MARKET BASKET ANALYSIS OF INSTACART

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

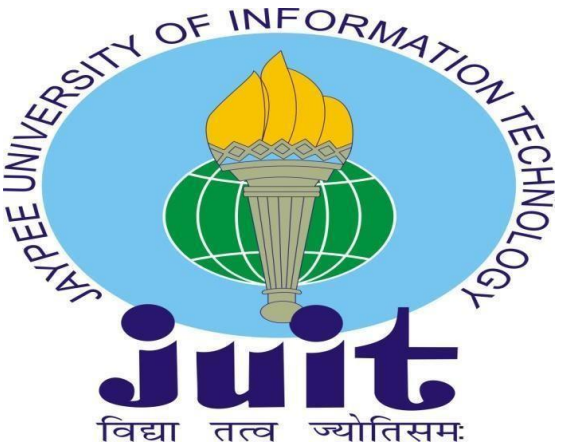
By

Reet Sethi (191222)

Under the supervision of

Dr. Anupriya Kaur, Dr. Rakesh Kanji

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

DECLARATION

I hereby declare that the work presented in this report entitled **"Market Basket Analysis of Instacart"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of
Information Technology Waknaghat is an authentic record of my work carried out over a period from January 2023 to May 2023 under the supervision of **Dr.Anupriya Kaur** (HSS) and **Dr. Rakesh Kanji** (CSE).

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream **Data Science**. The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Reet Sethi, 191222

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Anupriya Kaur	Dr. Rakesh Kanji
Associate Professor	Assistant Professor (SG)
HSS Department	CSE Department

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: .....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _____ **Department:** _____ **Enrolment No** _____

Contact No. _____ **E-mail.** _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at .....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><td>Copy Received on</td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="3">Similarity Index (%)</td><td colspan="2">Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td colspan="2">Character Counts</td><td></td></tr>
<tr><td rowspan="2">Submission ID</td><td colspan="2">Total Pages Scanned</td><td></td></tr>
<tr><td></td><td colspan="2">File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing that makes it possible to complete the project work successfully.

I am grateful and wish my profound indebtedness to Supervisor **DR. ANUPRIYA KAUR,** ASSOCIATE PROFESSOR, Department of Humanities and Social Science,s and cosupervisor **DR. RAKESH KANJI**, ASSISTANT PROFESSOR (Senior Grade),
Department of CSE, Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisors in the field of "**Research Area**" to carry out this project. Their endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, establishive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **DR. ANUPRIYA KAUR**, ASSOCIATE PROFESSOR, Department of Humanities and Social Sciences, and **DR. RAKESH KANJI**, ASSISTANT PROFESSOR (Senior Grade), Department of CSE, for their kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and noninstructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

REET SETHI (191222)

Department of Computer Science & Engineering

Jaypee University of Information Technology

TABLE OF CONTENT

S.No Title **Page No.**

LIST OF FIGURE

LIST OF ABBREVIATIONS

Abbreviation	Full Form
FP	Frequent Patterns
TF-IDF	Term Frequency-Inverse Document Frequer

LIST OF TABLES

ABSTRACT

Market Basket Analysis is primarily one of the data mining techniques which is utilized by retailers to discover key associations between items or items or goods. The goal of the effective study of items bought along with is to enhance customer satisfaction and improve the supplier's sales while increasing profitability. Instacart is an America-based grocery store that provides same-day delivery. The main purpose of this project is to utilize the open data file provided by Instacart to uncover customer purchase arches using transactional orders, descriptive analysis on items bought along with, and finding rules to enhance sales of the store. Various models have been developed to mine the association rules. The most common models are Apriori and FP growth, and their variations in data reduction techniques. In this report, we propose an approach that compares association rules mined at different minimum support counts for the most ordered product at the store. We have also implemented the recommendation system using the TF-IDF neighborhood model that utilizes cosine similarity measures to recommend items or goods from similar utilizers.

Keywords: Market Basket Analysis, Association rule mining, Apriori, TF-IDF neighborhood, Recommendation system.

Chapter 1 - INTRODUCTION

1.1 Market Basket Analysis

The rise of web-based business has created more opportunities for retailers to reach those shoppers, and innovation has accustomed shoppers to new purchasing behaviors. Retailers are increasingly using information analysis to identify growth opportunities and support customer item contributions. Market bin examination is a fundamental method utilized by major retailers to discover connections between merchandise. It works by looking for parts combinations that often occur alongside in exchanges. To put it another way, retailers care about how shopper interact with the items or goods they buy.

Starting right off the basics Market basket analysis is a data processing method utilized by companies to increase sales by attempting to understand the buyers buying repetition arch. It entails analyzing huge data sets, such as purchase history, to reveal product groups, as well as merchandise that appear to be brought alongwith.

It is conceivable for merchants to sell the items or goods alongwith or to create target prospects for the second product if it is known that shopper who buy one product are likely to buy another. If alcohol is sold right next to an aisle where diapers are sold, shopper who are likely to buy beer will be more likely to do so. Even though the young dads' outcome makes sense, one would often immediately consider collecting up on stuff for themselves and their younger ones before the weekend begins. It is vital information that may be utilized to cross-sell or up-sell items or goods based on how strong those relationships are.

Association rules are employed in market basket analysis to forecast the chance that 2 product are bought alongwith. By count however typically bound things occur alongwith, association rules search for associations that occur a number of times more often than they should.

Association rule-based models basically consist of AIS, SETM, and Apriori. The Apriori model is persistently utilized by data scientists in research publications about market basket analysis and is utilized to find persistently occurring items or goods in the repository and then assess their repetition if we enlarge the data files in size.

Market basket analysis is powerful enough to spice up sales and client satisfaction. Shop keepers will optimize product placement, supply special deals, and make new product bundles by mistreating information to see that items or goods are often purchased alongwith.

Using these upgrades gives us the power to extend sales for the merchandiser whereas additionally creating the searching expertise is a lot of productive and valuable for shopper. Customers could feel a more powerful connection or complete loyal towards the corporate if market basket analysis is utilized.

1.2 Problem Statement

In the present day scenario, businesses need to build adequate and inexpensive advertising strategies in customer-centered markets that can adapt to shifts in purchaser perceptions and product demands. Data Mining has perhaps received the finest solution to this need with innovation.

- Market basket analysis will look into customer baskets such as to monitor buying arches and improve customer experience.
- Analyzing recurring arches would present related goods, thus, a common recurring progression can be found and therefore causing sales growth.
- Using non private randomized information on client orders over time to predict that antecedently purchased items or goods are during a utilizer's next order can enhance shopper satisfaction thereby increasing the ratings of the store.

1.3 Objectives

It is the need of the hour for all the retailers to know what a customer would add to their basket at the time of buying groceries in order to increase their sales and profit and hence, a recommendation system based on previous items bought by a utilizer can be a solution to the problem.

The main objectives of our project includes:

a. The study of arch of the purchase behavior of purchasers (items purchased with similar items).
b. To study the effect of time on shopper and the relative items or goods they buy.
c. To check **which items or goods are likely to be bought** by the shopper again and recommending them to them.
d. To discover **Association rules using the apriori model and FP growth** model and analyze transactional repositories.
e. To **study the changes in the arch growth** and to note down the uplift or otherwise, of the market.
f. To **make recommendations** on the basis of items in the customer cart.

1.4 Methodology

A recommending system's aim is to get perceptive suggestions for things or product which will are liked by a bunch of utilizers. The elemental model, admire Apriori and FP Growth, gathers info about individuals' liking along with realizing that once people have amendments in their preferences,

They oftentimes tend to be usually inquisitive about gravies, thus get alimentary paste and wine. The central element in making a recommendation engine is that of the association rule. While applying the Association Rule to a knowledge set containing information from previous looking baskets, it produces a variety of rules. Every rule includes an assortment of product names acting as antecedents, a single product name acting as a consequence, Associate in Nursing a number of category measures acting as lift, support, confidence, and antecedent and support.

The Apriori modelic rule is utilized for persistent Data mining, has been improved with FP growth (AKA Association Rule Mining). It is utilized as the systemic method to get common arches or associations in information sets. If we have a tendency to take, for example, a persistent arch normally stores group action data may well be that folks sometimes get chips and cold drinks alongwith.

The Apriori model establishes persistent arches by generating itemsets and deciding the foremost persistent itemset over a "minimal support count" threshold. One easy principle greatly reduces the scale of the itemset within the repository:

If an item set is persistent, most of its subparts will also be persistent respectively.

However, the Apriori model carries a deadly error. Apriori required a couple of repository scans to decide the assist remember of every object and itemset. When the repository is massive, this may devour loads of disc input output and computing power. As a result, the FP-Growth set of rules turned into advanced to make amends for this downfall. It best scans the repository two times and shops all the records in a tree structure (FP-tree). The base represents null, every node represents an object, and the node institutions are itemsets with the order maintained whilst making the tree. The FP-tree is concise and is utilized to establish massive itemsets directly. Once an FP-

tree has been established, the common itemsets are found by the use of a repetitive divide-and-overcome approach.

Apriori model:

One of the most basic technique for basket analysis is apriori model and this is an testament of purchasers buying basket.

The final aim is to search for persistent itemsets, which are product combinations that are persistently purchased alongwith. The domain's method call is persistent Itemset Mining.

When we utilize common object units and the Apriori set of rules, basket evaluation isn't the best sort of evaluation we are able to perform. In theory, it may be implemented to any subject matter in which you need to observe often taking place elements.

Fast models for Mining Association Rules, posted via way of means of scientists Agrawal and Srikant, changed into the primary to advocate a option to this problem. The well-known Apriori set of rules changed into their first solution.

Although the proceedings of this article are intended to be more practical than technical, it is critical to understand the fundamentals of the Apriori model.

It is important to note that there are several factors to consider here:

- How do you find persistently occurring itemsets in a data file?
- How can you efficiently find persistent item sets in a data file?
- How should you prepare your data for the Apriori model?

Let's begin at the beginning: you have a data set in which shopper purchase multiple items or goods. Your goal is to discover which product combinations are persistently purchased alongwith.

You must organize the data in such a way that each part has a subset of items or goods. Each of those sets contains items purchased in the same transaction.

The most basic solution would be to loop through all the proceedings and count all the product combinations within the transactions. Unfortunately, this will take far too long, so something better was needed as a solution.

FP Growth:

Han et al paper's mining persistent arches, while not candidate generation, describes the FP growth model, wherever "FP" stands for the persistent arch. The primary step in FP-growth is to enumerate item frequencies and establish persistent things given a data file of transactions. Unlike Apriori-like models arched for an equivalent purpose, the second step of FP-growth encodes transactions employing a suffix tree (FP-tree) structure instead of expressly generating candidate sets that are usually valuable to get in terms of your time and space. Following the secondary step, the FP-persistent tree's itemsets may be extracted. We tend to enforce a parallel version of FP-growth known as PFP in spark. mllib, as represented in Li et al., PFP: Parallel FP-growth for question recommendation. PFP distributes the work of growing FP-trees supported dealings suffixes and is so a lot of scalable than one machine interpretation. FP growth works on itemsets that may be extracted reckoning on the kind of model we tend to are using. The following hyper-parameters are taken into consideration once implementing FP growth algorithm:

Minimum Support: the vacant minimum of support needed for an itemset to be classified as persistent. For example, if an item seems in 3 out of 5 transactions, it's 3/5=0.6 support.

Minimum Confidence: the amount of confidence required to get associate Association Rule. Confidence indicates however often an association is found to be true. For example, if X seems fourfold within the transactions itemset and X and Y co-occur solely twice, the arrogance for the rule X => Y is 2/4 = 0.5. They are doing not have an effect on persistent itemset mining, however it specifies the minimum confidence for generating association rules from persistent itemsets.

By equation raise is outlined as:

$$\text{Lift} = \text{Support (antecedent U consequent)} / (\text{support (antecedent)} \times \text{support (consequent)})$$

The lift could be a live of the rule's deviation from the applied mathematics independence model of the rule body and rule head. The raise could be a number starting from zero to infinity:

A lift worth bigger than one indicates that the rule body and rule head seem along a lot of often than expected, implying that the incidence of the rule body influences the occurrence of the rule head.

A lift below one indicates that the rule body and rule head appear less persistently alongwith than expected, implying that the rule body includes a negative result on the rule head's occurrence.

Number Partitions: the quantity of partitions want to distribute the work.

The FP Growth Model offers:

1. Persistent Itemsets: persistent itemsets in information Frame format with the subsequent Columns: Items: array: a group of items.
2. Repetition: long: A count of what number times this itemset was seen given the model parameters that were configured.
3. Association Rules: association rules confidently bigger than minimum Confidence, within the style of a knowledge Frame with the following columns:

3.1 Array: The itemset that contains the hypothesis of the association rule. Consequent: array: associate itemset that perpetually contains one part representing the association rule's result.

3.2 Double your confidence: For a definition of confidence, see minimum Confidence above. Lift could be a metric of how well the antecedent predicts the consequent.

4. Transform: The rework technique can compare the things in each dealings in things Col to the antecedents of every association rule. If the record contains all of the antecedents of a selected association rule, the rule is deemed applicable, and its consequences are supplementary to the prediction result. As a prediction, the rework

technique can summarize the outcomes of all applicable rules. The prediction column is of an equivalent information kind as items gap and doesn't contain any existing items from items Col.

Disadvantages of FP Growth:

1. FP Tree is more long and troublesome to establish than Apriori.
2. It may be more time consuming.
3. The formula might not slot in shared memory if the repository is large.

Advantages of FP Growth:

1. When compared to Apriori, that scans the information for every iteration, this formula solely has to scan the repository one more time.
2. This model doesn't combine items, which permits it to perform faster. .
3. It is better and economical for mining each long and short persistent arches.

Association Rules:

A model that mines the association between things is thought as an association rule mining model, and it's often wont to mine the association data between things. Association rule mining models will discover potential connections between completely different qualities of faculty students based mostly on information from school students' lives and learning, which might assist academics in discovering the issues and their own strengths of various students and achieving teaching supported their aptitude.

In our case maybe association rules are wont to notice the association between things ordered by the shoppers from the market.

Association rule mining formulas are primarily utilized to confirm the association method of redundant and non-critical data in a very information based on attribute importance. The idea of the association rule mining model is a lot of vital in data processing theory's attribute technique. This method saves a big quantity of search space whereas getting the simplest or near-optimal reduction set. For giant information

mining, association rule mining models may be combined with several attribute reduction models. Notice the minimum reduction set or core mistreatment the association rule heuristic technique additionally the established discriminant matrix to search out the core attribute set.

