase criteria and product reviews [12]	✘	✘	✘
Products Recommendation for Mobile Devices [13]	✘	✓	✘
Recommendation system for grocery store considering data sparsity [14]	✘	✓	×
This research	✓	✓	✓

`

1.3. Research Problem

It is a difficult task to find the personal preference of customers on a wide range of choices offered for a particular product. We are unable to make sure whether the customers tend to purchase accessories with their favorite main products. Customer preferences will always change beyond your imagination. Customers are subjected to difficulties at supermarkets in selecting products from a large variety of products. They spend a lot of time in finding their desired products. Majority of people didn't plan about the contents of their purchase before the inception of supermarkets. They are unable to remember the list of goods while purchasing. Such condition prevails when people attempt to buy things for a social event/official ceremony or to prepare certain cuisines. We all know that the customers purchase some particular items on daily, weekly, monthly basis in compliance to the varying up seasonal requirement. In case where a new customer enters into the supermarket, he /she finds it difficult to choose products due to the lack of purchasing history that instils confidence and recommendation which can be simply interpreted as cold start problem.

`

1.4. Research Objectives

1.4.1. Main Objectives

Key objective of the product recommendation system is to recommend personalized items to customers to eliminate inconveniences arising from delay and confusion in searching and make customers feel more comfortable while purchasing.

1.4.2. Specific Objectives

In order to realize the key objective, the following set of specific objectives should be achieved.

- ✓ Implementing the first level of recommendation in prior to choose a product by using the clustering method.
- ✓ Implementing the second level of recommendation in consequent with the choosing of product by using association rule mining.
- ✓ Recommending the products to the new customers.

`

2. Methodology

The methodology explains how the project was planned and type of methodology used in order to deliver a proper product. It discourses description of methodology, planning, requirement gathering, and analysis are used in this project.

I. Cosine similarity

It is the measure of similarity between two vectors by calculating the cosine of the angle between them. It can be applied to items available on a dataset to compute similarity to one another via keywords or other metrics [15]. The algorithm generates recommendations based on customers who are very similar to the user. It can measure the similarity of two customers, A and B respectively by measuring the cosine of the angle between the two vectors.

$$similarity(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|}$$

Equation 1.4-1: Cosine Similarity Equation

II. Association Rule Mining

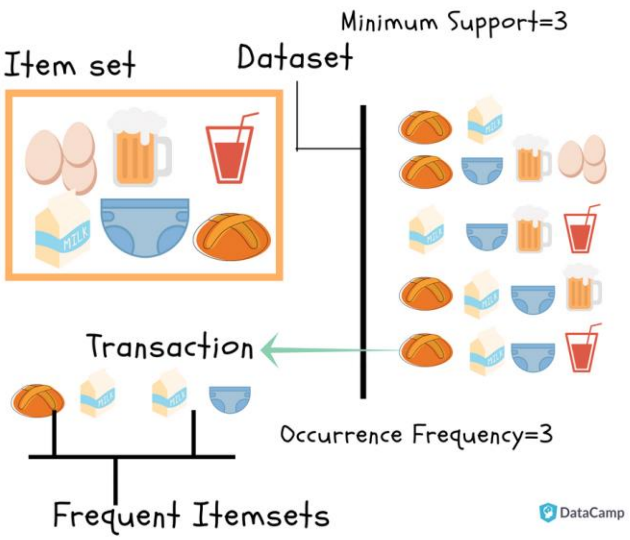


Figure 2:1Association rule mining

`

Association rule is mainly used to identify ‘interesting’ hidden relationships among attributes of huge data set. It finds features which occur together and the features which are correlated [16]. Association rules mining finds repeated items in a set of transactions as frequent patterns. In this case, Apriori algorithm is used to discover relationship among products. Apriori algorithm has been used to find out strong association rules among item sets. The algorithm uses a “bottom-up” approach, where frequent subsets are extended one item at once (candidate generation) and groups of candidates are tested against the data [17]. The algorithm terminates when no further successful rules can be derived from the data. There are measures related to association rule mining. They are termed as support confidence and lift as well.

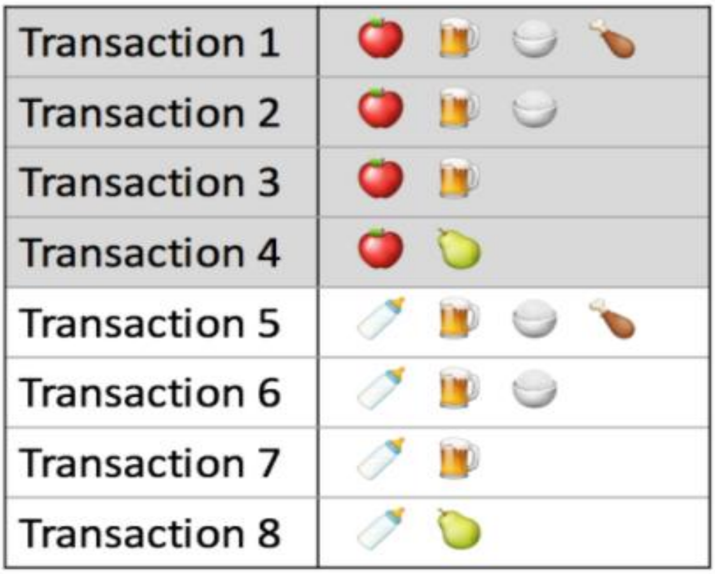


Figure 2:2:Transactions

Support- The number of transactions that include items in the {X} and {Y} parts of the rule as a percentage of the total number of transactions. It is a measure of how frequently the collection of items occur together as a percentage of all transactions [18].

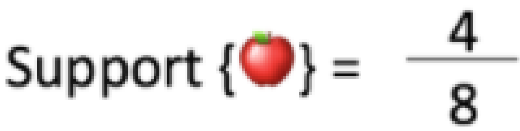


Figure 2:3: Calculating support

`

Confidence- It is the ratio of the no of transactions that includes all items in {B} as well as the no of transactions that includes all items in {A} to the no of transactions that includes all items in {A} [18].

$$\text{Confidence}\ \{🍎 \rightarrow 🍺\} = \frac{\text{Support}\ \{🍎, 🍺\}}{\text{Support}\ \{🍎\}}$$

Figure 2:4: Calculating confidence

Lift- The lift of the rule X=>Y is the confidence of the rule divided by the expected confidence, assuming that the item sets X and Y are independent of each other. The expected confidence is the confidence divided by the frequency of {Y} [18].

$$\text{Lift}\ \{🍎 \rightarrow 🍺\} = \frac{\text{Support}\ \{🍎, 🍺\}}{\text{Support}\ \{🍎\} \times \text{Support}\ \{🍺\}}$$

Figure 2:5: Calculating lift:

III. Collaborative filtering

This filtering method is usually based on collecting and analyzing information on user's behaviors, their activities or preferences and predicting what they will like based on the similarity with other users [19].

IV. Item-Based Collaborative filtering

Item-based collaborative filtering is one kind of recommendation method, which looks for similar items based on the item users have already either preferred positively interacted [20].

`

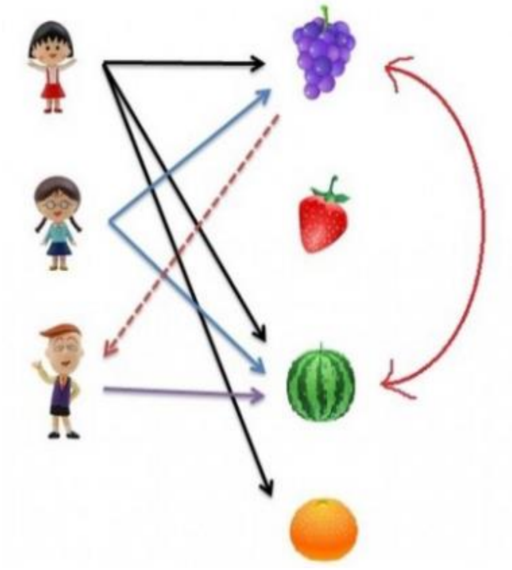


Figure 2:6: Item-based collaborative filtering

V. User-based Collaborative filtering

User-based collaborative filtering is another kind of recommendation method, which looks for similar users based on the item users have already liked or positively interacted with.

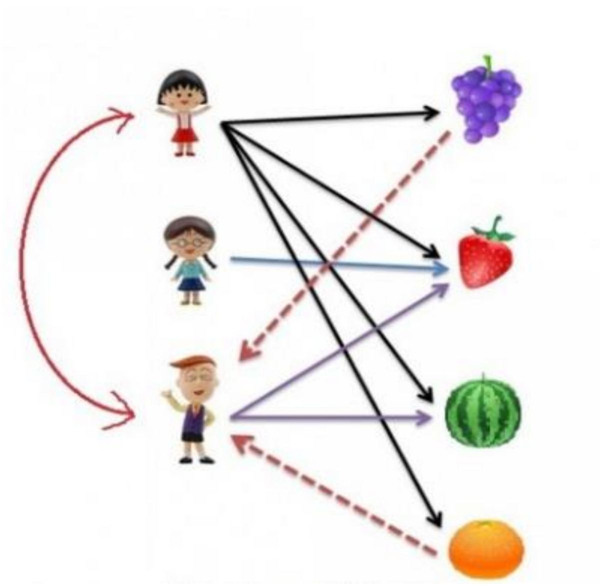


Figure 2:7: User-based collaborative filtering

`

VI. Apriori Algorithm

Apriori algorithm is a classical algorithm in data mining. It is used for mining frequent item sets and relevant association rules [21]. It assumes all subsets of a frequent item sets must be frequent and for any infrequent itemset, all its supersets must be infrequent too. The most prominent practical application of the algorithm is to recommend products based on the products already present in the user's cart [22]. Easy for implementation, easy for understanding and capable on use of large item sets [21] are the advantages of using Apriori algorithm.

Frequent Item set- A set of items is called frequent if it satisfies a minimum threshold value for support and confidence [23].

