Market Basket Analysis of Instacart Data

Illinois Institute of Technology
CSP571-Data Preparation and Analysis
Professor: Jawahar Panchal

Shraddha Patel (A20499171)
Computer Science
Illinois Institute of Technology
spatel174@hawk.iit.edu

Naga Surya Suresh (A20492550)
Computer Science
Illinois Institute of Technology
nlnu1@hawk.iit.edu

Nevil Jack Denis (A20474215)
Computer Science
Illinois Institute of Technology
psubramanian@hawk.iit.edu

Table of Content

Abstract

Market Basket Analysis (MBA) [1] is a process of identifying associations among entities and objects that frequently appear together, such as the collection of items in a shopper’s Basket. When used appropriately, MBA can be an effective tool for Businesses/Companies in understanding consumer behavior better and influencing it.

MBA is one of the key techniques used by large retailers that uncover associations between items by looking for combinations of items that occur together frequently in transactions[4]. In other words, it allows retailers to identify relationships between the items that people buy.

For example, if customers are buying bread, how probably are they to also buy milk in the same transaction? This information may lead to increase sales by helping the business by doing product placement, shelf arrangements, up-sell, cross-sell, and bundling opportunities.

Association Rules are widely used to analyze retail basket or transaction data, is intended to identify strong rules discovered in transaction data using some measures of interestingness, based on the concept of strong rules.

Overview

Don't you hate it when you go shopping and forget to pick up something you meant to?

To address the relationship between what items to purchase and, as a result, to increase and improve the company's sales and comprehend consumer behavior. In this project, we develop capabilities of reordering data for specific products and user preferences for products. Additionally, we construct future orders based on the users' previous order records. For Example, everyone enjoys an Apple, so the metrics for each product should reflect the quality of a product getting reordered on its own merits. The reorder metrics for Mr. A should account for this preference as well as the reordering measures that are unique to him because Mr. A like some unusual meal that is rarely enjoyed by anyone else. Other metrics based on orders might include ordering patterns, preferred times (day of the week/hour of the day), etc.

For Market basket analysis we are using support, confidence, and lift to understand the association rules of the products and items

Objective

The goal of this project is to identify patterns in consumer purchase behavior on Instacart and suggest product combinations that might be included in various promotions. This project also aims to predict which previously purchased products will be in a user's future order by using anonymized data on customer orders over time.

Specific Questions

- Which products will an Instacart consumer purchase again?
- What Frequency of order number amongst the customers/users?
- which products will be in a user's next order?
- What Day of the week and which hour customer placed the Order?

Data Preparation

Data Properties

The data that we used in this project are collected from Kaggle, Instacart Market Basket Dataset is found on Simplified Instacart Market Basket Dataset | Kaggle. The data set is a relational group of files that tracks the orders that customers place over time. An anonymous sample of more than 3 million grocery orders from more than 200,000 Instacart users make up the data set. Each user receives between 4 and 100 of their order details, including the order of the things they purchased in each order, the day and week it was placed, and the amount of time between orders.

Data Definitions

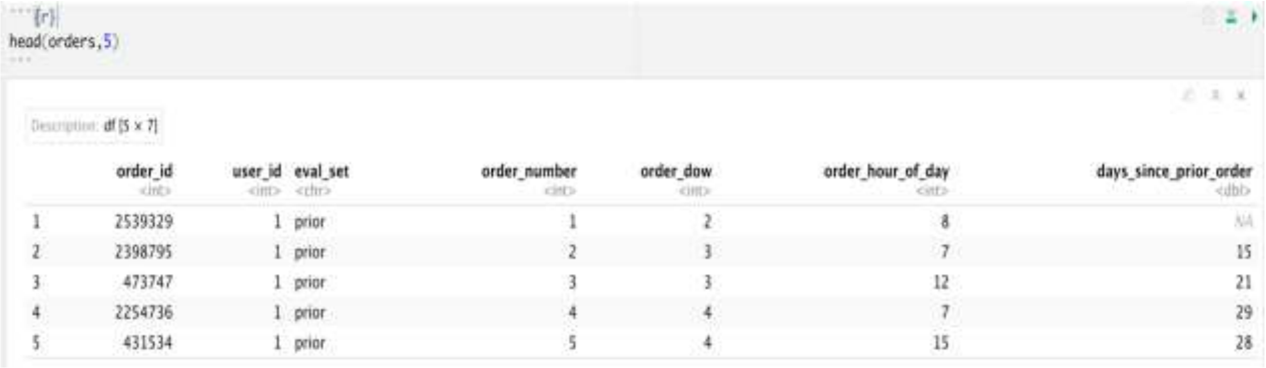
We have collected a detailed dataset of all kinds of data related to Instacart Market Basket. The description of the data and its columns/features in the dataset is mentioned below. Each entity (customer, product, order, aisle, etc.) has an associated unique id and its related data.