At the terribly basic level, our work is to feature attributes consistent with the importance of different attributes till the knowledge gained is comparable to the classification ability of the initial call table.

Recommendation System:

A recommender system, also referred to as a recommendation system, could be a form of info filtering system that tries to predict a utilizer's "rating" or "preference" for associate item. They're principally employed in industrial settings.

Examples of such applications embody Amazon product recommendations, Spotify music recommendations, and, of course, Medium stories. The well-known Netflix Prize is additionally a contest utilized for recommendation systems.

More formally, the recommender drawback may be outlined as decisive the mapping (c, I) R, wherever c represents a utilizer, I represents an item, and R represents the utility of the utilizer being counselled with the item. The things are then sorted by utility, and therefore the high N items are recommended to the utilizer.

There are numerous styles of recommendation manufacturing strategies gift within the market, one in every of them that we've got utilized for our project is as follows:

Even if one limits oneself to rules with just one item as a result, association rule mining sometimes establishes so much too several rules. (This will increase the assembly length by increasing the things) and a lot of complicated rules add nearly nothing to the insights on the info set. Take into account the easier rules, which discuss with a rule with multiple items in the outcome, i.e. rules with an equivalent antecedent however only one item from the complex rule's effect. All of those rules should be gift within the results as a result of their support and confidence can't be below that of the lot of complicated rule.

That is, if a rule A B ->F T, the foundations A B ->F and A B ->T should even be present in the output.

Collaborative filtering for Recommendation systems:

Collaborative filtering filters info by utilizing the system's interactions and information collected from different utilizers. It's supported the belief that individuals who in agreement on sure items' evaluations are doubtless to agree once more in the future.

The idea is simple: after we wish to utilize a replacement product from a store, we regularly ask our friends for recommendations. We tend to naturally place a lot of trust in recommendations from friends who have similar tastes to our own.

The questionable similarity index-based technique is employed by the bulk of cooperative filtering systems. Variety of utilizers are chosen within the neighbourhood-based approach supported their similarity to the active utilizer. A weighted average of the ratings of the chosen utilizers is utilized to infer the active utilizer.

The relationship between utilizers and items is that the focus of collaborative-filtering systems. The similarity of things is set by the similarity of their ratings by utilizers who rated each items.

A utility matrix is commonly wont to confirm cooperative methods. The recommender model is tasked with the learning operator that predicts the utility of match or similarity to every utilizer. The utility matrix is often terribly sparse, massive, and contains values that are removed.

The circular function similarity formula is that the most elementary model for decisive vector similarity. The utility matrix that follows the primary matrix contains solely partial data, that is needed to predict the probability of the expected rating by the "root" that the utilizer might provide.

TF-IDF (Term repetition-Inverse Document repetition) based neighbourhood for recommendations

On a TF-IDF weighted utility matrix, we investigate a neighbourhood technique that makes product recommendations to a target utilizer based on what other utilizers who have similar tastes have bought.
TF-IDF is persistently utilized to evaluate a word's significance inside a document. When a term appears more persistently in a document, it becomes more important; when it appears more persistently in documents overall, it becomes less important. A corpus can be utilized to establish documents (one bag of words) related to a query (another bag of words), regardless of the word order. We therefore utilized an analogy. Each product a utilizer has purchased can be thought of as a word in a document containing their purchase history.

We can locate similar utilizers (documents) and recommend the items or goods of similar utilizers in order to create recommendations for a target utilizer (a query) who will receive the recommendations.

Each row of the matrix represents a phrase, or more precisely, a product that utilizers have bought. Each row of the matrix represents a document that includes the utilizer's various product purchase history, sometimes referred to as UPH.

How do the TF-IDF work?

- By calculating the frequencies of items or goods for each row, the utility matrix is utilized to establish a term repetition matrix (utilizer purchase history).
- As shown below, an inverse document repetition vector is created for each product p by dividing the total number of utilizers by the total number of utilizers who have purchased p.

$$idf_p = \log\left(\frac{\#(UPH)}{\#(UPH\ containing\ p)}\right).$$

- The TF matrix and IDF vector can then be multiplied to create the TF-IDF matrix.

- For each pair of the target utilizer and the other utilizers, the TF-IDF matrix is utilized to compute the cosine similarity vector, as illustrated below.

$$cosine\_similarity(i,j) = cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

- By taking the greatest K values from the Cosine Similarity vector (in our experiment, K = 20), the top K utilizers that are similar to the target one are chosen.
- According to overall sales from candidate recommendations, which are composed of the fewest number of the most similar utilizers and whose size is greater than or equal to N (in our study, N = 10), the top N goods of similar utilizers are recommended

1.5 Organization

The document contains a total of five chapters.

Chapter-1 provides a brief introduction about market basket analysis and various models utilized in the whole project. It also describes the problem statement and objectives of the project.

Chapter-2 addresses the literature review for the proposed work, that is, results of the related work along with the limitations of the existing solutions.

Chapter-3 describes the system development part that explains the detailed methodology for the proposed work.

Chapter-4 analyses the performance of the entire project line that includes the evaluation and results at different stages.

Chapter-5 discusses the conclusions and future scope for the proposed work.

Chapter-2 LITERATURE SURVEY

[1] **Market Basket Analysis Using Apriori and FP-Growth model**

Authors: Malika Hossain, AHM Sarowar Sattar, Mohit Kumar Paul

Journal: International Conference on Computer and Information Technology, 2019

Publisher: IEEE

Methodology: The goal here is to reduce the computational complexity. The outcomes show that by utilizing prime mercantilism items, it is possible to obtain nearly identical persistent itemsets and association rules in a very short period of time when compared to the outputs obtained by computing all the items. A comparison of time complexity shows that the FP Growth rule outperforms the Apriori model.

During this analysis, the authors of this paper utilized the FP Growth models and Apriori models to find in-style items in transactional data files and to discover relationships between those items. We additionally projected a replacement methodology for mining association rules by selecting a specific share of persistent things from our data file, and we ran various tests to replicate our findings.

Disadvantages: The range of percentage of top selling items or goods is not set where association rules and persistent item set will remain the same with or without reduction.

[16] **Efficient Market Basket Analysis based on FP-Bonsai**

Authors: Behera Gayatri

Journal: ISMAC 2017

Publisher: IEEE

Methodology: To mine persistent arches, an economical FP-tree modelic rule is utilized in this paper. The items or goods are organized based on the mining results in order to best suit the customer's desires and fascinate. The FP-bonsai model produces an extremely efficient persistent itemset mining model that effectively exploits similar toned constraints. Using FPbonsai instead of the standard FP-magnification model improves potency and reduces the time required to find persistent arches. Using the ExAnte property, all of the FP-trees created recursively during the FP-magnification computation are heavily reduced and cropped, resulting in a computation with an additional minute range of more minuscule trees. An FPBonsai is a miniature FP-tree created by growing and pruning. The outcoming methodology overcomes the most significant disadvantage of FP magnification, namely its memory requirements, on the one hand, and the main disadvantage of ExAMiner [4], that is the input output rephasing the reduced data files to disc on the other.

Disadvantages: The pruning could have been performed better.

[3]Comparative Analysis of Market Basket Analysis through Data Mining Techniques

Authors: Sonu Mittal,Shish Kumar Dubey,Seema Chattani,Vinod Kumar Shukla

Journal: ICCIKE 2019

Publisher: IEEE

Methodology:

Exploration of techniques like the Market Basket Analysis greatly help the purchaser and for the betterment of sales strategies by analyzing the sales of items with data-mining techniques utilized for this purpose .The differences include the establishion of purchasing arches from the purchaser end and production arches from the corporate end, which as an alternative aids in procuring or purchasing the product. It is critical to evaluate the activities of business shopper, and numerous methoding techniques are available to do so. This paper compares two widely utilized data mining techniques for understanding buyer behaviour: Association Rule Mining (ARM) and cooperative Filtering (CF), both of which are utilized in product recommendation.

Data mining helps businesses make sound business decisions and understand purchaser behaviour. Market Basket Analysis is the procedure of finding associations amongst buyer's purchasing arches, whereas Association Rule Mining is the processlogy utilized to discover associations between completely different things bought by the customer.

[4]Market Basket Analysis with Data Mining Methods

Authors: Andrej Trnka

Journal: ICNIT

Publisher: IEEE

Methodology:

This paper explains how Market Basket Analysis is utilized with a 6 alphabetic character methodology. Data processing methods open up a wide range of market opportunities. Basket Market Analysis is one of them. Six alphabetic character methodology makes utilize of a number of applied math techniques. We can improve the method's results and change its alphabetic character performance level by applying Market Basket Analysis (as part of information mining) to 6 alphabetic characters (in one of its phases).

In our research, we utilized the GRI (General Rule Induction) formula to derive connection rules between market basket items or goods. These associations demonstrate the items or goods' variability. To demonstrate the mutuality of the items or goods, an internet plot was utilized. Taking the ultimate formula into account was CS.O.

This system was first and foremost utilized to establish rule-based profiles.

[5]Application in Market Basket Analysis Based on FP-growth model

Authors: Liu Yongmei, Guan Yong

Journal: WCCSIE

Publisher: IEEE

Methodology:

Market basket analysis provides insight into the merchandise by indicating which items or goods are likely to be bought alongwith and which are likely to be bought separately. Market basket analysis is a powerful tool, particularly in retailing, where large baskets are required, due to its ability to figure with thousands of items. The FP-growth recursive programme may be a good choice for mining persistent arches. It is not necessary to obtain lengthy candidate sets. It only scans the data twice, and thus the most persistently

.

In this paper, Visual C++ is utilized to create a programme that mines persistently occurring itemsets using the FP-growth modelic programme.

Disadvantages: It is difficult to mine favorably when the information is too large to create an FP-tree in memory, mine favourably. Based on the encouraging results, we intend to investigate the FP-tree model as future work.

[6]Data Mining Techniques to Build a Recommender System

Authors: Alicia Huidobro Espejel, Francisco J. Cantu-Ortiz

Journal: ISCSIC 2021

Publisher: IEEE

Methodology:

Despite the fact that recommender frameworks are not a new concept, they continue to be significant due to their wide range of applications and potential impact on sales or earnings. There are several approaches to developing a proposal framework, and the one chosen depends on a number of factors. For instance, the assets that are specialised for the organisation and the information that is easily available. The most effective way to create a recommender system using information mining techniques is the topic of this study. Then, we proceed with local area revelation and market crate investigation in greater detail. The two techniques are effective and simple to use. We outline our method for carrying them out as well as the benefits and drawbacks we discovered.

Despite not being a new concept, recommender frameworks are nonetheless significant due to their wide range of applications and potential impact on sales or earnings. Fostering a proposal framework can be done in a variety of ways, and the method chosen depends on a number of factors. For instance, the information that can be removed and the organization's specialised resources This study focuses on the most effective way to create a recommender system using information mining techniques. Then, we proceed with two other tactics in further detail: neighbourhood revelation and market crate analysis. Both of the two methods are quick and effective. We discuss the advantages and drawbacks we discovered, as well as our system for implementing them.

Disadvantage: Better alternatives for recommendation systems could be used for the recommendations.

[7]Apriori Algorithm & FP-Growth Algorithm to Mine Association Rules

The first thought might be to look for a pre-made set of rules on the scikit-research website. This set of rules, on the other hand, isn't supported by using Scikit-research. Sebastian Raschka's MLxtend library, fortunately, includes Apriori and FP-Growth algorithms for extracting common object units for similar evaluation. The affiliation examination consists of a few calculation phrases that are simple for non-technologists to understand. Furthermore, because it is an unmonitored mastering device that searches for hidden patterns, no data evaluation or version building is required. This could be very useful in a few cases of information exploration and could pave the way for a deeper dive into the information.

Apriori is a popular formula used in association rule learning applications to retrieve frequently seen item sets. The Apriori algorithm was created to control massive data transactions, such as store purchases. The FP-growth algorithm is a more advanced version of the APRIORI algorithm, which is commonly used for frequent mining patterns. It is common practise to examine patterns or associations in data sets.

An itemset is said to be "frequent" if it meets a user-defined level of support. For example, if the support price is set to 0.3 (30%), a frequent item set is defined as a collection of items that appear in at least 30% of all transactions in the directory.

Chapter-3 SYSTEM DEVELOPMENT

3.1 Understanding data file

The data file utilized for the implementation of the proposed work is Instacart's open Dataset which contains orders of the customers over time. The utilizer details are unidentified and have a sample of 200,000 utilizers of Instacart and 3 million records of groceries. There are on an average a range of 4 and 100 grocery orders by an individual purchaser. The data file also contains t [10] he hour of day, day of the week.

Here, we have imported a total of five data files containing different details of the customer purchases and order placing.

Below is the schema of the data file we have utilized.