`

2.1. Methodology

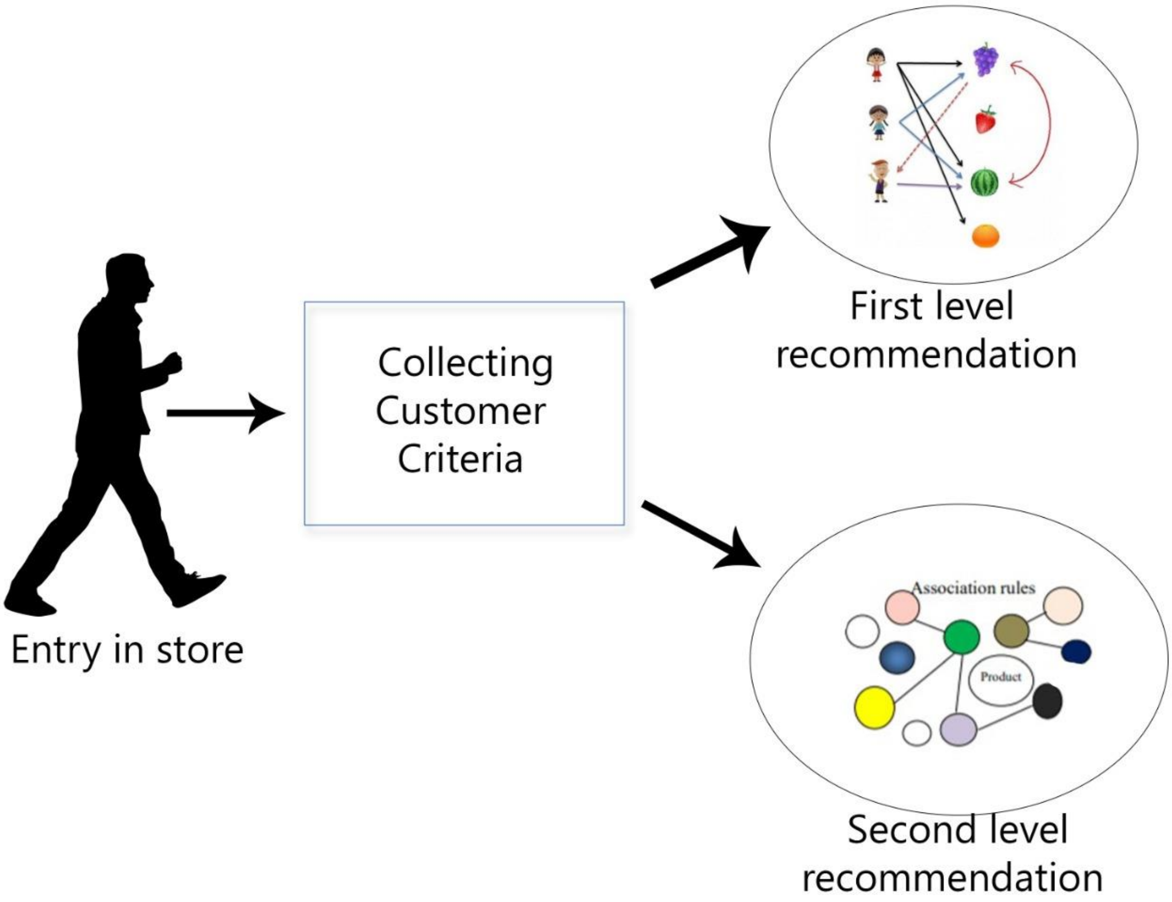


Figure 2:8 :System Diagram

The figure 2.8 shows the system diagram of the product recommendation system. The system includes the product recommendation at two stages. In the first stage, recommendation is done in prior to the purchasing of product. In the second stage, recommendation is done in consequent with the purchasing of product. Firstly, data on previous purchases of products are collected. Initially, product recommendation system recommends the products that have closely related with the criteria of customers.

User-based systems use the customer profile data. They can incorporate demographics data along with the historical purchasing data to identify the neighborhood for the active customer. Products are then recommended according to their similarities to this neighborhood. Recommendations are based on calculating the similarities of two users. User-to-user matrix is built by iterating through all user pairs and computing similarity matrix for each pair. Similarity

`

between two items is measured using up cosine similarity. Similar products at the top are recommended based on cosine similarity. Giving recommendation to User B is depending on the buying pattern of User A.

In contrast, the item-based method uses only the purchasing data history to identify similarities between different items. In Item-based method, a table with similar items is built by finding the items on which the customers have a tendency to buy together. Therefore, product-to-product matrix is built by iterating through all item pairs and computing similar matrix for each pair. Similarity between two items is measured using up cosine similarity. The algorithm finds similarity in each purchase and the ratings of the user, aggregates the items purchased and recommend the most popular items.

In the second stage, it recommends the purchasing of associated products with the desired product of customers to complete the buying process and to make customers aware of potentially related products with their desired products. Information about the history of shopping behavior includes the products that are purchased very often with other products. As a result, the relationship between products can be explored in terms of a data mining application called association rule mining. Using these rules, we can find buying patterns. The relationship between the products will steadily increase the likelihood for buying the desired products with associated products.

New customers mean obviously the customers with no purchasing history. The system recommends products in two ways. In one method, the most popular products are recommended to the new customers. Most popular products are identified through ratings given by the regular customers of the supermarket. In the other method, products are recommended based on product description. K-means clustering is used to find top words in each cluster on the basis of product description. In case where a word appears in multiple clusters, the algorithm chooses the cluster with the highest frequency of occurrences of the word. The recommendation system display items from the corresponding product clusters based on the product descriptions.

`

2.1.1. Planning

Planning phase is the first phase of the Software Development Life Cycle (SDLC) which is the most crucial phase that led to success of project. Gantt chart and Work Breakdown Structure (WBS) is designed to get a good idea about the product recommendation system. Gantt chart is designed to schedule the product recommendation system according to the deadlines. Work break down structure which helps to break the larger task into smaller tasks to perform in an effective manner.

2.1.2. Requirement Gathering and Analysis

Requirement gathering and analysis phase is the second stage of the SDLC, which is also an important phase for the product recommendation system. There are two ways requirement gathering namely primary data gathering and secondary data gathering. Requirements are gathered as secondary data using resources such as research papers, journal articles, conference papers and reliable websites. Requirement analysis is done in order to find out what kind of product recommendation systems are used already, and what kind of technology and methodology are implemented.

2.1.3. Design

UML diagrams help to reduce the complexity of system. System is designed based on the requirements and analysis. Design phase is the most important phase in the development of system. Programming language, hardware and software platform are decided in this stage. As far as the development process of the product recommendation system is concerned, it helps the developer to carry on with the system in a smooth way. Design models will give a proper idea on how the functions work in the product recommendation system during the implementation phase in the SDLC. Rarely, errors may occur during the implementation and it will guide and keep track on mistakes.

`

2.1.4. Use Case Diagram

The following Use- case diagram explains the functionalities of the system and the user roles within the system. There are two actors including customer and system.

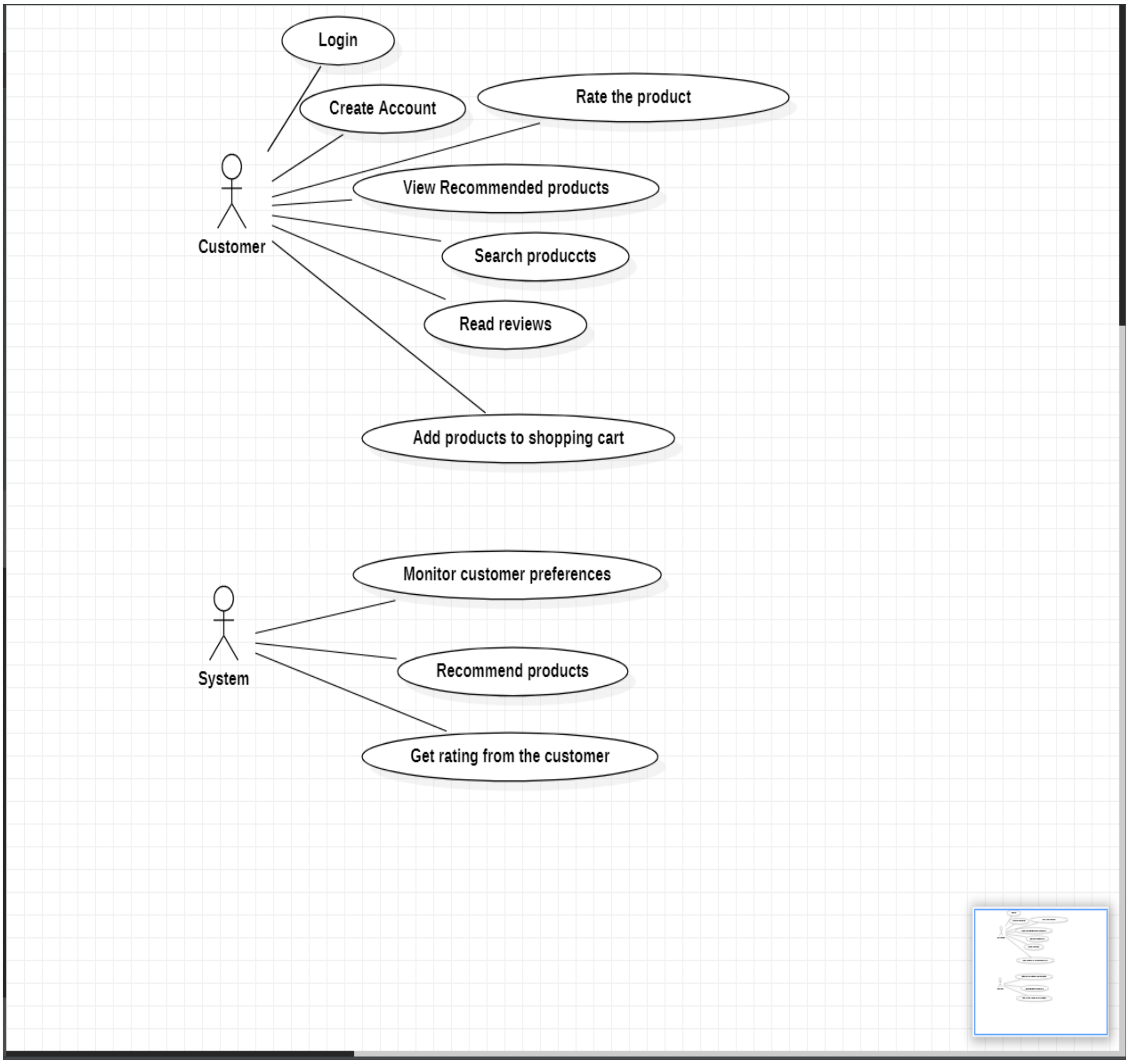


Figure 2:9: Use-case diagram of product recommendation system

`

2.2. Commercialization aspect of the product

Product Recommendation system would be efficient at supermarkets as it acts as a filtering system to find the most relevant items for a particular user as per his/her preference. Product recommendation systems are beneficial to both the users and service providers. Majority of customers would prefer for recommendation system as it give them the access to get to know about the feedback given by other customers. This feedback helps to increase the purchases. Product recommendation system improves the decision-making process. Product recommendation system makes the search easy to the customers on products through furnishing the experience of users. It keeps the customer happy with the services. Improved experience of the user increases the sales as the customers tend to make purchases even if they are not sure about the contents of their purchase. Good product recommendation system increases the retention rate of customers. This system helps to retain the existing customers and attract the new customers as well. There is no need for a shop keeper as the product recommendation system carry out the task of recommendation of similar products whenever a customer requires to choose a product from a wide range of brands. Therefore, the product recommendation system reduces the work load of a shop keeper. This system is designed in a way to improve the shopping behavior of customers in supermarkets.

`

2.3. Testing and Implementation

2.3.1. Testing

Testing is a mandatory requirement in a system, to assess its function in the pre-determined way. Testing was carried out on the system to ensure the perfection of functional and non-functional elements for its compliance to the requirement of traders and users. Arrangements to conduct testing will help to detect impending potential faults that would occur during the development phase in order to ensure the proper function of product recommendation system and its component. The description of tests that were conducted on the system are given below.

- **Unit testing-** Each unit in the system was tested separately to develop the error-free model.

- **Component testing-** Model was tested on several instances in view for its accuracy by using different datasets and different methodologies.

- **Integration testing-** All the accurate models were integrated and tested whether the system works properly.

- **System testing-** This testing was done to check the integrated system works properly to verify that it meets its requirements and functionality of the system.

`

Table 2-1: Test cases of product recommendation system

Description	Expected Results	Actual Results	Status
Results based on Item-based Collaborative Filtering	Recommending highly co-related items for the item selected by the customer	As expected	Pass
Results based on User-based Collaborative Filtering	Finding the most similar customer for the existing customer	As expected	Pass
Results based on k-means Clustering	Chooses the cluster for the product description with the highest frequency of occurrences of the word	As expected	Pass
Results based on Association Rule Mining	Recommending best product based on the generated association rules.	As expected	Pass

`

2.3.2. Implementation

In consequent with the designing of each iteration, action was taken to check its veracity in terms of practically for its compliance to the methodology. Some technologies and languages were used to achieve the goals and objectives of product recommendation system while in implementation of the product recommendation system are given below.

- ✓ Python

  It is an interpreted, high-level and general-purpose programming language, which was created by Guido van Rossum. It has a simple syntax similar to the English language. It supports multiple programming paradigms such as structured, object-oriented and functional programming. It helps programmers write clear, logical code for small and large-scale projects.

- ✓ Jupyter Notebook

  The Jupyter notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. It is used for data cleaning and transformation, numerical simulation, statistical modeling, data visualizations, machine learning etc.

- ✓ Flask

  Flask is micro web framework written in python. It classified as a micro framework as it does not require particular libraries or tools.

- ✓ Ionic Framework

  It is an open source mobile UI toolkit which is used for building high quality, cross-platform native and web app experiences.

`

2.3.2.1. Code Implementation

The code segment in the figure 2.10 below shows the python libraries needed for the implementation. Pandas is a python package that helps in data manipulation tasks and data analysis. Matplotlib is a comprehensive library used to create interactive visualizations. NumPy is a library which helps to add large, multi-dimensional arrays and matrices. Seaborn, which is based on matplotlib, that provides high-level interface for creating attractive statistical graphics. Cosine similarity is used to find cosine between two vectors.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.metrics.pairwise import cosine_similarity
```

Figure 2:10: Code segment of importing libraries

The code segment in the figure 2.11 below shows how products recommended to the new customers. Most popular products were recommended to the new customers. Most popular products were identified by the ratings provided by the customers. The ratings were sorted in the descending order to find the products with the highest rating. Here we can choose the number of products recommended to the customers according to the need. Here this code takes top ten products with the highest rating.

```
most_popular = popular_products.sort_values('Rating', ascending=False)
most_popular.head(10)
```

Figure 2:11: Code segment of finding popular products

`

The code segment in the figure 2.12 below shows that the unwanted columns were dropped. Columns such as ‘InvoiceDate’, ‘UnitPrice’, ‘Gender’, ‘Payment_Type’, ‘Quantity’, ‘Description’ were dropped from the dataset.

```
df.drop(['InvoiceDate', 'UnitPrice', 'Gender', 'Payment_Type', 'Quantity', 'Description'], inplace=True, axis=1)
```

Figure 2:12:Code segment of dropping unwanted columns

The code segment 2.13 below shows the creation of customer-product matrix. It was created by considering the ‘Customer_ID’ as index of the matrix, ‘Product_ID’ as column of the matrix, Rating as the values of the matrix.

```
customer_product_matrix = df.pivot_table(index='Customer_ID',columns='Product_ID',values='Rating',aggfunc='sum')
```

Figure 2:13: Code segment of building cutomer_product matrix

The code segment in the figure 2.14 below shows that to find the item-item similarity first cosine similarity was applied to the transpose of customer-product matrix.

```
item_item_similarity_matrix = pd.DataFrame(cosine_similarity(customer_product_matrix.T))
```

Figure 2:14:Code segment of building item_item_similarity matrix

`

The code segment in the figure 2.15 below shows that index was set as 'Product_ID' to the transpose of the customer-product matrix.

```
item_item_similarity_matrix.columns = customer_product_matrix.T.index
item_item_similarity_matrix['Product_ID'] = customer_product_matrix.T.index
item_item_similarity_matrix = item_item_similarity_matrix.set_index('Product_ID')
```

Figure 2:15: Code segment of setting Product_ID as index

The code segment in the figure 2.16 below shows how top ten similar items were found. If a customer chooses, the product with 'Product_ID- 21873' the system finds the similar items by using the item-item similarity matrix.

```
top_10_similar_items = list(
    item_item_similarity_matrix\
        .loc[21873]\
        .sort_values(ascending=False)\
        .iloc[:10]\
    .index
)
```

Figure 2:16: Code segment of finding top similar items

The code segment in the figure 2.17 below how to find retrieve the products bought by a customer from the customer-product matrix. Customer_ID is passed to find the products purchased by the customer.

```
items_bought_by_A = set(customer_product_matrix.loc[1113].iloc[customer_product_matrix.loc[1113].nonzero()].index)
```

Figure 2:17: Code segment of finding the items bought by A

`

The code segment in the figure 2.18 below shows that the dataset is clustered into ten groups.

```
kmeans = KMeans(n_clusters = 10, init = 'k-means++')
y_kmeans = kmeans.fit_predict(X)
```

Figure 2:18: Code segment of clustering using k-means

The code segment in the figure 2.19 below shows that the products are clustered into 5 groups using k-means and top five products were selected from each group.

```
true_k = 5

model = KMeans(n_clusters=true_k, init='k-means++', max_iter=100, n_init=1)
model.fit(X1)

print("Top products per cluster:")
order_centroids = model.cluster_centers_.argsort()[:, ::-1]
terms = vectorizer.get_feature_names()
for i in range(true_k):
    print_cluster(i)
```

Figure 2:19:Code segment of finding top products per cluster

`

The code segment in the figure 2.20 below shows that the dataset is converted into list in-order to apply the Apriori algorithm.

```
records = []
for i in range(1, 7501):
    records.append([str(Basket.values[i,j]) for j in range(0, 20)])
```

Figure 2:20:Code segment of converting records as list

The code segment in the figure 2.21 below shows that Apriori algorithm is used with passing minimum support as 0.0045, minimum confidence as 0.2, minimum lift as 3 and minimum length as 2.

```
association_rules = apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=2)
association_rules = list(association_rules)
```

Figure 2:21:Code segment of Apriori algorithm

`

Backend

The figure 2.22 shows the code segment that is used to get the details of all products from database.

```
#     else:
#         return jsonify({'faild'})
@app.route('/getAll', methods=['GET'])
def get_product():
    user = mongo.db.product
    result = []
    response = user.find().limit(400)
    if response.count() > 0:
        for fields in response:
            result.append({'Product_ID': fields['Product_ID'], 'Description': fields['Description'],
                           'InvoiceDate': fields['InvoiceDate'],
                           'UnitPrice': fields['UnitPrice'], 'Gender': fields['Gender'],
                           'Payment_Type': fields['Payment_Type'],
                           'Customer_ID': fields['Customer_ID'], 'Quantity': fields['Quantity'],
                           'Rating': fields['Rating'],
                           'Product_Name': fields['Product_Name']})
        return jsonify(result)
```

Figure 2:22: Get All products

`

The figure 2.23 and figure 2.24 shows the code segment that is used to integrate the recommendation model.

```
app.py    requirements.txt    Procfile

@app.route('/recommend', methods=['POST'])
def recommendation():
    _json = request.get_json()
    product = mongo.db.product
    product_id = _json['product_id']
    # ML PART
    df = pd.read_csv("Data.csv")
    df.drop(['InvoiceDate', 'UnitPrice', 'Gender', 'Payment_Type', 'Quantity', 'Description'], inplace=True, axis=1)
    df = df.dropna()
    customer_product_matrix = df.pivot_table(index='Customer_ID', columns='Product_ID', values='Rating', aggfunc='sum')
    customer_product_matrix = customer_product_matrix.applymap(lambda x: 1 if x > 0 else 0)
    item_item_similarity_matrix = pd.DataFrame(cosine_similarity(customer_product_matrix.T))
    item_item_similarity_matrix.columns = customer_product_matrix.T.index
    item_item_similarity_matrix['Product_ID'] = customer_product_matrix.T.index
    item_item_similarity_matrix = item_item_similarity_matrix.set_index('Product_ID')
    top_10_similar_items = list(
        item_item_similarity_matrix.loc[product_id].sort_values(ascending=False).iloc[:10].index)
    products = df.loc[
        df['Product_ID'].isin(top_10_similar_items),
        ['Product_Name']
    ].drop_duplicates().set_index('Product_Name')
    result = []
    for i in top_10_similar_items:
        fields = product.find_one_or_404({'Product_ID': i})
        result.append({'Product_ID': fields['Product_ID'], 'Description': fields['Description'],
                       'InvoiceDate': fields['InvoiceDate'],
                       'UnitPrice': fields['UnitPrice'], 'Gender': fields['Gender'],
                       'Payment_Type': fields['Payment_Type'],
                       'Customer_ID': fields['Customer_ID'], 'Quantity': fields['Quantity'],
```

Figure 2:23: Recommendation system model Integration 1

`