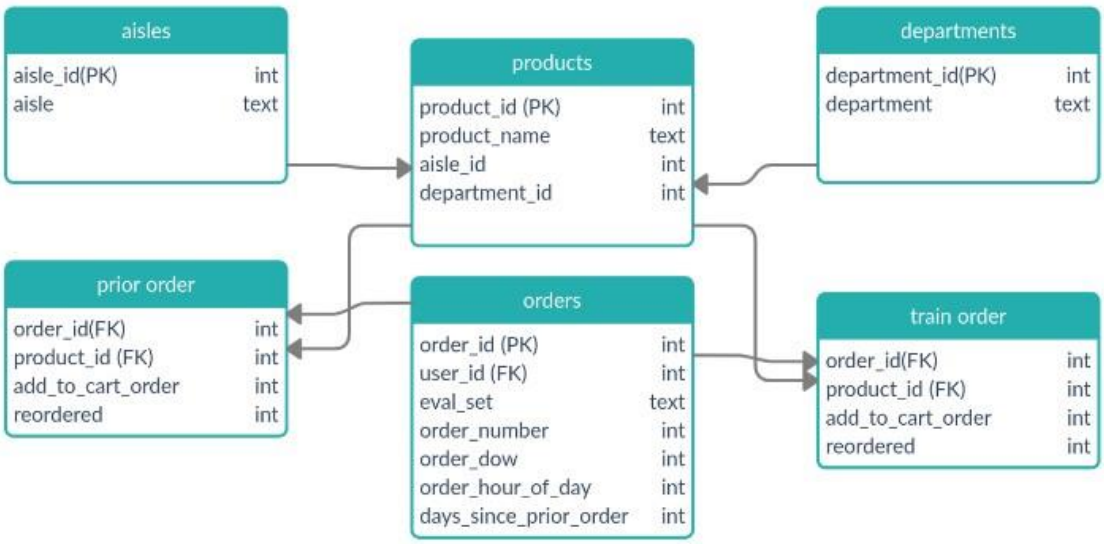


Table 1: Schema of the data file

1. Aisles (134 rows, 2 columns) :

The aisles file contains details of the aisle id and names of the aisles. The file contains details of 134 different aisles with columns:

a. aisle_id : Aisle Identifier

b. Aisle : Name of the aisle

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134 entries, 0 to 133
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   aisle_id  134 non-null    int64 
 1   Aisle     134 non-null    object
dtypes: int64(1), object(1)
memory usage: 2.2+ KB
```

Figure 1: Information of Aisle data file

	aisle_id	Aisle
0	1	prepared soups salads
1	2	specialty cheeses
2	3	energy granola bars
3	4	instant foods
4	5	marinades meat preparation

Figure 2: Sample Data view of aisle.csv

2. Departments (21 rows, 2 columns) :

The file describes the 21 different departments of the Instacart store. The columns in the data file are:

a. department_id : Department Identifier, primary key for the table

b. department_name : name of department

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Department_id  21 non-null     int64
 1   Department     21 non-null     object
dtypes: int64(1), object(1)
memory usage: 464.0+ bytes
```

Figure 3: Information of department.csv

```
array(<bound method Series.unique of 0             frozen
1               other
2              bakery
3             produce
4             alcohol
5       international
6           beverages
7                pets
8     dry goods pasta
9                bulk
10      personal care
11       meat seafood
12             pantry
13          breakfast
14       canned goods
15         dairy eggs
16          household
17             babies
18             snacks
19               deli
20            missing
Name: Department, dtype: object>, dtype=object)
```

Figure 4: Different departments

3. items or goods (49688 rows, 4 columns) :

The items or goods file contains the details of all the 49688 items or goods within 134 aisles and 21 departments. The columns in the file are :

a. product_id : product identifier, primary key for the table.

b. product_name : name of the product

c. aisle_id : foreign key from the aisle's table

d. Department_id : foreign key from the departments table

Few items or goods are: All-Seasons Salt, Bananas, Chocolate sandwich cookies etc.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49688 entries, 0 to 49687
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   2              49688 non-null  int64
 1   product_name   49688 non-null  object
 2   aisle_id       49688 non-null  int64
 3   department_id  49688 non-null  int64
dtypes: int64(3), object(1)
memory usage: 1.5+ MB
```

Figure 5: Information of items or goods table

	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13
2	3	Robust Golden Unsweetened Oolong Tea	94	7
3	4	Smart Ones Classic Favorites Mini Rigatoni Wit...	38	1
4	5	Green Chile Anytime Sauce	5	13

Figure 6: Sample data file view of items or goods.csv

4. Orders (1048575 rows, 7 columns) ;

The file contains the details of the orders, that is, hour of the day, day of the week, order id, order number etc. The columns in the table are:

a. order_id: order ID
b. user_id: customer ID
c. eval_set: to which evaluation this order belongs
d. order_number: The order sequence number (1 = first, n = nth)
e. order_dow: order was placed on the day of the week
f. order_hour_of_day: the time of day on which the order was placed

Figure 7: Information of orders.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 7 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   order_id                1048575 non-null  int64
 1   user_id                 1048575 non-null  int64
 2   eval_set                1048575 non-null  object
 3   order_number            1048575 non-null  int64
 4   order_dow               1048575 non-null  int64
 5   order_hour_of_day       1048575 non-null  int64
 6   days_since_prior_order  985475 non-null   float64
dtypes: float64(1), int64(5), object(1)
memory usage: 56.0+ MB
```

	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0

Figure 8: Sample data view of orders.csv

5. Prior Order (30 million rows, 4 columns) :

This file contains product identification information, product name, aisle ID, and department ID

a. order_id: foreign key
b. product_id: foreign key
c. add_to_cart_order: order in which each product was added to cart
d. reordered: 1 if this product has been ordered by this user in the past. 0 otherwise
    - "prior": orders prior to that users most recent order (~3.2m orders)
    - "train": training data supplied to participants (~131k orders)
    - "test": test data reserved for machine learning competitions (~75k orders)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32434489 entries, 0 to 32434488
Data columns (total 4 columns):
 #   Column             Dtype
---  ------             -----
 0   order_id           int64
 1   product_id         int64
 2   add_to_cart_order  int64
 3   reordered          int64
dtypes: int64(4)
memory usage: 989.8 MB
```

Figure 9: Information of prior_orders.csv

	order_id	product_id	add_to_cart_order	reordered
0	2	33120	1	1
1	2	28985	2	1
2	2	9327	3	0
3	2	45918	4	1
4	2	30035	5	0

Figure 10: Sample data view of prior_ orders.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 4 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   order_id           1048575 non-null  int64
 1   product_id         1048575 non-null  int64
 2   add_to_cart_order  1048575 non-null  int64
 3   reordered          1048575 non-null  int64
dtypes: int64(4)
memory usage: 32.0 MB
```

Figure 11: Information on train_orders.csv

	order_id	product_id	add_to_cart_order	reordered
0	1	49302	1	1
1	1	11109	2	1
2	1	10246	3	0
3	1	49683	4	0
4	1	43633	5	1

Figure 12: Sample Data view on train_orders.csv

The various datafiles were merged alongwith to form a master dataframe, in order to analyze and visualize various aspects of customer purchase arches.

```
Number of Rows :  10245522
Unique user_id:  63100 (  0.62% )
Unique order_id:  1015727 (  9.91% )
Unique product_id:  47956 (  0.47% )
Unique aisle_id:  134 (  0.00%
Unique department_id:  21 (  0.00% )
Unique add_to_cart_order:  145 (  0.00% )
Unique reordered:  2 (  0.00% )
Unique product_name:  47956 (  0.47% )
Unique aisle:  134 (  0.00% )
Unique department:  21 (  0.00% )
Unique eval_set:  2 (  0.00% )
Unique order_number:  100 (  0.00% )
Unique order_dow:  7 (  0.00% )
Unique order_hour_of_day:  24 (  0.00% )
Unique days_since_prior_order:  31 (  0.00% )
```

Figure 13: Information on master_df

	user_id	order_id	product_id	aisle_id	Department_id	add_to_cart_order	reordered	product_name	Aisle	Department	eval_set	order_number
0	22352	6	15873	75	17	2	0	Dryer Sheets Geranium Scent	laundry	household	prior	4
1	22352	6	41897	101	17	3	0	Clean Day Lavender Scent Room Freshener Spray	air fresheners candles	household	prior	4
2	22352	6	40462	31	7	1	0	Cleanse	refrigerated	beverages	prior	4
3	3107	8	23423	43	3	1	1	Original Hawaiian Sweet Rolls	buns rolls	bakery	prior	5
4	45082	13	36086	108	16	11	0	Philadelphia Original Cream Cheese	other creams cheeses	dairy eggs	prior	2
...	...	...	...	...	...	...	...	...	...	...	...	...
10245517	25247	3421083	45309	92	18	2	0	Purple Carrot & blueberry Puffs	baby food formula	babies	prior	24
10245518	25247	3421083	7854	117	19	1	0	Freeze Dried Mango Slices	nuts seeds dried fruit	snacks	prior	24
10245519	25247	3421083	11352	78	19	7	0	Organic Mini Sandwich Crackers Peanut Butter	crackers	snacks	prior	24
10245520	25247	3421083	21162	92	18	3	0	Organic Mixed Berry Yogurt & Fruit Snack	baby food formula	babies	prior	24
10245521	25247	3421083	39678	74	17	6	1	Free & Clear Natural Dishwasher Detergent	dish detergents	household	prior	24

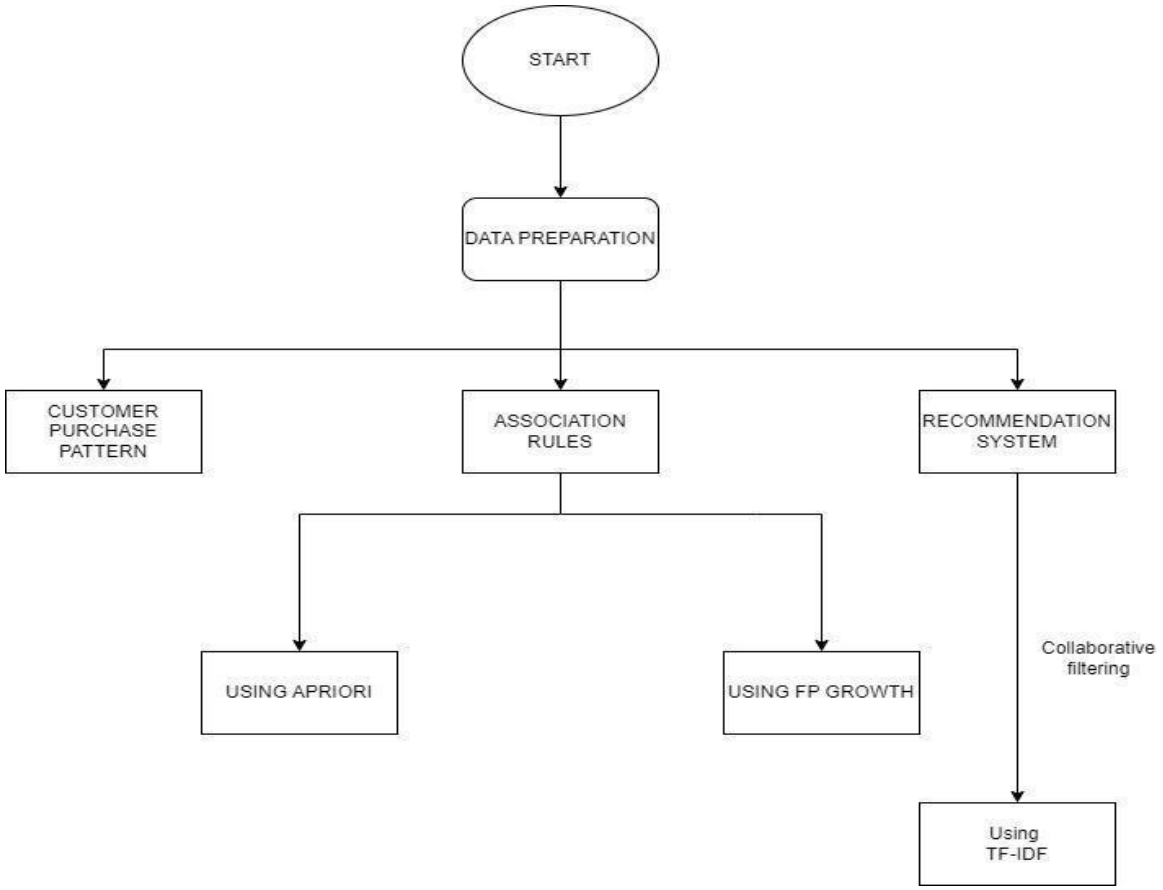
10245522 rows × 15 columns

Figure 14: Sample data view of master df

3.2 Flow of the Project:

Sequentially, data collection was done and then data cleaning to prepare the data for analysis and mining the association rules. This cleaned data is also utilized to recommend items or goods to the utilizers based on the items in their cart or their basket, previously ordered items or goods by an individual utilizer. The association rules are mined using two models - Apriori and FP growth. Both the models finds persistent item sets and then utilize these persistent item sets to mine the association rules. Here, association rules at different minimum support count are mined in order to see the rules difference for the most ordered product - banana. The recommendation system is based on similar utilizer cart, because they are more chances that a customer is more likely to order the items or goods of a similar utilizer cart.

Table 2: Work flow of the project



3.3 Libraries utilized

AI is the review space that assists computers in learning without being exactly adjusted. They are usually utilized to overcoming a variety of problems in life. In comparison to previous generations, Python libraries, systems, and modules are remarkably simple and robust today. Python is one of the most well-known programming languages for this task nowadays, having replaced many of the industry's dialects, thanks in part to its massive library selection. The below mentioned libraries in Python were utilized in the project:

i) Pandas package

Pandas is an open source, Python-approved package that gives the Python programming language better performance, intuitive information arches, and information research tools. The central information structure is the information casing. Pandas can perutilize various things. A line document in addition to a portion record, or a two-layered set of values, make up a data packaging. A series is an especially great assortment of record values.

ii) Numpy

An application for handling arrays in general is called NumPy.

It is code for "Mathematical Python." It is a collection of exhibit handling schedules and multi-layered cluster objects. Straight polynomial math and the utilize of arbitrary numbers are both supported by NumPy.

iii) Matplotlib

The art of displaying information through graphs, symbols, introductions, and much more is what Matplotlib is known for. Deciphering difficult information into understandable chunks of knowledge for a non-specialized audience is typically accepted as normal. One of the most outstanding Python packages utilized for Datum Perception is Matplotlib. The goal of this cross-stage system is to create two-layered charts using data from exhibits. Additionally, this provides a programming interface with items that makes it easier to do things like install plots.

iv) Seaborn

Seaborn is an enhancement for matplotlib, not a swap for it. This is on the grounds that it is based on top of matplotlib, and drawing easier plots that are as of now accessible through the namespace pyplot much of the time requires straightforwardly utilizing matplotlib capabilities. Despite the fact that Matplotlib is no doubt versatile, it very well may be trying to know which settings to utilize to make a convincing plot.

A modern Shruthi Gurudath 2990078 UI and various custom subjects to copy the matplotlib look are incorporated with Seaborn. It is firmly connected with the PyData stack and supports information structures for NumPy, Pandas, and SciPy models as well as factual calculations.

v) MLxtend

Various key AI and information mining procedures and apparatutilizes are carried out by the library known as MLxtend. The principal objective of MLxtend is to foster instruments that are extensively utilized to focus just on consistency with current AI libraries on basic and clear APIs.

A great many capabilities are implemented by MLxtend, yet some stick out. These incorporate continuous example mining calculations, stacked speculation executions for grouping and relapse, and consecutive choice element calculations. MLxtend gives various instruments that expand on Python's logical registering stack, improving its utilizefulness.