```python
    customer_product_matrix = df.pivot_table(index='Customer_ID', columns='Product_ID', values='Rating', aggfunc='sum')
    customer_product_matrix = customer_product_matrix.applymap(lambda x: 1 if x > 0 else 0)
    item_item_similarity_matrix = pd.DataFrame(cosine_similarity(customer_product_matrix.T))
    item_item_similarity_matrix.columns = customer_product_matrix.T.index
    item_item_similarity_matrix['Product_ID'] = customer_product_matrix.T.index
    item_item_similarity_matrix = item_item_similarity_matrix.set_index('Product_ID')
    top_10_similar_items = list(
        item_item_similarity_matrix.loc[product_id].sort_values(ascending=False).iloc[:10].index)
    products = df.loc[
        df['Product_ID'].isin(top_10_similar_items),
        ['Product_Name']
    ].drop_duplicates().set_index('Product_Name')
    result = []
    for i in top_10_similar_items:
        fields = product.find_one_or_404({'Product_ID': i})
        result.append({'Product_ID': fields['Product_ID'], 'Description': fields['Description'],
                       'InvoiceDate': fields['InvoiceDate'],
                       'UnitPrice': fields['UnitPrice'], 'Gender': fields['Gender'],
                       'Payment_Type': fields['Payment_Type'],
                       'Customer_ID': fields['Customer_ID'], 'Quantity': fields['Quantity'],
                       'Rating': fields['Rating'],
                       'Product_Name': fields['Product_Name']})
    return jsonify(result)
```

Figure 2:24: Recommendation system model Integration 2

Frontend

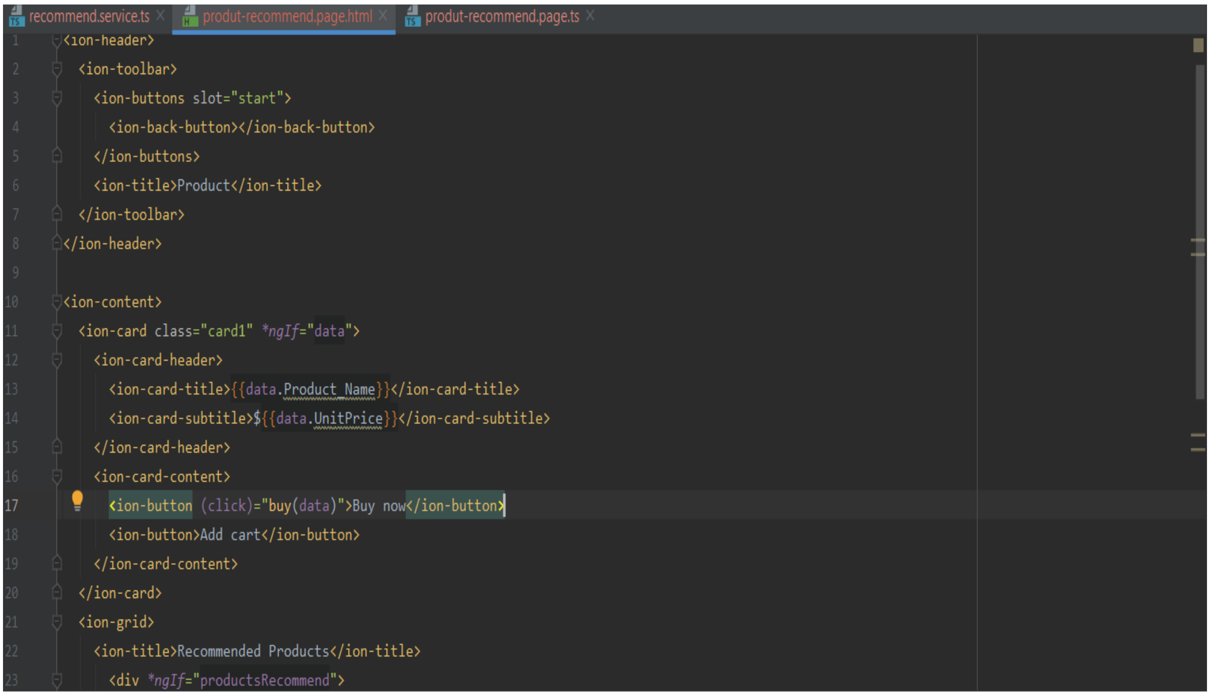
The figure 2.25 shows the code segment that is used to get the detail of products and get the recommended products.

```typescript
import ...
@Injectable({
    providedIn: 'root'
})
export class RecommendService {
    URL = environment.url;

    constructor(private http: HttpClient, private router: Router) {
    }

    getAll() {
        return this.http.get( url: this.URL + '/getAll');
    }

  recommendedSystem(id) {
        const body = {
            product_id: id
        };
        return this.http.post( url: this.URL + '/recommend', body);
    }

}
```

Figure 2:25: service.ts

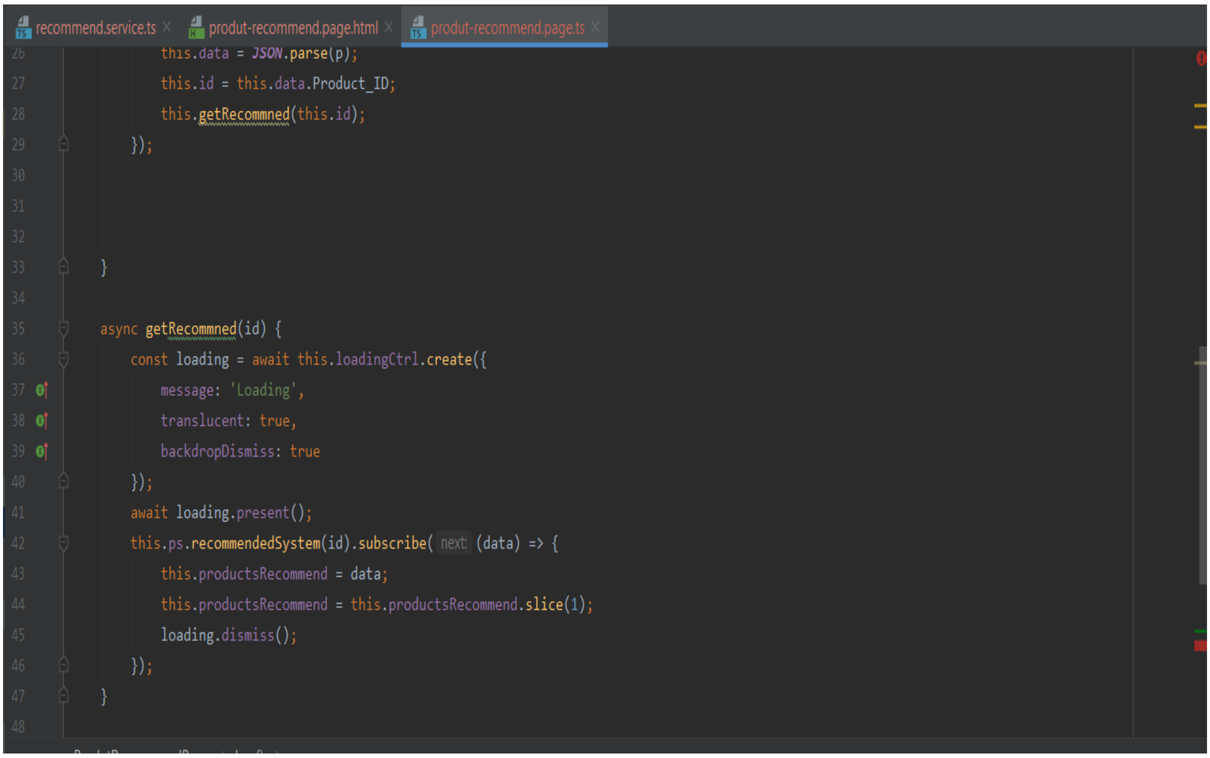
`

The figure 2.26 and figure 2.27 shows the recommended products in the frontend.



```
recommend.service.ts   produt-recommend.page.html   produt-recommend.page.ts
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="start">
      <ion-back-button></ion-back-button>
    </ion-buttons>
    <ion-title>Product</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-card class="card1" *ngIf="data">
    <ion-card-header>
      <ion-card-title>{{data.Product_Name}}</ion-card-title>
      <ion-card-subtitle>${{data.UnitPrice}}</ion-card-subtitle>
    </ion-card-header>
    <ion-card-content>
      <ion-button (click)="buy(data)">Buy now</ion-button>
      <ion-button>Add cart</ion-button>
    </ion-card-content>
  </ion-card>
  <ion-grid>
    <ion-title>Recommended Products</ion-title>
      <div *ngIf="productsRecommend">
```

Figure 2:26:Recommended.html



```
recommend.service.ts   produt-recommend.page.html   produt-recommend.page.ts
            this.data = JSON.parse(p);
            this.id = this.data.Product_ID;
            this.getRecommned(this.id);
        });



    }

    async getRecommned(id) {
        const loading = await this.loadingCtrl.create({
            message: 'Loading',
            translucent: true,
            backdropDismiss: true
        });
        await loading.present();
        this.ps.recommendedSystem(id).subscribe( next: (data) => {
            this.productsRecommend = data;
            this.productsRecommend = this.productsRecommend.slice(1);
            loading.dismiss();
        });
    }
```

Figure 2:27: Recommended.ts

`

3. Results & Discussion

3.1.Results

A well-developed recommendation system improves the shopping experience of the customers. Whenever a new customer without any purchasing experience visits a supermarket for the first time, the product recommendation system recommends the most popular products. Most popular products are suggested by the ratings given by the regular customers of the supermarket. The following figure 3.1 shows the pattern of recommendation of top 10 popular products to new customers.

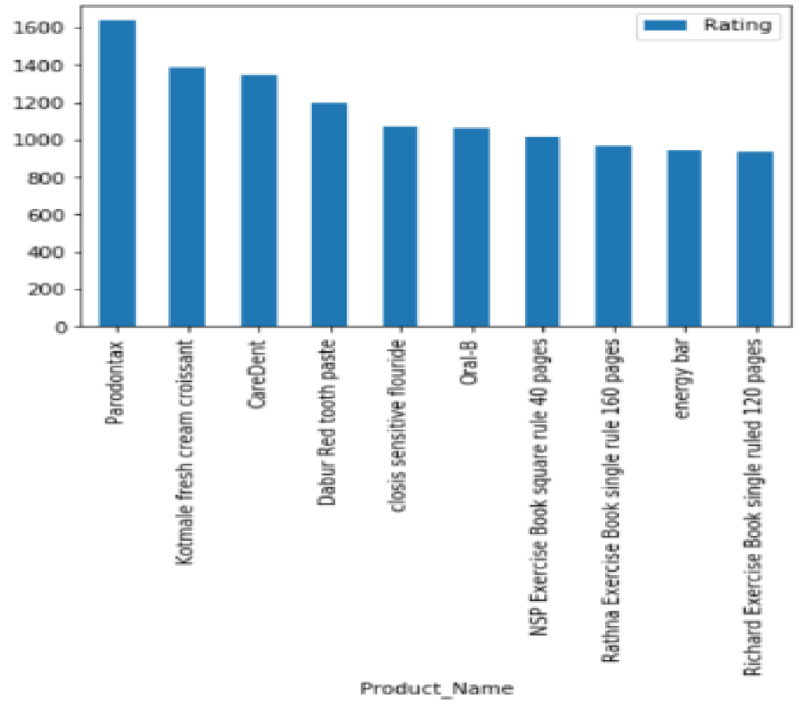


Figure 3:1: Top 10 products per each cluster

In case where a new customer searches for some specific product, the product recommendation system recommends products based on textual clustering analysis given in the product description. K-means clustering is used to find top words in each cluster based on product description. If a word appears in multiple clusters, the algorithm chooses the cluster with the highest frequency of occurrences of the word. The system display items from the corresponding product clusters based on product descriptions. Figure 3.2 shows that the products are clustered into 5 groups. Each cluster contains the top 5 products.

`