Chapter-4 PERFORMANCE ANALYSIS

4.1Customer Purchase Patterns:

4.1.1 Which day of the week the order purchase rate is the highest?

It can clearly be seen through the below graph that on Sunday, that is, Day 0, the purchase rate is the highest, and on Monday, day 2, it's the second highest. The probable reason for such an observation can be:

a. The grocery buying decision is a family making decision that is to be done on a holiday.

b. The grocery at many houses are bought at initial days of the week referring to the budget management decisions of the individual or the family

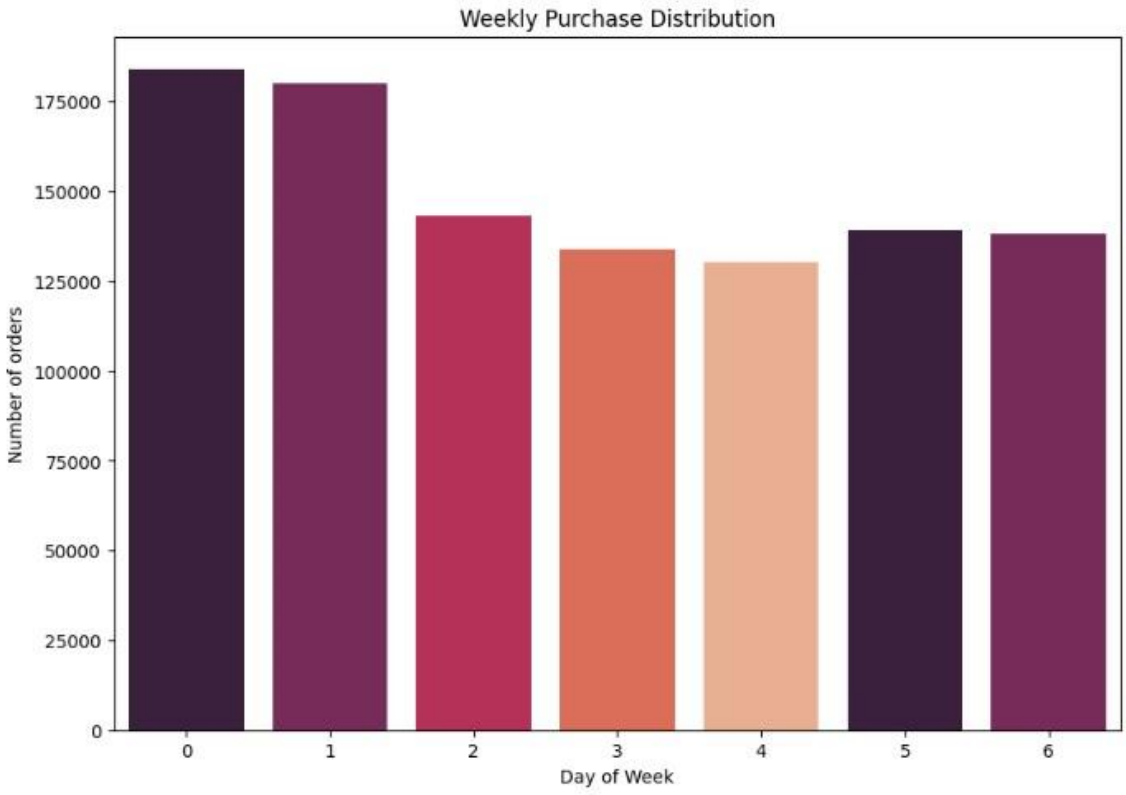


Figure 15: Highest orders on which Day of the week

4.1.2 **At which hour of the day, order purchase rate is the highest?**

It is observed that there is a hike in purchase of items or goods between 10 am and 4 pm. The highest orders are at 10am and least orders are between 3am - 4am in the middle night. Hence,

a. It can be inferred that most orders are done in office hours, that is, at work signifying that, the first thing most people do at work is buy groceries and then start their day at work.
b. Instacart can engage its shopper in this period of time, by popping notifications to trigger their thought of ordering and hence increasing sales.

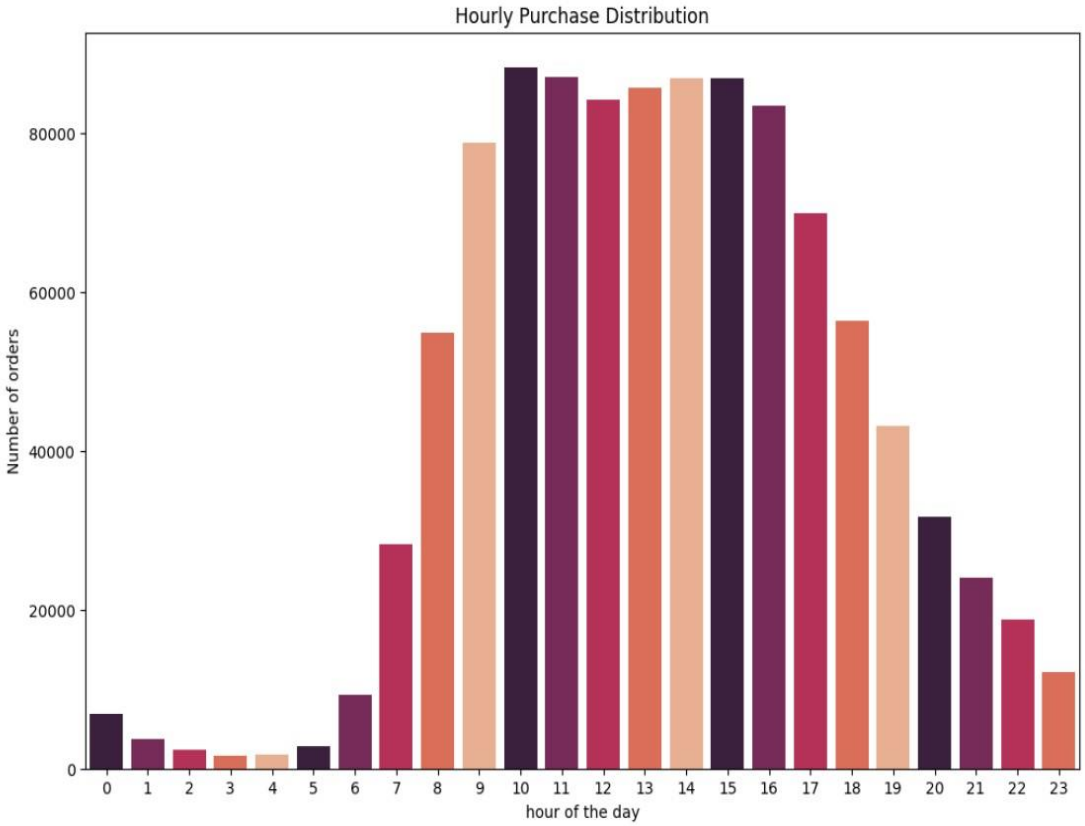


Figure 16: Hourly purchase arch

The combined heatmap for the above two observations can be seen below. The heatmap provides you with some of the important observations listed as following:

There can be promotional campaigns on Sundays and Mondays

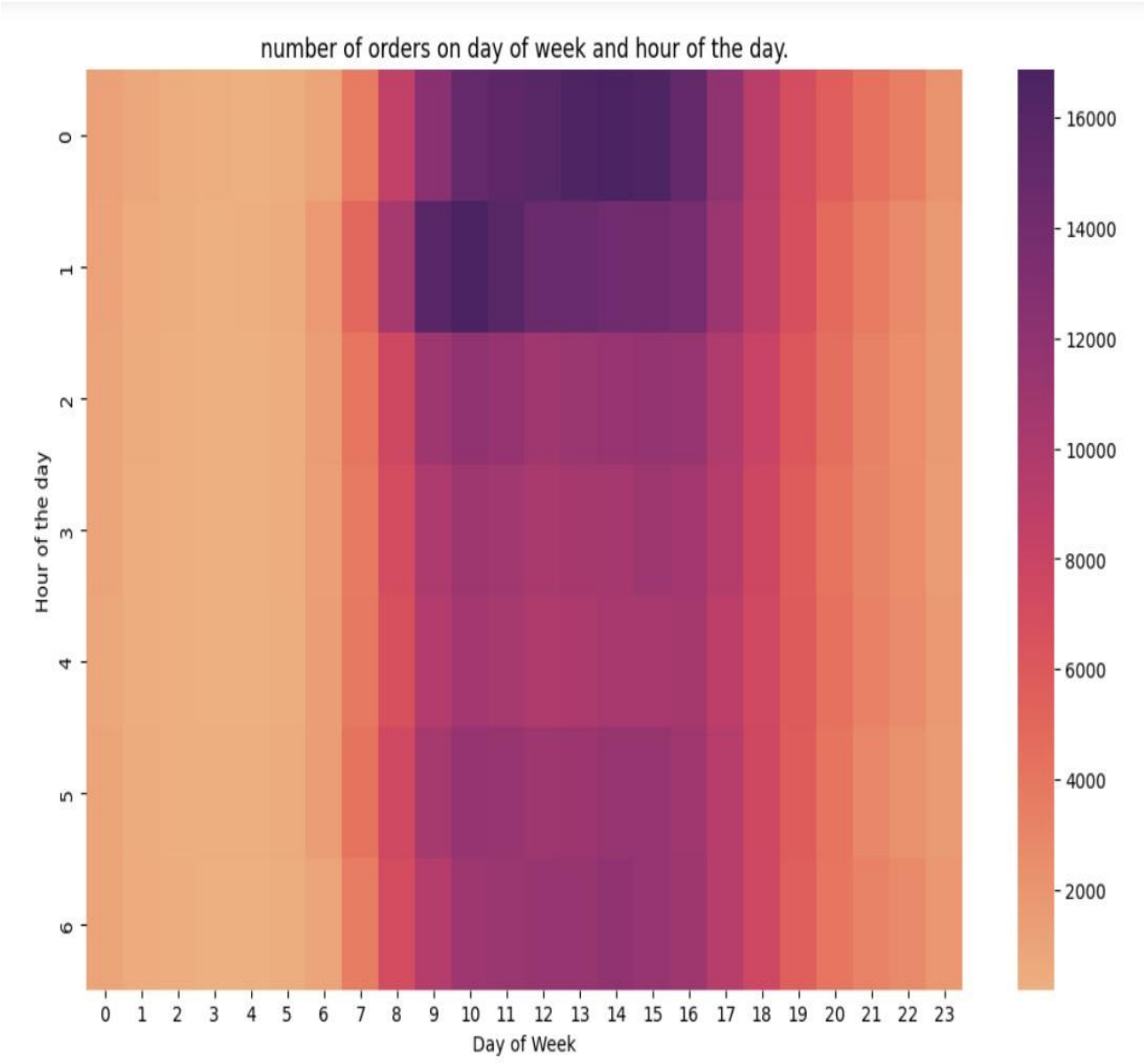


Figure 17: Combined weekly and hourly purchase arch

4.1.3 **How many shopper ordered a particular number of orders?**

It can be seen that most of the utilizers or the shopper order 4 to 10 items or goods per order. There are many less shopper who have ordered more than 40 items or goods in one order. This shows that very few utilizers buy the items or goods for fun but more utilizers buy items or goods for need, the essential need.

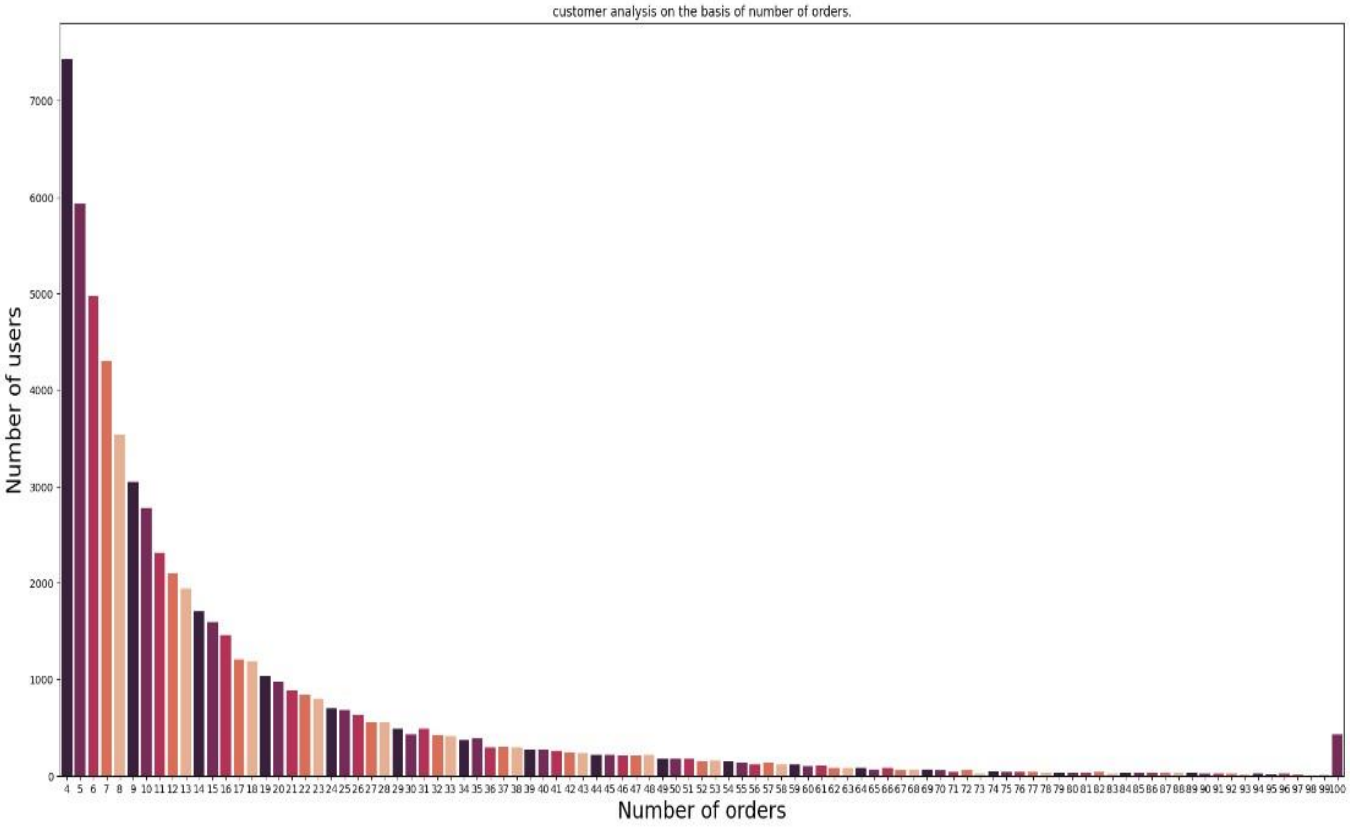


Figure 18: Number of orders by a single utilizer

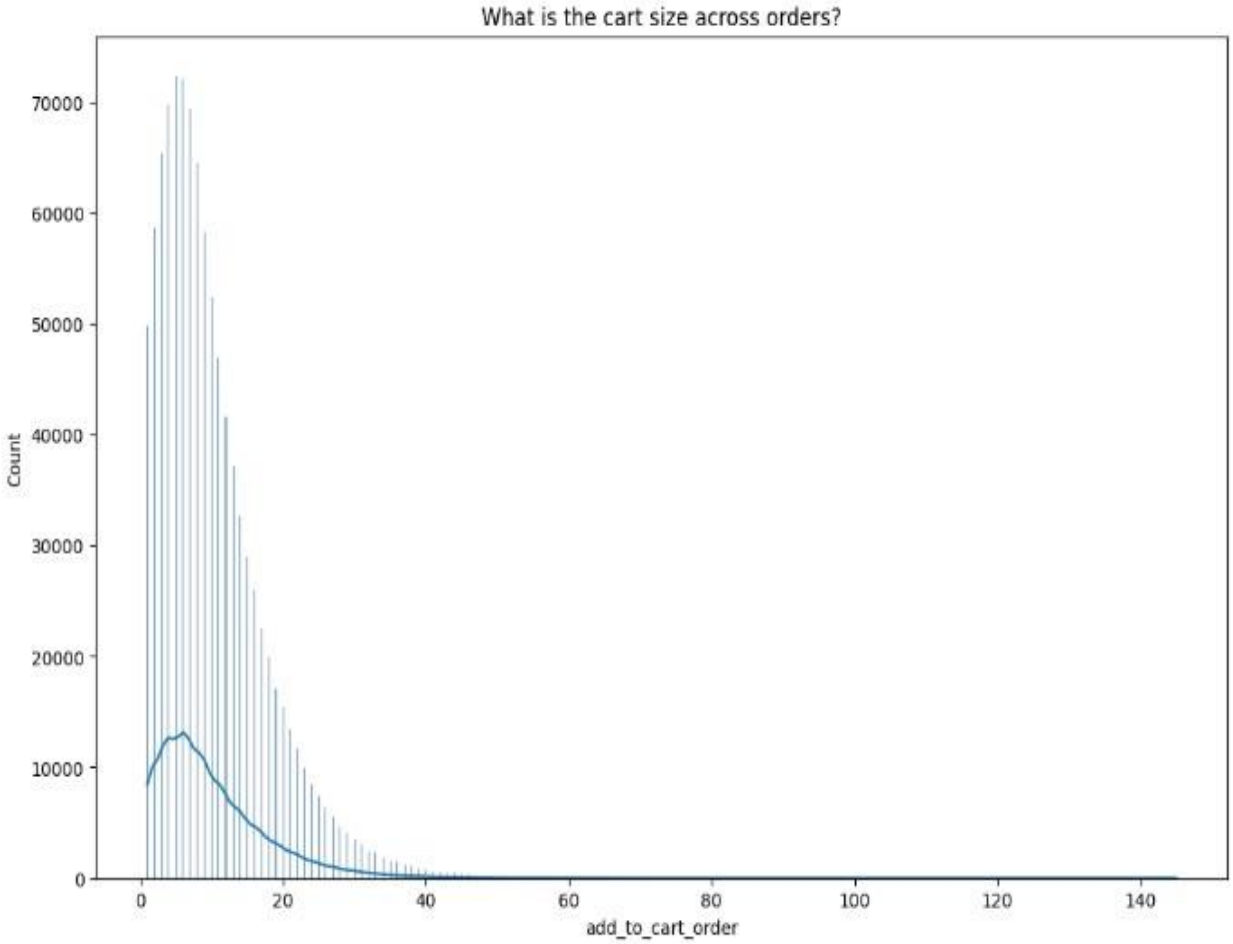


Figure 19: Very few orders more than 40

4.1.4 **How many days before has a utilizer recently ordered?**

It is observed that most utilizers order monthly that describes their monthly balance of groceries. Few shopper purchase groceries weekly as well, which signifies that there are people who buy their groceries on a weekly basis. There may be two explanations for this observation, one is there might be more consumption at a few houtilizes, and the other is that the budget is weekly distributed due to less earnings of the family.

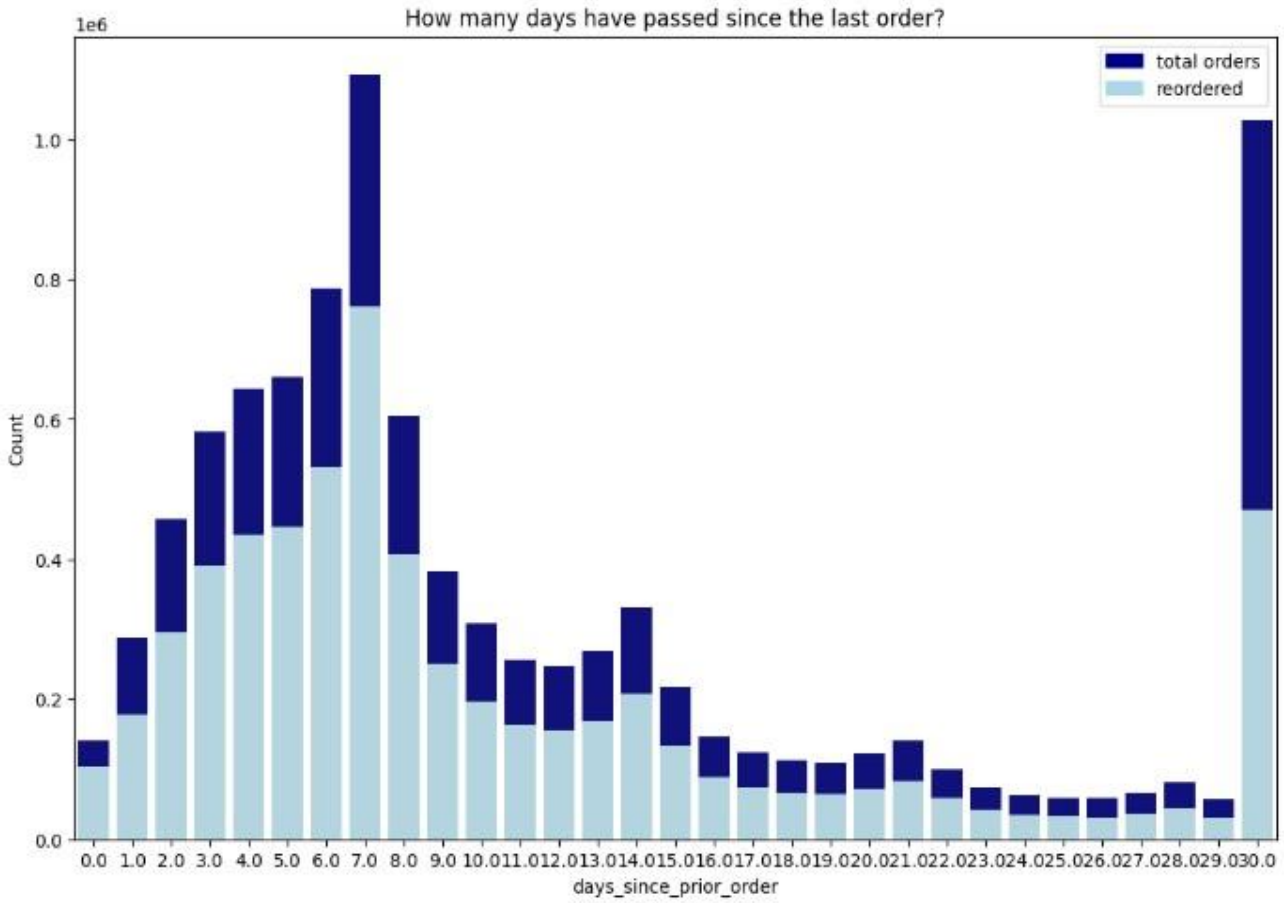
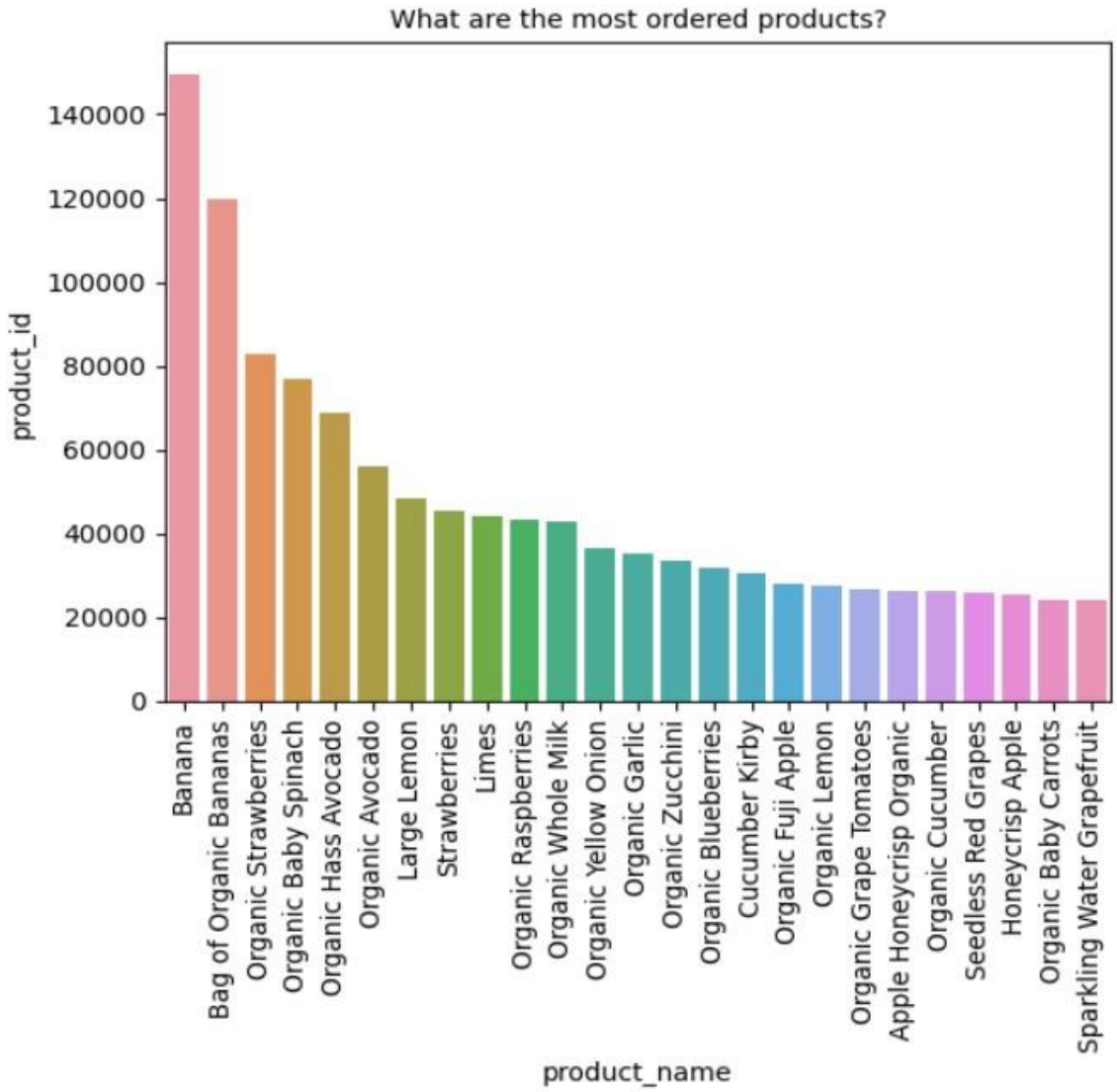


Figure 20: How often a customer purchase from the store

4.1.5 **Most ordered items or goods and most popular department.**

It is not surprising to note that the most ordered product is banana, a fruit and a healthy food supplement that is a basic requirement of every family. Organic strawberries, organic baby spinach, organic has avocados, large lemons, organic whole milk are some of the top 10 selling items or goods. These observations pretty much explain the most popular aisle, fresh fruits.




```
['Banana',
 'Bag of Organic Bananas',
 'Organic Strawberries',
 'Organic Baby Spinach',
 'Organic Hass Avocado',
 'Organic Avocado',
 'Large Lemon',
 'Strawberries',
 'Limes',
 'Organic Whole Milk']
```