```
Top products per cluster:
Cluster 0:
 Elephant house icecream strawberry 500ml
 Highland processed cheese
 Snack cracker
 body spray
 Promate Exercise Book single rule 280 pages
Cluster 1:
 NSP Exercise Book single rule 360 pages
 SchoolMate Exercise Book single rule 240 pages
 Promate Exercise Book single rule 280 pages
 spaghetti
 SchoolMate Exercise Book square rule 120 pages
Cluster 2:
 Kotmale fresh cream croissant
 Graph Book 160 pages
 Jason sea fresh
 Rathna Exercise Book single rule 400 pages
 Highland processed cheese
Cluster 3:
 Richard Exercise Book single ruled 120 pages
 Highland Ghee
 Highland Ghee
 Elephant house icecream faluda 400ml
 Dilmah Premius Ceylon Tea 500g
Cluster 4:
 Dilmah Premius Ceylon Tea 2000g
 Dr Pepper tin soda
 Atlas Exercise Book single ruled 320 pages
 NSP Exercise Book single rule 280 pages
 Ginger soda 1250ml
```

Figure 3:2: Top products per each cluster

`

```
show_recommendations("blue bowl")
```

```
Cluster 0:
 Elephant house icecream strawberry
 Highland processed cheese
 Snack cracker
 body spray
 Promate Exercise Book single rule
```

Figure 3:3: Recommendation using clusters

Figure 3.3 shows that in case if a customer searches “blue bowl” it first identifies the best cluster and here it is found as cluster 0. Then, the system displays the products, which is clustered into cluster 0.

The system recommends products to the regular customers using up two techniques such as user-based collaborative filtering and item-based collaborative filtering. User-item rating matrix is built at first in both techniques. Figure 3.4 shows the user-item matrix, which is a 5*5 matrix. However, this matrix is built for all users along with all products in the dataset.

Product_ID	10002	10080	10120	10125	10133
Customer_ID					
1069	0	0	0	0	0
1113	0	0	0	0	0
1823	0	0	0	0	0
2189	0	0	0	0	0
3667	0	0	0	0	0

Figure 3:4: User-Item Matrix

`

Similarity between two users is calculated in user-based technique. Cosine similarity is used to calculate the similarity between two users. The algorithm builds a similar-user table to determine the most-similar match for a given item. User-to-User matrix is built by iterating through all user pairs and computing a similar matrix for each pair. Figure 3.5 shows the user-user matrix, which is a 5*5 matrix. However, this matrix is built for all users along with all customers in the dataset.

Customer_ID	1069	1113	1823	2189	3667
Customer_ID					
1069	1.0	0.0	0.0	0.0	0.0
1113	0.0	1.0	0.0	0.0	0.0
1823	0.0	0.0	1.0	0.0	0.0
2189	0.0	0.0	0.0	1.0	0.0
3667	0.0	0.0	0.0	0.0	1.0

Figure 3:5: User-User Matrix

Recommendation to user ‘B’ depends on buying pattern of customer ‘A’. Recommendation is based on the following formula.

Items recommend to B= Items bought by A - Items bought by B

Equation 3.1-1: Recommendation Formula

Similarity between two items is calculated in item-based technique. Cosine similarity is used to calculate the similarity between two items. The algorithm builds a similar-item table to determine the most-similar match for a given item. Item-to-item matrix is built by iterating through all user pairs and computing a similar matrix for each pair. Figure 3.6 shows the item-item matrix, which is a 5*5 matrix. However, this matrix is built for all users along with all products in the dataset.

`

Product_ID	10002	10080	10120	10125	10133
Product_ID					
10002	1.00000	0.0	0.000000	0.000000	0.008360
10080	0.00000	1.0	0.000000	0.000000	0.000000
10120	0.00000	0.0	1.000000	0.018831	0.013041
10125	0.00000	0.0	0.018831	1.000000	0.014735
10133	0.00836	0.0	0.013041	0.014735	1.000000

Figure 3:6: Item-Item Matrix:

With the use of similar-items table, the algorithm finds items similar to each purchase and rating of users, aggregates those items, and then recommends the most similar items. This computation is very quick, depending only on number of items purchased or rated by the user.

Product_Name
Almonds
whole wheat rice
Signal
Blueberry jelly
captain fish
closis sensitive flouride
gillette vector
Listerine Essential care
Milk Shorties
Elephant house icecream berry 450ml

Figure 3:7: Recommended List

`

Association rule mining identify hidden relationships among items. Usually the customers are following a pattern for their purchase. There are different algorithms that can be used to implement association rule. Apriori algorithm is used in this instance.

```
association_rules = apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=2)
association_rules = list(association_rules)
```

Figure 3:8: Applying Apriori

Here we assume that we are supposed to generate rules only to those items that are purchased 5 times a day. Therefore, for a week 7*5=35 times. As the dataset is for one-week time period and it contains 7500 records. Therefore, Support is calculated as 35/7500. The minimum confidence is 0.2. We specify that lift is 3 and min_length as 2 as the rule should contain at least two products. The output can be displayed as follows in figure 3.9.

```
Rule: chicken -> light cream
Support: 0.004533333333333334
Confidence: 0.2905982905982906
Lift: 4.843304843304844
=====================================
Rule: mushroom cream sauce -> escalope
Support: 0.005733333333333333
Confidence: 0.30069930069930073
Lift: 3.7903273197390845
=====================================
Rule: pasta -> escalope
Support: 0.005866666666666667
Confidence: 0.37288135593220345
Lift: 4.700185158809287
=====================================
Rule: ground beef -> herb & pepper
Support: 0.016
Confidence: 0.3234501347708895
Lift: 3.2915549671393096
```

Figure 3:9: Generated rules

`

The below figure 3.10 shows the mobile application interface which was designed using Ionic designed as the main interface for the product recommendation system which display the list of products.

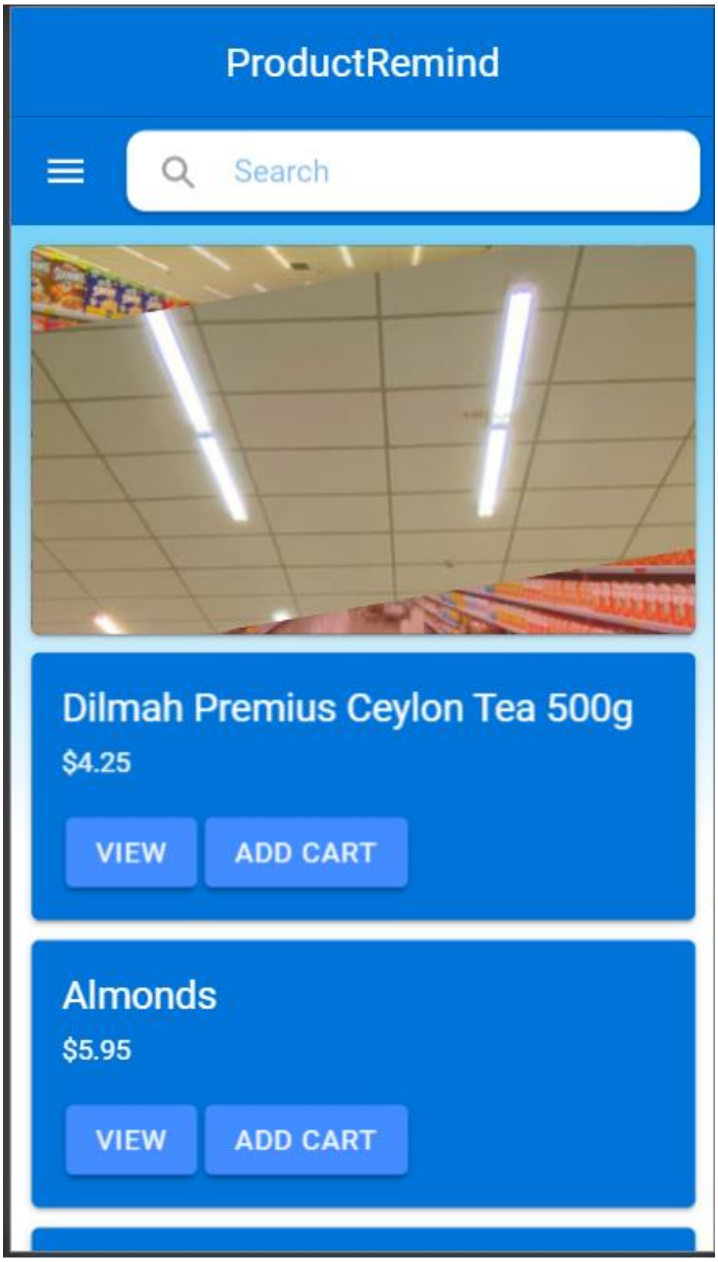


Figure 3:10 List of Products Interface

`

The below figure 3.11 shows the mobile application interface which was designed using Ionic designed as the interface for the product recommendation system which display the recommended products.

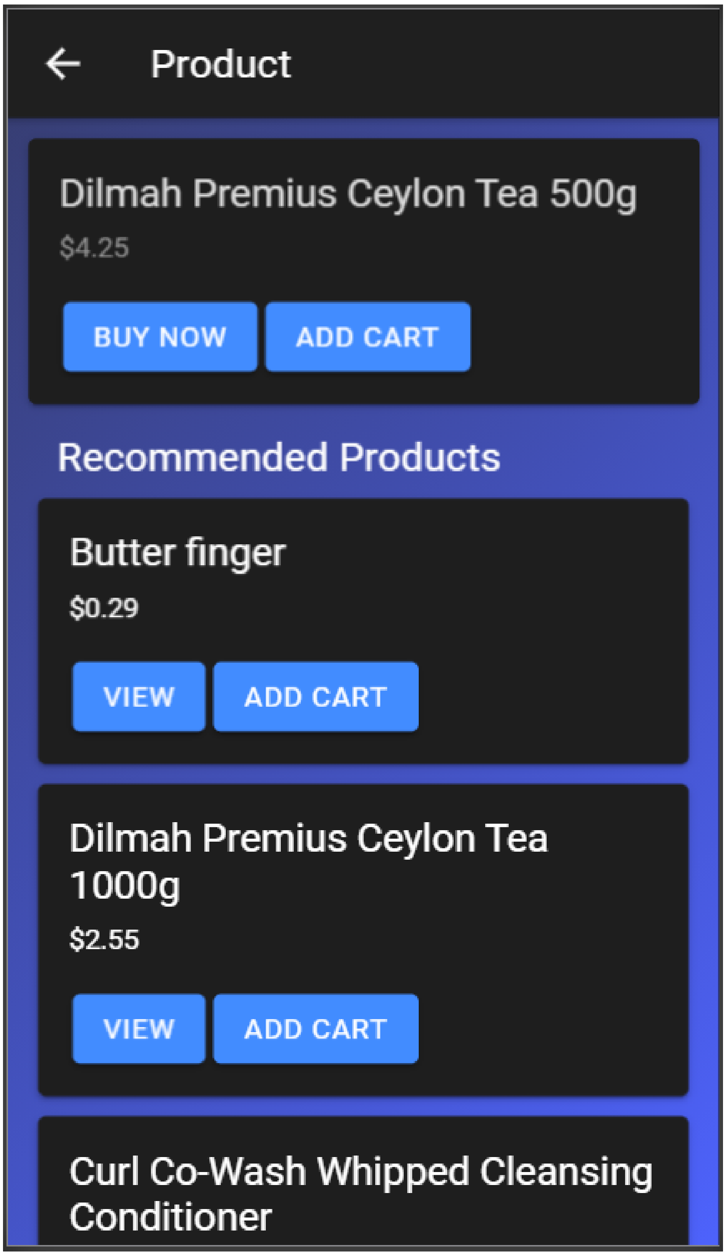


Figure 3:11:Recommended products Interface

`

3.2.Research findings

A survey is conducted among the people who are using supermarkets for purchasing products in order to identify what people think about the product recommendation system.

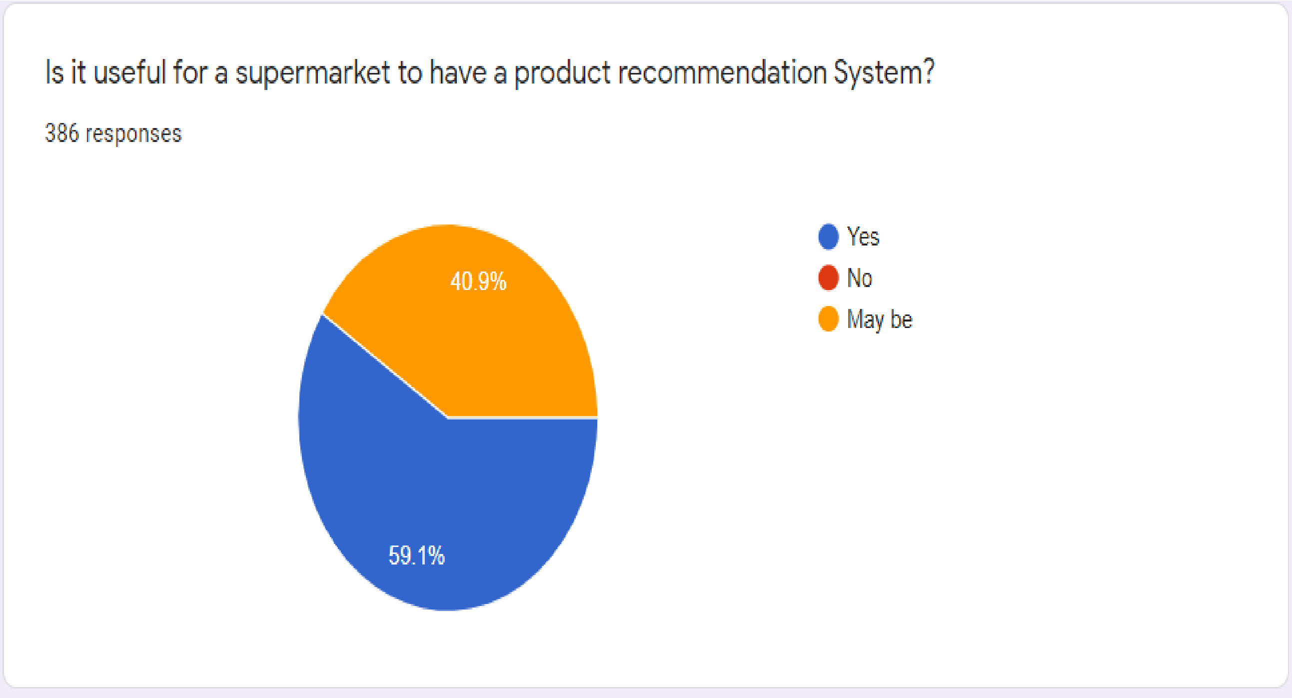


Figure 3:12: Pie chart of usefulness of product recommendation system

Figure 3.12 shows that 59.1% of the people felt that it is useful to have a product recommendation system at the supermarket while the 40.9% of people felt that the product recommendation system may be useful at supermarkets. No one had stated about the uselessness of product recommendation system at supermarkets. Majority of people preferred about the usefulness of having product recommendation system. Therefore, it is quite clear about the usefulness of product recommendation system at supermarkets.

`

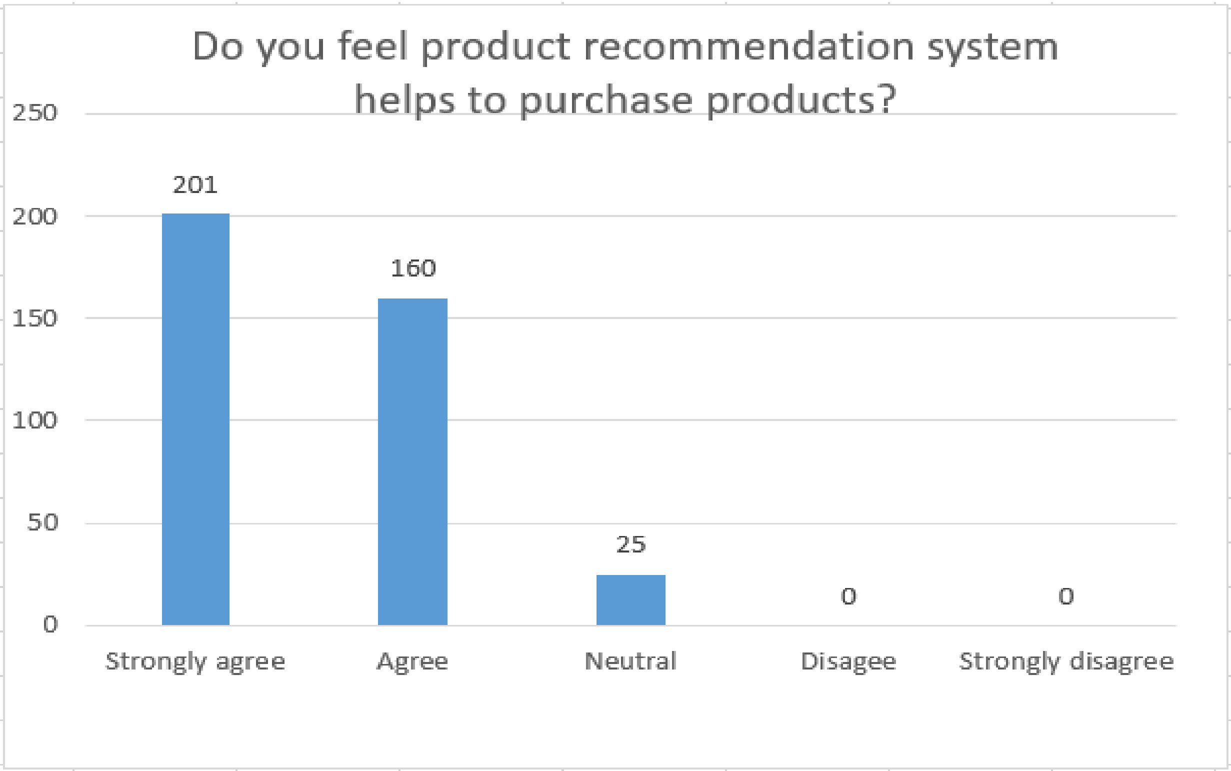


Figure 3:13: Bar chart of whether product recommendation helps in purchasing products

Figure 3.13 shows that out of 386 people, 201 people strongly agree that product recommendation system helps to purchase products, 160 people agree that product recommendation system helps to purchase products and the rest of the 25 people remained neutral. However, no one disagree or strongly degree that product recommendation system helps to purchase products. As most of the people strongly agree, product recommendation system helps to purchase products.

`

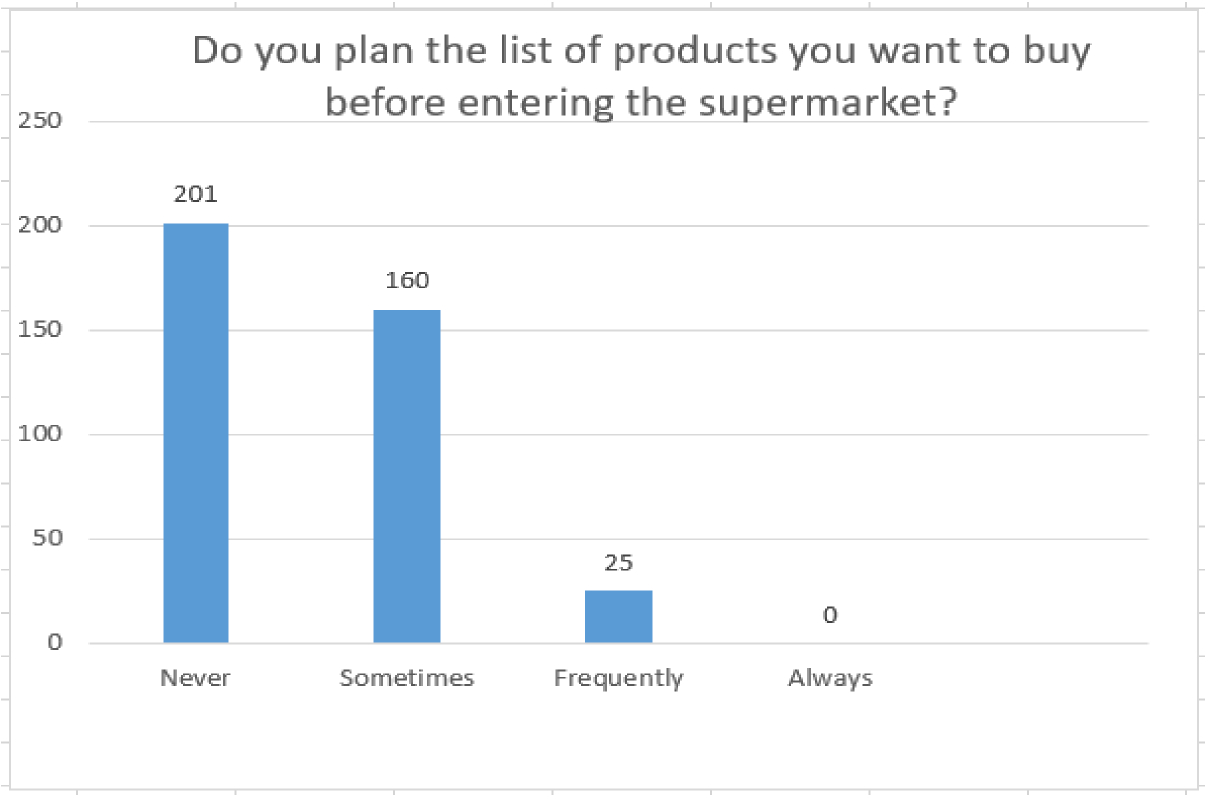


Figure 3:14: Bar chart of whether people plan before going to supermarket

Figure 3.14 shows that out 386 people, 173 people sometimes plan the list of products before entering the supermarket, 153 people never plan the list of products before entering the supermarket, 51 people frequently plan the list of products before entering the supermarket and the rest 9 people always plan the list of products before entering the supermarket. As per the results most of the people sometimes plan the list of products before entering the supermarket. It shows that they have no clear idea about the contents of their purchase. Therefore, product recommendation system will be helpful to the customers to purchase products.

`

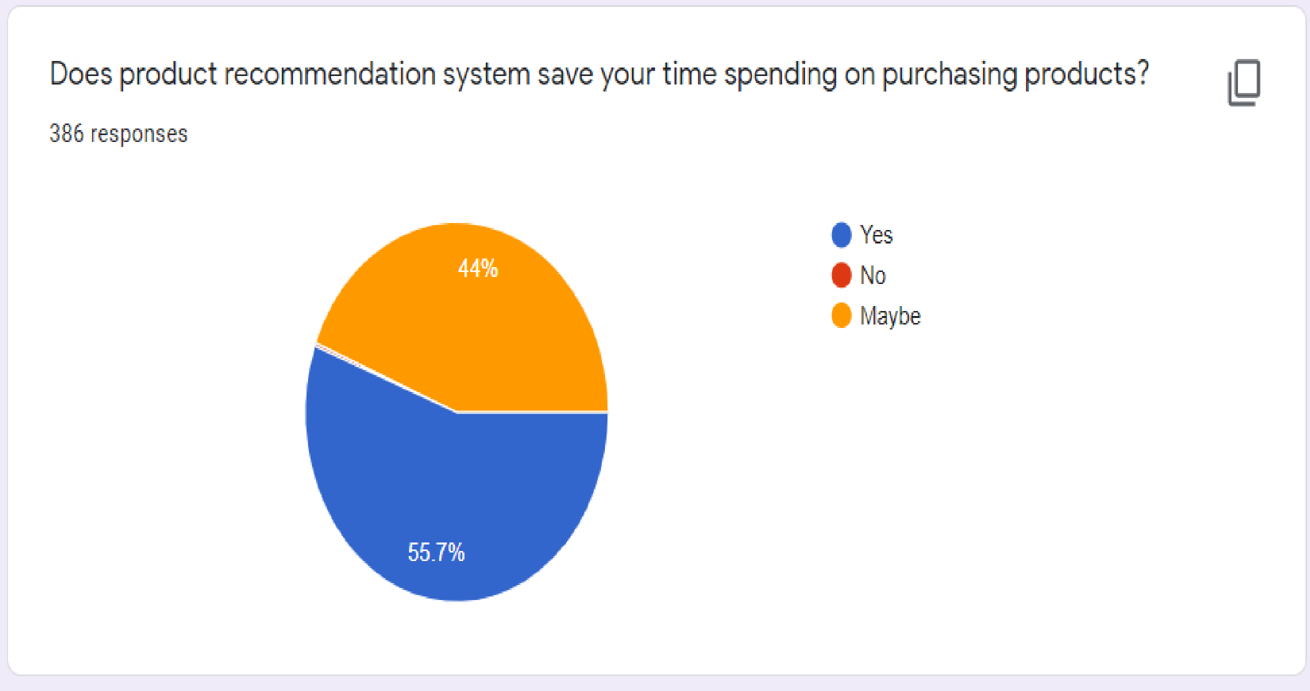
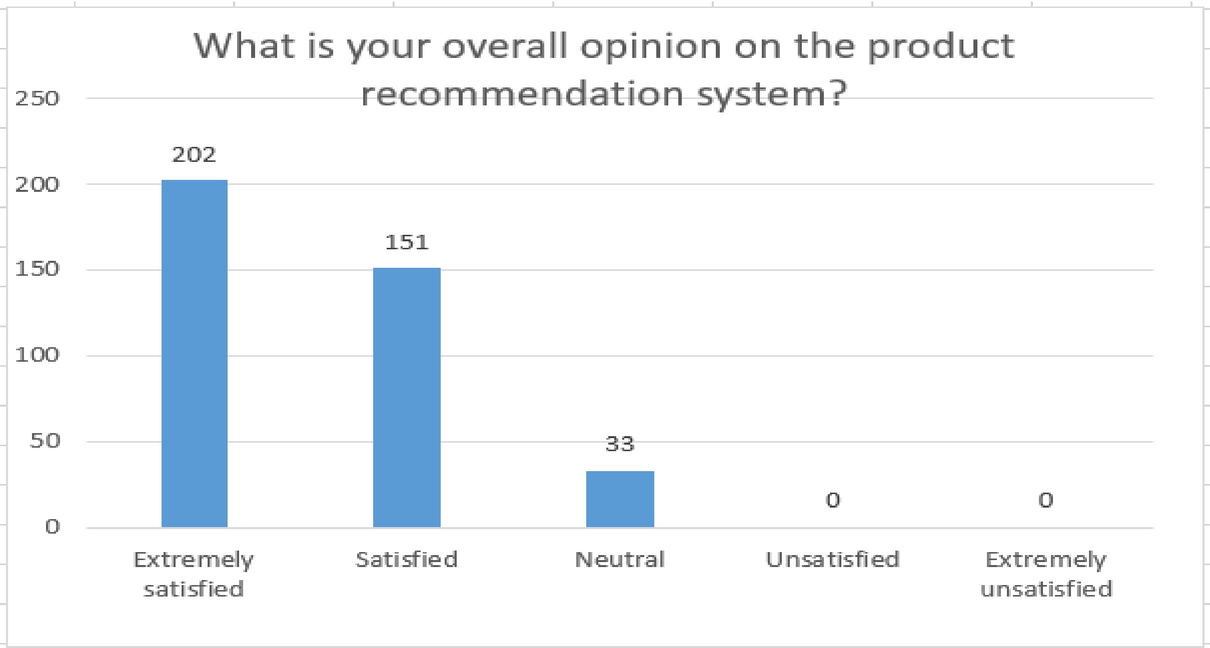


Figure 3:15: Pie chart of whether product recommendation system saving time

Figure 3.15 shows that 55.7% of people felt that the product recommendation system saves the time spend on purchasing products. 44% of people felt that product recommendation system may save the time spend on purchasing products. Clear majority of people felt that the product recommendation system saves time. Therefore, it is quite clear about the usefulness of product recommendation system at supermarkets.



`

Figure 3:16: Bar chart of overall opinion about product recommendation system

Figure 3.16 shows that out of the 386 people, 202 are extremely satisfied on the product recommendation system, 151 people are satisfied on the product recommendation system, 33 people felt it is neutral. As most of the people are extremely satisfied on, the product recommendation system it states that product recommendation is needed for the people in supermarket to help purchasing the products.