Figure 21: Most ordered items or goods

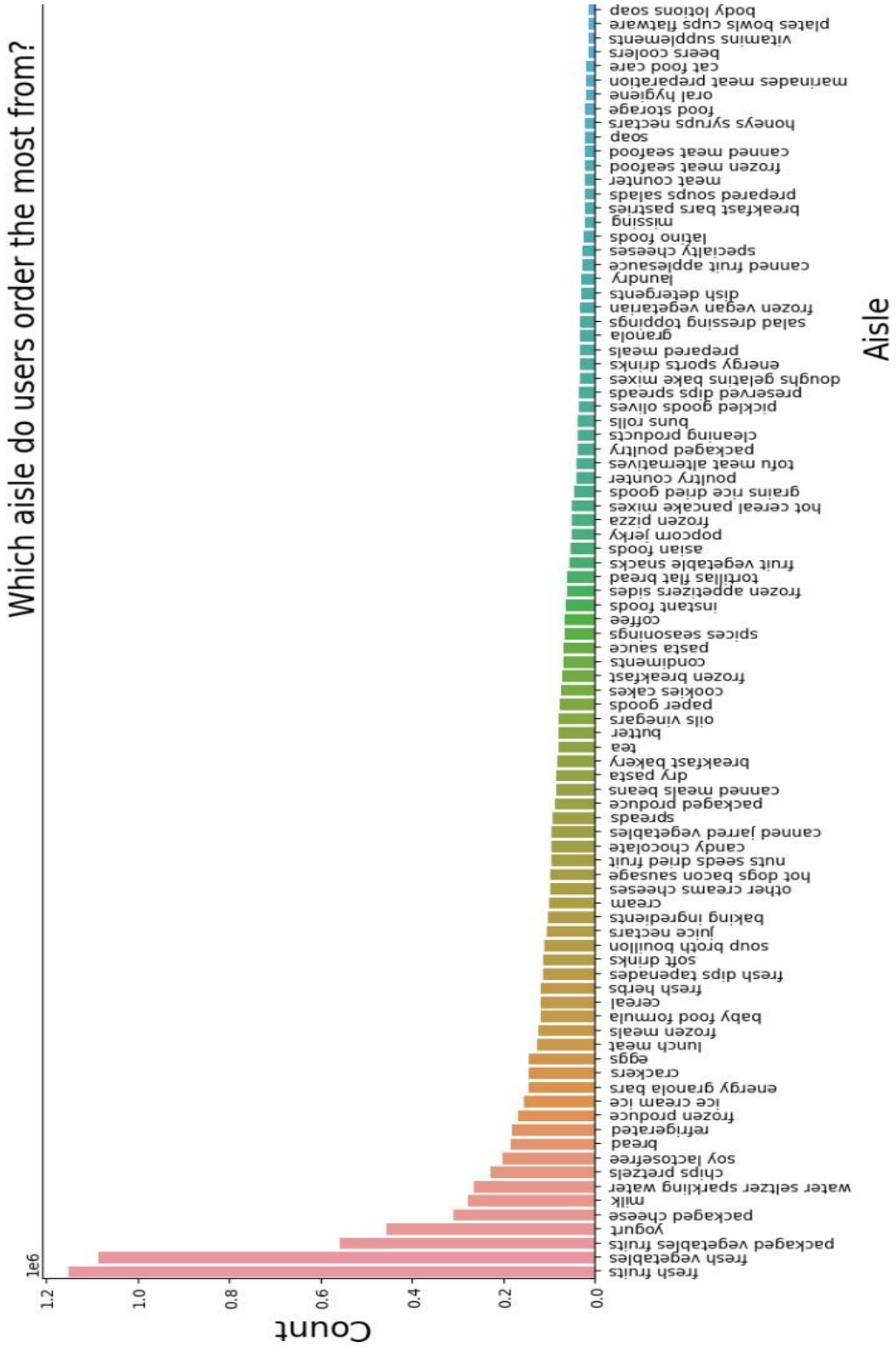


Figure 22: Most popular Aisle

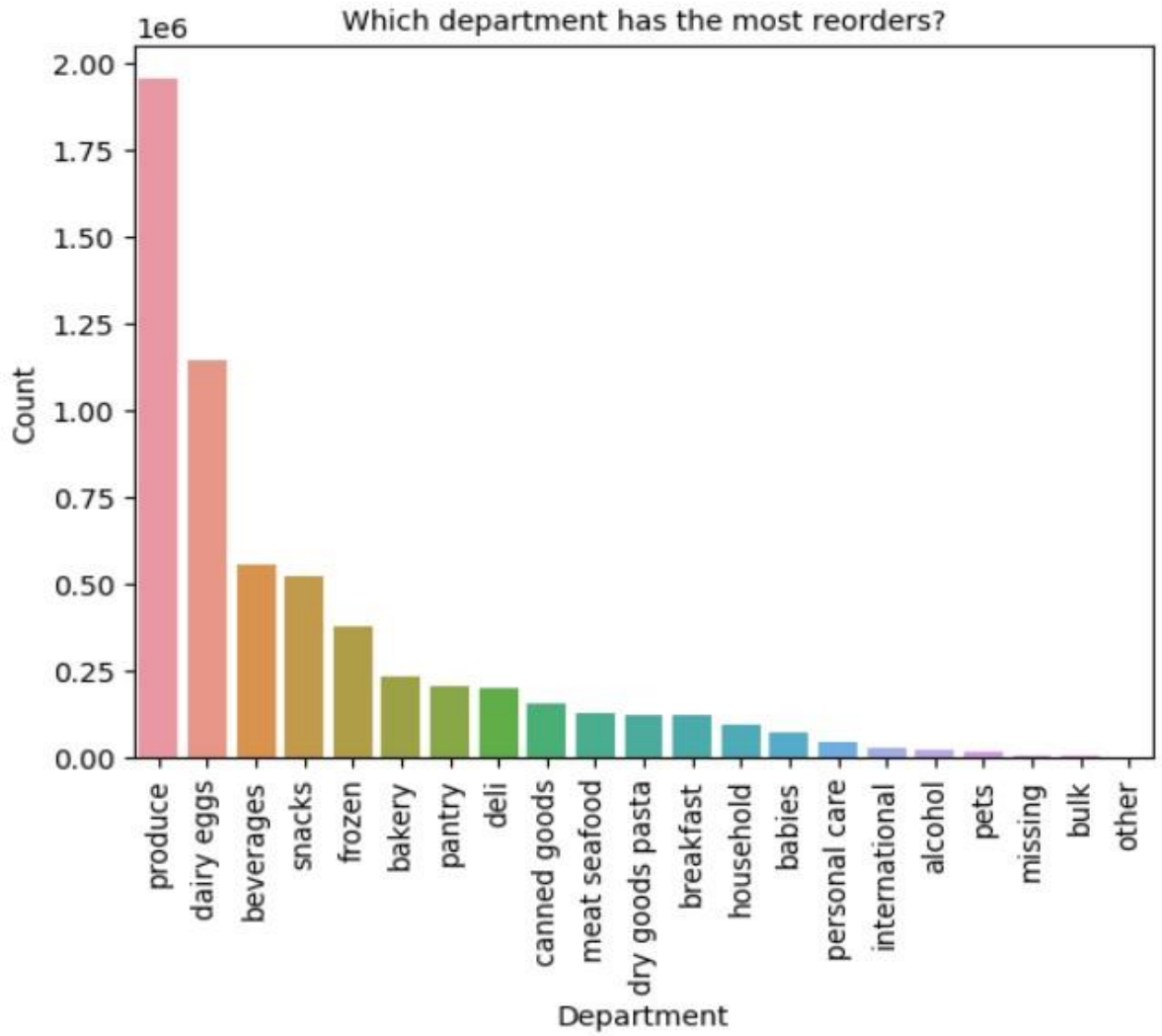


Figure 23: Most popular Department

4.2 Association Rule Mining

4.2.1 Support of persistent Items

As mentioned earlier Banana is the most ordered product with 17% support, which means, 17% of orders contains banana as a product in them.

	support	itemsets
6	0.1746	(Banana)
5	0.1360	(Bag of Organic Bananas)
64	0.0892	(Organic Strawberries)
31	0.0808	(Organic Baby Spinach)
49	0.0747	(Organic Hass Avocado)

Figure 24: Support of Itemsets

4.2.2 Difference of Association rules using Apriori and FP Growth

Authors in [1] have already found that Apriori model for mining association rules is slower than the FP growth model for the same purpose. It is also noticed that a total of 30 association rules including the frozensets are discovered from both apriori and FP growth model, giving the validation of association rules discovered. Apparently, in this study we found out that, apriori is slower and gives association rules of lower confidence when compared to FP growth model for association rules. Also, it is discovered that both models gives different rules with higher associativity. This can be explained with an example here. Rules using Apriori, the associativity of Organic Raspberries and Bag of Organic Bananas is the highest, when minimum support count is 1%.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
4	(Organic Raspberries)	(Bag of Organic Bananas)	0.0443	0.1360	0.0128	0.288939	2.124552	0.006775	1.215086
5	(Bag of Organic Bananas)	(Organic Raspberries)	0.1360	0.0443	0.0128	0.094118	2.124552	0.006775	1.054994
14	(Organic Fuji Apple)	(Banana)	0.0277	0.1746	0.0101	0.364621	2.088322	0.005264	1.299067
15	(Banana)	(Organic Fuji Apple)	0.1746	0.0277	0.0101	0.057847	2.088322	0.005264	1.031997
3	(Bag of Organic Bananas)	(Organic Hass Avocado)	0.1360	0.0747	0.0212	0.155882	2.086778	0.011041	1.096174
2	(Organic Hass Avocado)	(Bag of Organic Bananas)	0.0747	0.1360	0.0212	0.283802	2.086778	0.011041	1.206370
21	(Organic Hass Avocado)	(Organic Baby Spinach)	0.0747	0.0808	0.0114	0.152610	1.888743	0.005364	1.084743
20	(Organic Baby Spinach)	(Organic Hass Avocado)	0.0808	0.0747	0.0114	0.141089	1.888743	0.005364	1.077295
25	(Organic Hass Avocado)	(Organic Strawberries)	0.0747	0.0892	0.0120	0.160643	1.800926	0.005337	1.085116
24	(Organic Strawberries)	(Organic Hass Avocado)	0.0892	0.0747	0.0120	0.134529	1.800926	0.005337	1.069129

Figure 25: Association rules using Apriori

Whereas, Rules using FP growth, the associativity of Organic raspberries and organic strawberries is the highest.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
19	(Organic Raspberries)	(Organic Strawberries)	0.044829	0.086265	0.010484	0.233878	2.711160	0.006617	1.192675
18	(Organic Strawberries)	(Organic Raspberries)	0.086265	0.044829	0.010484	0.121538	2.711160	0.006617	1.087322
29	(Banana)	(Organic Fuji Apple)	0.170260	0.026555	0.010365	0.060877	2.292509	0.005844	1.036547
28	(Organic Fuji Apple)	(Banana)	0.026555	0.170260	0.010365	0.390322	2.292509	0.005844	1.360948
16	(Organic Raspberries)	(Bag of Organic Bananas)	0.044829	0.134676	0.013035	0.290774	2.159063	0.006998	1.220097
17	(Bag of Organic Bananas)	(Organic Raspberries)	0.134676	0.044829	0.013035	0.096788	2.159063	0.006998	1.057527
10	(Organic Hass Avocado)	(Bag of Organic Bananas)	0.074397	0.134676	0.021302	0.286333	2.126085	0.011283	1.212504
11	(Bag of Organic Bananas)	(Organic Hass Avocado)	0.134676	0.074397	0.021302	0.158174	2.126085	0.011283	1.099518
12	(Organic Strawberries)	(Organic Hass Avocado)	0.086265	0.074397	0.013180	0.152780	2.053579	0.006762	1.092518
13	(Organic Hass Avocado)	(Organic Strawberries)	0.074397	0.086265	0.013180	0.177152	2.053579	0.006762	1.110454

Figure 26: Association rules using FP Growth

4.3 Recommendation System

The choice of candidate recommendations, which are a subset of the final ones, has a significant impact on the recall of the TF-IDF method. Currently, we extend the candidate recommendations with the set difference of the product set of similar utilizer and that of target utilizer, iterating the top K similar utilizers from the most to the least similar ones. The iteration comes to an end once the size of candidate recommendations exceeds N. Finally, choose the top N items or goods from candidate recommendations based on the data file's overall prior sales, which ensures high similarity.

What would happen if target utilizer purchases were replaced by those made by the top K similar utilizers instead of candidate recommendations? The same method of choosing the top N items or goods will result in an unfavourable recall score.

Using the TF-IDF neighbourhood approach, the recommendations made by the system based on similar utilizers are