From the survey, found that there is positive impact on the product recommendation system in supermarket. People felt that product recommendation system helps in purchasing products and also felt that it saves their time spending on purchasing products. Therefore, it concludes that it is necessary to have a product recommendation system to attract both existing and new customers.

`

3.3.Discussion

The conclusion is reached that the user-based collaborative filtering is better in comparison with the item-based collaborative filtering from the results obtained through the techniques utilized for product recommendation system. The difficulty in finding similar customers at supermarkets is a clear reason for such result whereas the similar items can be easily found at supermarkets. It is clear that only a few products can be recommended through the user-based collaborative filtering. More products can be recommended through the use of item-based collaborative filtering. The proposed system solves the problem encountered by new customers and the problems at a new supermarket that has no purchasing history and product ratings. K-Apriori algorithm effectively generates the highly informative frequent item sets and the association rules to the supermarket.

The survey results show that the people tend to have a product recommendation system as it helps to choose products and save the time spend on purchasing the products. Therefore, it is a dire need to have a product recommendation system at supermarket.

A well-developed recommendation system helps the supermarkets to improve the experience of customers in purchasing and as well as to retain customers. The system provides suggestions not only from the preferences of regular customers but also from the opinions of other people as well.

`

4. Conclusion

The recommendation system is a much sought – after powerful software solution at supermarkets for many different sensitive issues encountered by customers over choosing the products from a wide range of products with the nearest standard. The recommendation system derives much satisfaction to the customers as it reduces the time taken for the search of their desired product and helps them to choose products in the dynamic environment of a supermarket in accordance with varying preferences of customers. Recommendation system is needed in supermarkets to retain the regular customers and to attract the new customers as well. Recommendation algorithms provide an effective form in marketing to achieve the target by furnishing the personal experience of each customer in the field of shopping. A methodology was adopted in this study for modelling and predicting the purchases at a supermarket. The main objective of this product recommendation system is to help customers to reach decisions on the purchase of preferred products through the recommendation. As far as a method of product recommendation is concerned, the purchasing history of each customer is collected and studied initially in view to increase the degree of satisfaction to potential customers. Recommendation is offered at two stages in this system. At first stage, recommendation is given in advance in prior to choose a product while in the second stage, the recommendation is given in subsequent to the selection of a product. The proposed system is based on collaborative filtering, clustering and association rules. Item-to-item collaborative filtering is preferred as it provides recommendations to all customers regardless the number of purchases and in the ratings given already on priority.

Following a particular sequence will increase the accuracy of the relationship between products and the sequential processing of dataset in supermarket might yield more solid features. These would be the delightful prospects for any potential student who would prefer to select this area of study in the future to improve the existing product recommendation system.

`

5. References

[1] K. C. A. S. A. Panayiotis Christodoulou, "A Real-Time Targeted Recommender System for Supermarkets," *19th International Conference on Enterprise Information Systems,* vol. 2, pp. 703-712, 2017.

[2] A. Demiriz, "Enhancing Product Recommender Systems on Sparse Binary Data," *Data Mining and Knowledge Discovery,* vol. 9, 2002.

[3] D. M. A. Yahya Dorostkar Navaei, "Dihedral Product Recommendation System for E-commerce Using Data Mining Applications," *International Journal of Computer & Information Technologies(IJOCIT),* vol. 3, pp. 610-631, 2015.

[4] H. Y. Duc-TrongLe, "Basket-SensitivePersonalizedItemRecommendation," *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence(IJCAI-17),* pp. 2060-2066, 2017.

[5] R. D. A. K. R. D. Debajyoti Mukhopadhyay, "A Product Recommendation System using Vector Space Model and Association Rule," *International Conference on Information Technology,* pp. 279-282, 2008.

[6] S. A. I. S. S. V. T. Andreas Kanavos, "Large Scale Product Recommendation of Supermarket Ware Based on Customer Behaviour Analysis," *Big Data and Cognitive Computing,* vol. 2, 2018.

[7] A. K. D. Loraine Charlet Annie M.C, "Market Basket Analysis for a Supermarket based on Frequent Itemset Mining," *International Journal of Computer Science Issues,* vol. 9, 2012.

[8] D. M. Ruma Dutta, "Offering A Product Recommendation System in E-commerce," 2011.

[9] B.-J. Y. J. S. S. M. K. Yong Soo Kim, "Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites," *Korea Advanced Institute of S cience and Technology,* pp. 381-393, 2005.

`

[10] Y. P. B. A. B. M. C. G. M. J. Y. K S Anupa Koti, "Digitalized Recommendation Engine for Supermarket Customers Using Frequency of Purchase," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 4, p. 4, 2016.

[11] M. S. M. P. O. K. M. B. S. Ruihai Dong, "Opinionated Product Recommendation," in *ResearchGate*, 2014.

[12] D. H. H. J. Jinyoung Kim, "Product recommendation system based user purchase criteria," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 9, pp. 5454-5462, 2019.

[13] C. d. P. P. E. D. C. Ruyther Parente da Costa, "Products Recommendation for Mobile Devices," *Journal of Information Systems Engineering & Management,* 2017.

[14] N. M. K. Y. T. S. Natsuki Sanoa, "Recommendation system for grocery store considering data sparsity," *19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems,* pp. 1406-1413, 2015.

[15] E. Haven, "Recommender System Application development," 6 December 2019. [Online]. Available: https://towardsdatascience.com/recommender-system-application-development-part-1-of-4-cosine-similarity-f6dbcd768e83#:~:text=Cosine%20similarity%20is%20the%20measure,via%20keywords%20or%20other%20metrics... [Accessed 10 September 2020].

[16] S. Remanan, "Association Rule Mining," 02 November 2018. [Online]. Available: https://towardsdatascience.com/association-rule-mining-be4122fc1793. [Accessed 07 September 2020].

[17] A. Nair, "analyticsindiamag," 05 June 2019. [Online]. Available: https://analyticsindiamag.com/beginners-guide-to-understanding-apriori-algorithm-with-implementation-in-python/. [Accessed 08 September 2020].

[18] AnishaD, "GeeksforGeeks," 14 September 2018. [Online]. Available: https://www.geeksforgeeks.org/association-rule/. [Accessed 07 September 2020].

`

[19] M. Techlabs, "towardsdatascience," 28 September 2017. [Online]. Available: https://towardsdatascience.com/what-are-product-recommendation-engines-and-the-various-versions-of-them-9dcab4ee26d5. [Accessed 08 September 2020].

[20] m. qutbuddin, "towardsdatascience," [Online]. Available: https://towardsdatascience.com/comprehensive-guide-on-item-based-recommendation-systems-d67e40e2b75d. [Accessed 08 Spetember 2020].

[21] R. Jain, "hackerearth," 24 March 2017. [Online]. Available: https://www.hackerearth.com/blog/developers/beginners-tutorial-apriori-algorithm-data-mining-r-implementation/. [Accessed 08 September 2020].

[22] AlindGupta, "geeksforgeeks," 12 June 2019. [Online]. Available: https://www.geeksforgeeks.org/implementing-apriori-algorithm-in-python/. [Accessed 08 September 2020].

[23] "Software Testing Help," 02 August 2020. [Online]. Available: https://www.softwaretestinghelp.com/apriori-algorithm/. [Accessed 08 September 2020].

[24] E. Haven, "Recommender System Application development," 06 December 2019. [Online]. Available: https://towardsdatascience.com/recommender-system-application-development-part-1-of-4-cosine-similarity-f6dbcd768e83#:~:text=Cosine%20similarity%20is%20the%20measure,via%20keywords%20or%20other%20metrics.. [Accessed 07 September 2020].

`

Appendix- A MongoDB Atlas

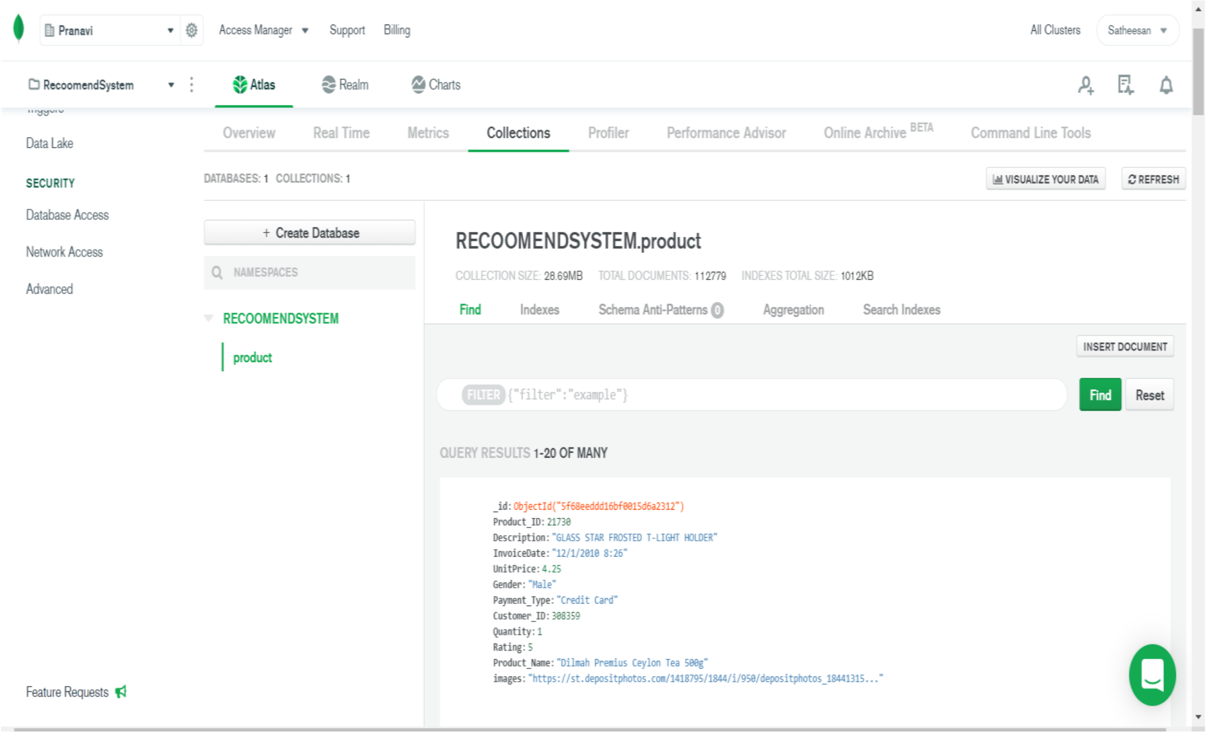


Figure 5:1:Mongodb Atlas

Appendix- B Deployment Backend

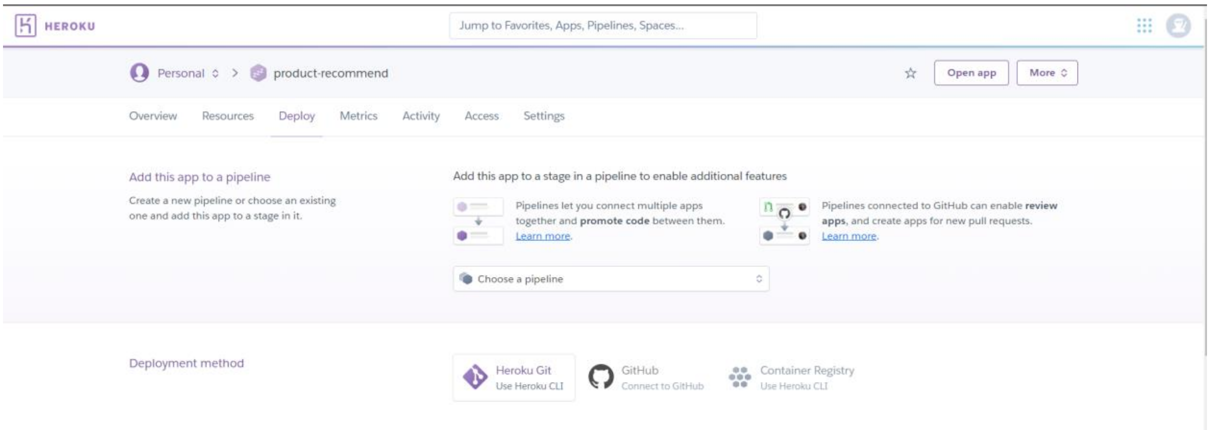


Figure 5:2:Deployment Backend