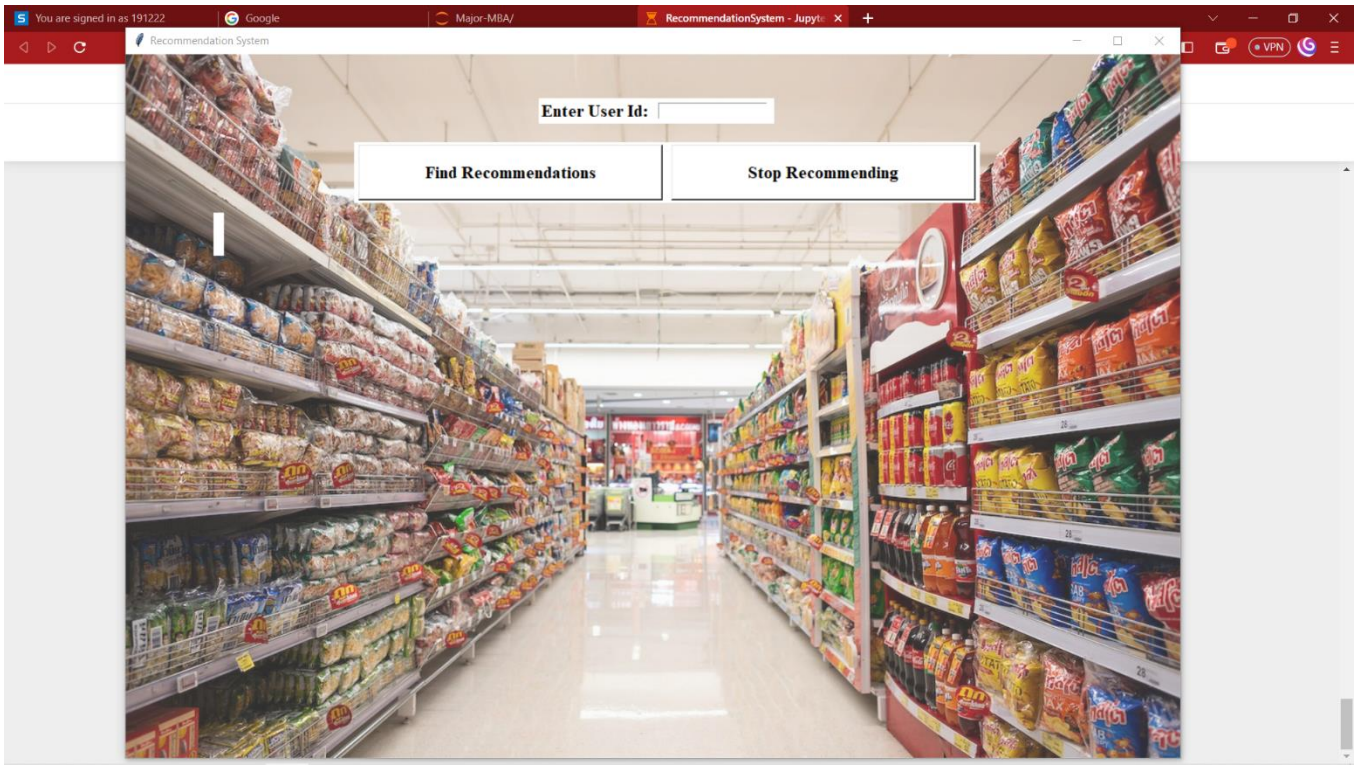


Figure : Sample Interface for recommendation engine

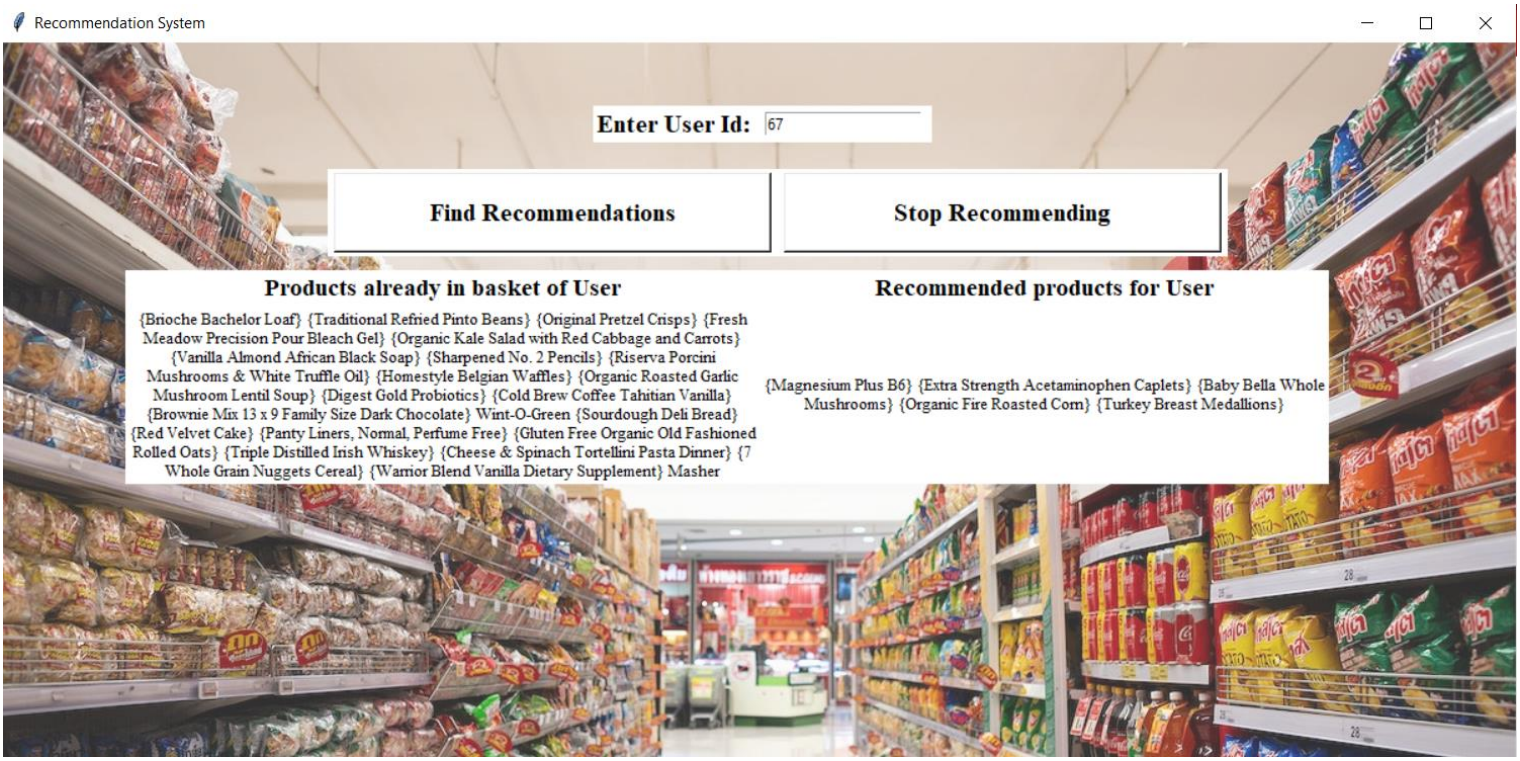


Figure 27: Recommendations for user 67

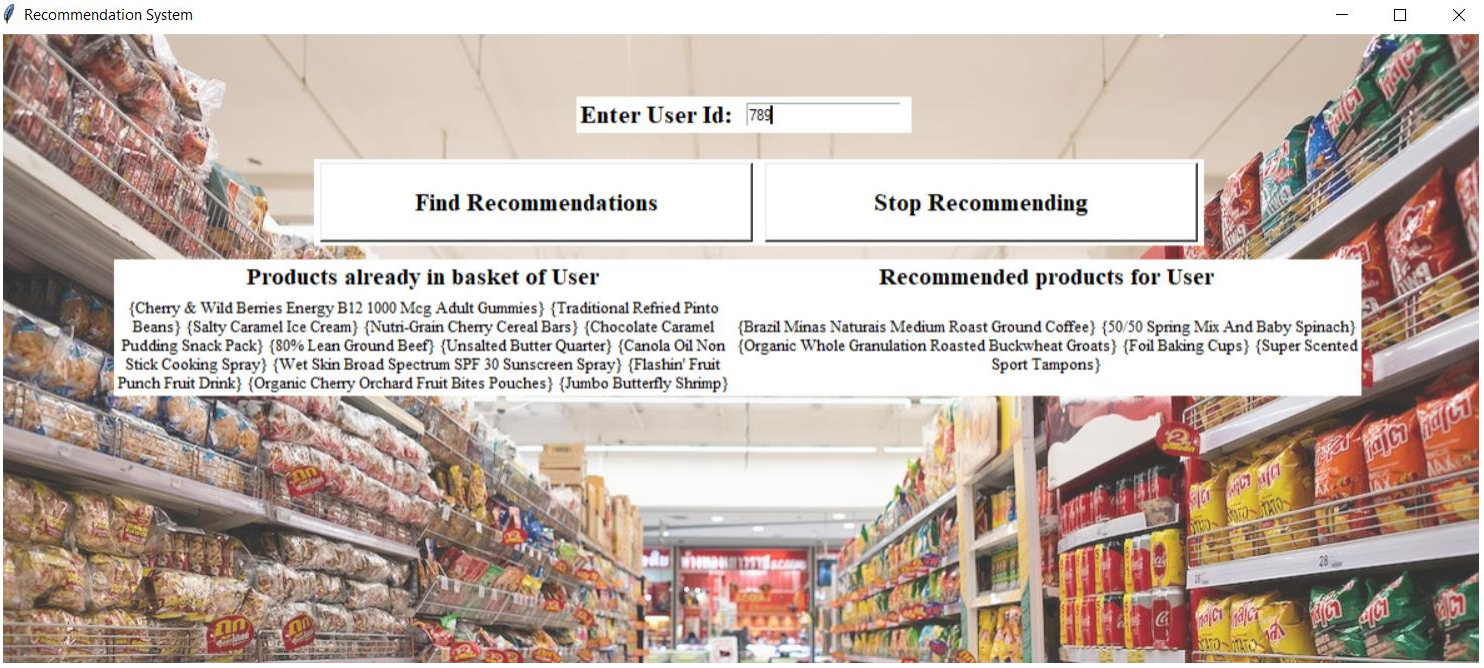


Figure 28: Recommendations for user 789

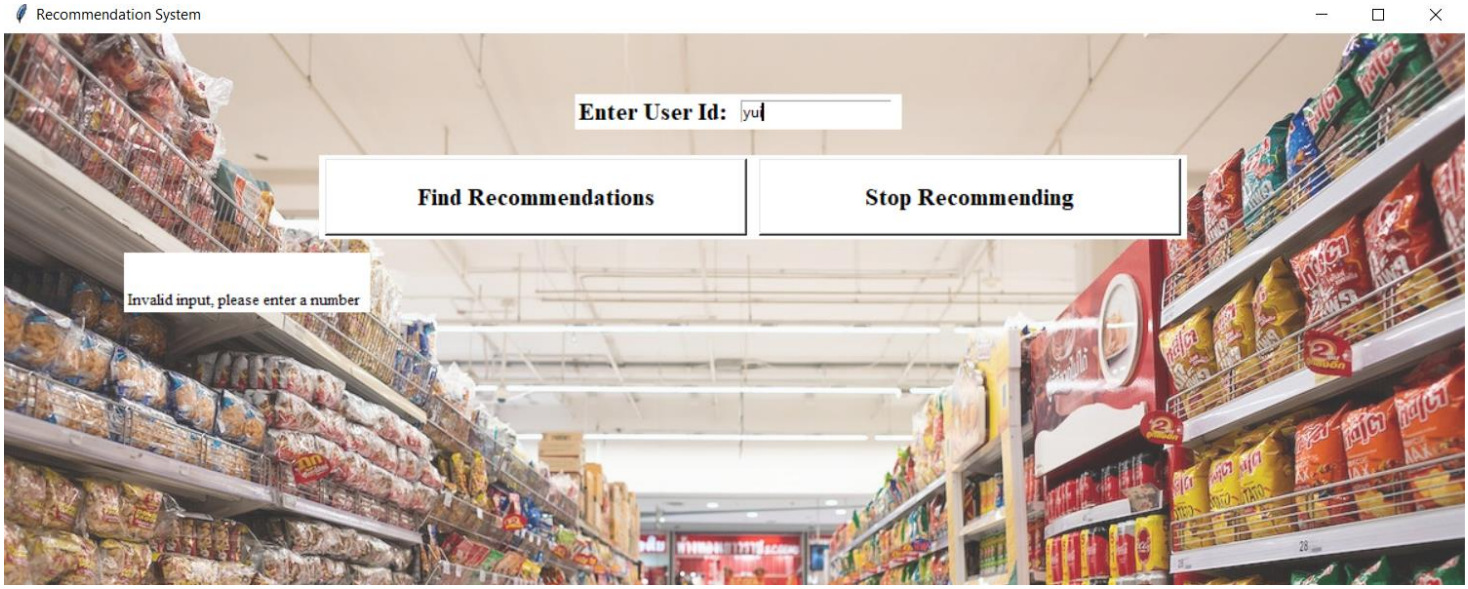


Figure 29: Incorrect user Id

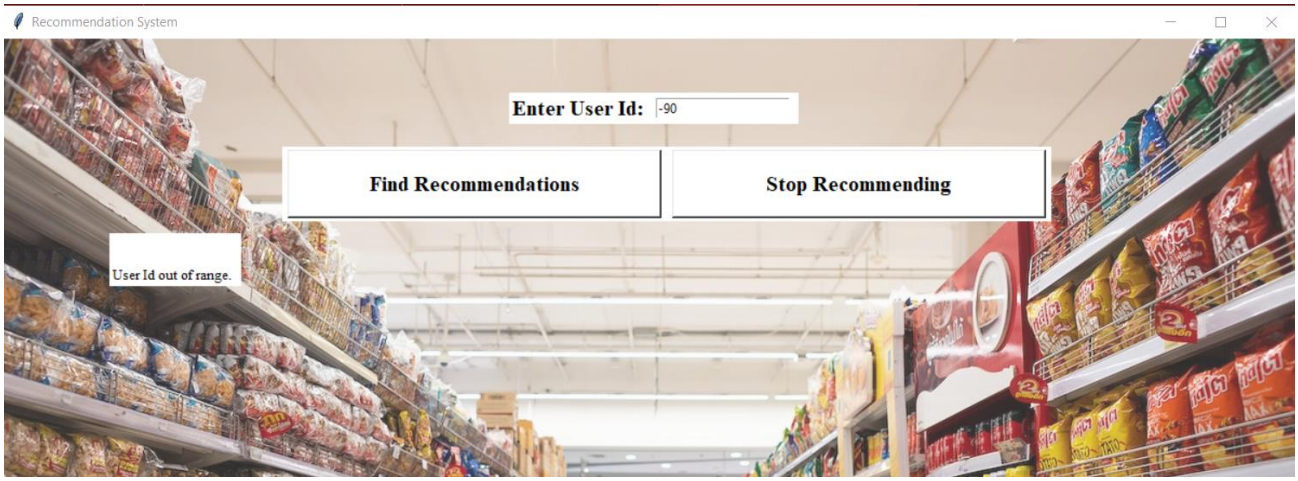


Figure 30: User Id out of range

Chapter-5 CONCLUSIONS

5.1 Conclusions

Market basket evaluation is an imaginary framework that routes from the sphere of marketing. It has been efficaciously applied in fields consisting of bioinformatics, nuclear technological know-how and immunology, geophysics. One of the motives for the growing recognition of MBA in educational fields is the usage of an inductive method to concept constructing lets in researchers to evaluate lifestyles membership rules. With that in mind, I agree with this recommendation by summarizing the whole. The system can efficiently influence marketing and sales research strategic business decisions.

5.2 Future Work

Implementing fresh and cutting-edge mining models along with with apriori, FP growth for better performance and quick results for sparse data files can boost the project. In the current strategy, we solely utilize association rules to take advantage of the collective knowledge, i.e. establishing a model by identifying similarities between purchasers' affiliations with items or goods and advising another customer to buy a similar connected item.

In upcoming work, association rules can also be utilized to take advantage of content-based resource, such as detecting product similarities and proposing items or goods based on the interest in related items or goods.

As a result of their incapacity to handle items or goods and utilizers that are new to the system, collaborative filtering techniques suffer from cold-start issues. The ideal solution for this is a hybrid strategy utilized in conjunction with content-based techniques. Therefore, gathering more information that would aid in profiling utilizers and objects would be a crucial step. Additionally, different strategies like Deep Neural Networks, Learning to Rank, and Social Recommendations could be investigated.

REFERENCES

[1] M. Hossain, A. H. M. S. Sattar, and M. K. Paul, "Market basket analysis using apriori and FP growth algorithm," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 2019, pp. 1–6.

[2] B. Gayathri, "Efficient market basket analysis based on FP-Bonsai," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (ISMAC)*, 2017, pp. 788–792.

[3] S. K. Dubey, S. Mittal, S. Chattani, and V. K. Shukla, "Comparative analysis of market basket analysis through data mining techniques," in *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 2021, pp. 239–243.

[4] A. Trnka, "Market Basket Analysis with Data Mining methods," in *2010 International Conference on Networking and Information Technology*, 2010, pp. 446–450.

[5] L. Yongmei and G. Yong, "Application in market basket research based on FP-growth algorithm," in *2009 WRI World Congress on Computer Science and Information Engineering*, 2009, vol. 4, pp. 112–115.

[6] A. H. Espejel and F. J. Cantu-Ortiz, "Data mining techniques to build A recommender system," in *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, 2021, pp. 217–221.

[7] M. Patil and T. Patil, "Apriori algorithm against fp growth algorithm: A comparative study of data mining algorithms," *SSRN Electron. J.*, 2022.

[8] B. Alam, "Implementation of FP-growth algorithm using Python," *Hands-On-Cloud*, 20-Feb-2022. [Online]. Available: https://hands-on.cloud/implementation-of-fpgrowth-algorithm-using-python/.

[9] A. Sagar, “Instacart market basket analysis : Part 1 (introduction & EDA),” *Medium*, 05-Apr-2021. [Online]. Available: https://asagar60.medium.com/instacart-marketbasket-analysis-part-1-introduction-eda-b08fd8250502.

[10] “Instacart market basket analysis,” *Kaggle.com*. [Online]. Available: https://www.kaggle.com/c/instacart-market-basket-analysis.

[11] “Instacart market basket analysis,” Kaggle.com. [Online]. Available: https://www.kaggle.com/competitions/instacart-market-basket-analysis/data.

APPENDICES

Code for Customer Purchase arches

```
master_df = pd.concat([order_products_prior,order_products_train]).sort_values(by=['order_id'])

# Merge the rest of the datasets
master_df = pd.merge(left = master_df, right = products,
                     left_on='product_id', right_on='product_id').sort_values(by=['order_id']).reset_index(drop=True)
master_df = pd.merge(left = master_df, right = aisles,
                     left_on='aisle_id', right_on='aisle_id').sort_values(by=['order_id']).reset_index(drop=True)
master_df = pd.merge(left = master_df, right = departments,
                     left_on='Department_id', right_on='Department_id').sort_values(by=['order_id']).reset_index(drop=Tru
master_df = pd.merge(left = master_df, right = orders,
                     left_on='order_id', right_on='order_id').sort_values(by=['order_id']).reset_index(drop=True)

col_order = ['user_id','order_id','product_id','aisle_id','Department_id','add_to_cart_order',
 'reordered','product_name','Aisle','Department','eval_set','order_number','order_dow','order_hour_of_day',
 'days_since_prior_order']

master_df = master_df[col_order]
master_df
```

42]:

	user_id	order_id	product_id	aisle_id	Department_id	add_to_cart_order	reordered	product_name	Aisle	Department	eval_set	order_number	order_c
0	22352	6	15873	75	17	2	0	Dryer Sheets Geranium Scent	laundry	household	prior	4	
1	22352	6	41897	101	17	3	0	Clean Day Lavender Scent Room Freshener Spray	air fresheners candles	household	prior	4	
2	22352	6	40462	31	7	1	0	Cleanse	refrigerated	beverages	prior	4	
3	3107	8	23423	43	3	1	1	Original Hawaiian Sweet Rolls	buns rolls	bakery	prior	5	
4	45082	13	36086	108	16	11	0	Philadelphia Original Cream Cheese	other creams cheeses	dairy eggs	prior	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	
517	25247	3421083	45309	92	18	2	0	Purple Carrot & blueberry Puffs	baby food formula	babies	prior	24	
518	25247	3421083	7854	117	19	1	0	Freeze Dried Mango Slices	nuts seeds dried fruit	snacks	prior	24	
								Organic Mini Sandwich					

```
import matplotlib.patches as mpatches
plt.figure(figsize=(12,8))
data1 = master_df.groupby('days_since_prior_order', as_index=False).size()
bar1 = sns.barplot(x="days_since_prior_order",  y="size", data=data1, color='darkblue')
data2 = master_df[master_df['reordered']==1].groupby('days_since_prior_order', as_index=False).size()
bar2 = sns.barplot(x="days_since_prior_order",  y="size", data=data2,  color='lightblue')
top_bar = mpatches.Patch(color='darkblue', label='total orders')
bottom_bar = mpatches.Patch(color='lightblue', label='reordered')
plt.legend(handles=[top_bar, bottom_bar])
plt.title('How many days have passed since the last order?')
plt.ylabel('Count')
plt.show()
```

```
plt.figure(figsize=(12,8))
data1 = master_df.groupby('order_dow', as_index=False).size()
bar1 = sns.barplot(x="order_dow",  y="size", data=data1, color='maroon')
data2 = master_df[master_df['reordered']==1].groupby('order_dow', as_index=False).size()
bar2 = sns.barplot(x="order_dow",  y="size", data=data2,  color='yellow')
top_bar = mpatches.Patch(color='maroon', label='total orders')
bottom_bar = mpatches.Patch(color='yellow', label='reordered')
plt.legend(handles=[top_bar, bottom_bar])
plt.title('Which days have the most orders/reorders?')
plt.ylabel('count')
plt.show()
```

```
data = master_df.groupby('product_name', as_index = False)['reordered'].sum().sort_values(by = 'reordered', ascending=False)
sns.barplot(x='product_name', y='reordered', data=data.head(25))
plt.xticks(ticks=range(len(data.head(25))), rotation=90)
plt.title('What are the most reordered products?',fontsize=10)
plt.xlabel('Product Name',fontsize=10)
plt.ylabel('Count',fontsize=10)
plt.show()
```

Preparing the data for apriori and FP growth model

```
basket = order_products.pivot_table(columns='product_name', values='reordered',
                                    index='order_id').reset_index().fillna(0).set_index('order_id')
```

```
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket = basket.applymap(encode_units)
basket.head()
```

product_name	100% Raw Coconut Water	100% Whole Wheat Bread	2% Reduced Fat Milk	Apple Honeycrisp Organic	Asparagus	Bag of Organic Bananas	Banana	Bartlett Pears	Blueberries	Boneless Skinless Chicken Breasts	...	Sparkling Natural Mineral Water	Sparkling Water Grapefruit	Spring Water
order_id														
13	0	0	0	0	0	0	0	0	0	0	...	0	0	0
14	0	0	0	0	0	0	0	0	0	0	...	0	0	0
15	0	0	0	0	0	0	0	0	0	0	...	0	0	1
22	0	0	1	0	1	0	1	0	0	0	...	0	0	0
40	0	0	0	0	0	0	0	0	0	0	...	1	0	0

5 rows × 100 columns

Finding the persistent Itemsets using Apriori and FP growth models

```
frequent_items_apriori = apriori(basket[:10000], min_support=0.01 , use_colnames=True)
frequent_items_apriori
```

```
frequent_items_fp=fpgrowth(basket, min_support=0.01, use_colnames=True)
frequent_items_fp
```

Finding the association rules

```
rules_apriori = association_rules(frequent_items_apriori, metric='lift', min_threshold=1)
rules_apriori.sort_values('lift', ascending=False)
```

```
import timeit

start = timeit.default_timer()

rules_apriori = association_rules(frequent_items_apriori, metric='lift', min_threshold=1)
rules_apriori.sort_values('lift', ascending=False)

stop = timeit.default_timer()
execution_time = stop - start

print("Program Executed in "+str(execution_time))
```

```
Program Executed in 0.004830599995329976
```

```
import timeit

start = timeit.default_timer()

rules_fp=association_rules(frequent_items_fp, metric="lift", min_threshold=1)
(rules_fp.sort_values('lift', ascending=False))

stop = timeit.default_timer()
execution_time = stop - start

print("Program Executed in "+str(execution_time))
```

```
Program Executed in 0.0047494000173173845
```

Recommendation System Helper Functions

```
def sparsity(matrix):
    """
    Given a matrix, returns its sparsity
    """
    total_size = matrix.shape[0] * matrix.shape[1]
    actual_size = matrix.size
    sparsity = (1 - (actual_size / total_size)) * 100
    return(sparsity)


def get_k_popularName(k, df):
    """
    Returns the `k` most popular products based on purchase count in the dataset
    """
    popular_products = list(df["product_name"].value_counts().head(k).index)
    return popular_products


def get_k_popularid(k, df):
    """
    Returns the `k` most popular products based on purchase count in the dataset
    """
    popular_productsid = list(df["product_id"].value_counts().head(k).index)
    return popular_productsid
```

```
prod_freq=df1['product_name'].value_counts()
prod_frequency=pd.DataFrame({'product_name':prod_freq.index, 'frequency':prod_freq.values})
prod_frequency
```

```
user_productName=df3.groupby('user_id',as_index=False).product_name.agg(list)
user_productName.head()
```

	user_id	product_name
0	1.0	[Soda, Original Beef Jerky, Pistachios, Organi...
1	2.0	[Artichoke Spinach Dip, Chipotle Beef & Pork R...
2	3.0	[Part Skim Ricotta Cheese, Organic Baby Spinac...
3	4.0	[Extra-Dry Cava, Original Orange Juice, Vegeta...
4	5.0	[Uncured Genoa Salami, Plain Whole Milk Yogurt...

```
from pandas.api.types import CategoricalDtype

user_cat = CategoricalDtype(categories=sorted(users), ordered=True)
product_cat = CategoricalDtype(categories=sorted(products), ordered=True)
user_index = user_product_prior["user_id"].astype(user_cat).cat.codes
product_index = user_product_prior["product_id"].astype(product_cat).cat.codes
```

```
coo = sparse.coo_matrix((user_product_prior["quantity"], (user_index, product_index)), shape=shape)
product_user_matrix = coo.tocsr()
```

```
sparsity(product_user_matrix)
```

```
99.86516295174103
```

```
product_user_matrix
```

```
<63100x47799 sparse matrix of type '<class 'numpy.int64'>'
        with 4066843 stored elements in Compressed Sparse Row format>
```

```
user_product_matrix = product_user_matrix.T
```

```
def tfidf_weight(tf):
    """
    Given a Term Frequency matrix
    Returns a TF-IDF weight matrix
    """

    tf_idf = coo_matrix(tf)

    # calculate IDF
    N = float(tf_idf.shape[0])
    idf = log(N / (1 + bincount(tf_idf.col)))

    # apply TF-IDF adjustment
    tf_idf.data = sqrt(tf_idf.data) * idf[tf_idf.col]
    return tf_idf

tf_idf = tfidf_weight(user_product_matrix)

# convert to Compressed Sparse Row format
tf_idf = tf_idf.tocsr()
```

```
tf_idf
```

```
<47799x63100 sparse matrix of type '<class 'numpy.float64'>'
        with 4066843 stored elements in Compressed Sparse Row format>
```

```
def generateRecommendations(target_user, cos_vec, K, N):
    """
    Given a target_user (a row), a cosine similarity vector, the number of similar users K,
          the number of products to be recommended.
    Returns product set by target user and N recommendations
    """

    # Select top K similar users
    top_K_similar_users = heapq.nlargest(K+1, range(len(cos_vec)), cos_vec.take)

    recommendations = []

    products_target_user = user_productId.loc[user_productId['user_id'] == target_user_id].product_id

    # Products of Target User
    productset_target_user = set(products_target_user.tolist()[0])

    # Fetch the preliminary recommendations
    for similar_user_id in top_K_similar_users:

        products_similar_user = user_productId.loc[user_productId['user_id'] == similar_user_id + 1].product_id

        # Recommend the products bought by the user who firstly differs in the purchase history from A.
        candidate_recommendation = set(products_similar_user.tolist()[0]) - productset_target_user

        # If similar_user_id equals to target_user_id or the candidate_recommendation is empty,
        # skip current user
        if similar_user_id == target_user_id or not candidate_recommendation: continue

        # One candidate_recommendation found, and extend it to the result
        recommendations.extend(candidate_recommendation)

        # If length of recommendations exceed N, break
        # Needed because this will ensure the recommentations are the products bought by most similar users
        if len(recommendations) > N: break

    # Pick the top N popularity (overall sales) to recommend
    h = []
    for rec in recommendations:
        heapq.heappush(h, (prod_frequencyId.loc[rec]['frequency'], rec))
        if len(h) > N:
            heapq.heappop(h)
    return productset_target_user, [item[1] for item in h]
```

```
# Selecting one user to test
target_user_id = int(input('Enter user Id : '))

# Fetch row of target user
target_user = tf_idf[target_user_id - 1]

# Calculate Cosine Similarity Vector of target user
similarities = cosine_similarity(tf_idf, target_user, False)
productset_target_user, recommendations_id = generateRecommendations(target_user, similarities.toarray(), 20, 5)

# Output the product_name of Target User's products as well as Recommendations
print('Products already in basket of User {}:'.format(target_user_id))
#print(productset_target_user)
basket_name=[]
for id in productset_target_user:
    basket_name.append(product_info.iloc[id]['name'])
print(basket_name)
print()
print('Recommended products for User {}:'.format(target_user_id))
#print(recommendations_id)
recommendations_name=[]
for id in recommendations_id:
    recommendations_name.append(product_info.iloc[id]['name'])
print(recommendations_name)
```

```
def UserId():
    try:
        num = int(entry.get())
        if num>=0 and num<=4066842:
            return num
        else:
            cart_obj.config(text='User Id out of range.')
    except ValueError:
        result = "Invalid input, please enter a number"
    cart_obj.config(text=result)

def end():
    #plot.destroy()
    root.destroy()
    print('The interface has been closed by you. Thank You for visiting!!! Come back soon.')
```

```
root = tk.Tk()
root.configure(bg='white')
root.title("Recommendation System")
root.geometry("1200x800")
root.minsize(800,800)
root.maxsize(1200,800)

from PIL import Image, ImageTk
image = Image.open("background.jpg")
alpha = 0.8  # set alpha value between 0 and 1
image.putalpha(int(255 * alpha))
photo = ImageTk.PhotoImage(image)
bg_label = tk.Label(root, image=photo)
bg_label.place(x=0, y=0, relwidth=1, relheight=1)

fnt=font.Font(family='Times', size ='15', weight='bold')
fnt1=font.Font(family='Times', size ='10')
fnt2=font.Font(family='Times', size ='14', weight='bold')
```

```
entries=tk.Frame(root, bg='white')
buttons=tk.Frame(root, bg='white')
text=tk.Frame(root, bg='white')

entry_label = tk.Label(entries, text="Enter User Id:", font=fnt, bg='white')
entry_label.grid(row=0, column=0)

entry = tk.Entry(entries)
entry.grid(row=0, column=1, padx=8, pady=3)

button = tk.Button(buttons, text="Find Recommendations", bg='white', width=28, height=2, font=fnt, command=show_list)
button.grid(row=1, column=0, padx=5, pady=3)

stop=tk.Button(buttons, text='Stop Recommending', bg='white', width=28, height=2, font=fnt, command=lambda: end())
stop.grid(row=1, column=1, padx=5, pady=3)
```

```
cart_obj=tk.Label(text, wraplength=500, font=fnt1, bg='white')
options_name=tk.Label(text, wraplength=500, font=fnt1, bg='white')
cart=tk.Label(text, wraplength=500, font=fnt2, bg='white')
options=tk.Label(text, wraplength=500, font=fnt2, bg='white')

cart.grid(row=2, column=0)
options.grid(row=2, column=1)
cart_obj.grid(row=3, column=0)
options_name.grid(row=3, column=1)

entries.place(x=470, y=50)
buttons.place(x=260, y=100)
text.place(x=100, y=180)

root.mainloop()
```

proj_may

ORIGINALITY REPORT

9%	6%	3%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	3%
2	Liu Yongmei, Guan Yong. "Application in Market Basket Research Based on FP-Growth Algorithm", 2009 WRI World Congress on Computer Science and Information Engineering, 2009 Publication	1%
3	spark.apache.org Internet Source	1%
4	Behera Gayathri. "Efficient market basket analysis based on FP-Bonsai", 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017 Publication	1%
5	Submitted to Gazi University Student Paper	1%
6	Submitted to Griffith College Dublin Student Paper	1%