Orders (3.4m rows, 206k users)

Column definition

- order_id : order identifier
- user_id : customer identifier
- eval_set: which evaluation set this order belongs in.
    - The eval set has three values: train, test and prior.
- order_number : the order sequence number for this user (1 = first, n = nth)
- order_dow : the day of the week the order was placed on
- order_hour_of_day : the hour of the day the order was placed on
- days_since_prior : days since the last order, capped at 30 (with NAs for order_number = 1)

Data view



~~~
```{r}
head(orders,5)
```
~~~

Description: df [5 × 7]

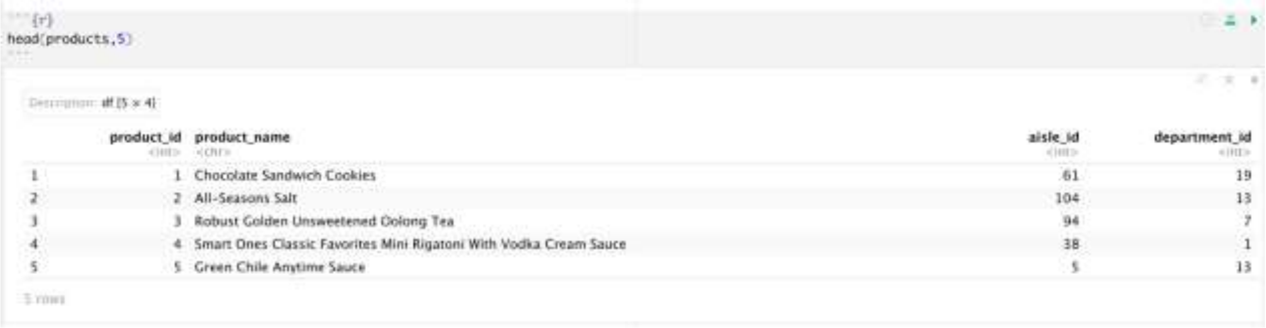
	order_id <int>	user_id <int>	eval_set <chr>	order_number <int>	order_dow <int>	order_hour_of_day <int>	days_since_prior_order <dbl>
1	2539329	1	prior	1	2	8	NA
2	2398795	1	prior	2	3	7	15
3	473747	1	prior	3	3	12	21
4	2254736	1	prior	4	4	7	29
5	431534	1	prior	5	4	15	28

Products (50k rows)

Column Definition

- product_id : product identifier
- product_name : name of the product
- aisle_id : foreign key
- department_id : foreign key

Data View



~~~
```{r}
head(products,5)
```
~~~

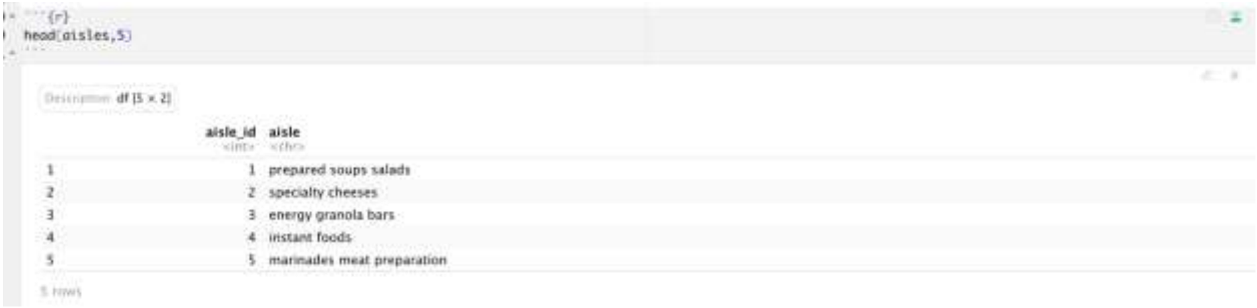
```
Description: df [5 × 4]

  product_id product_name                                              aisle_id department_id
       <int> <chr>                                                        <int>         <int>
1          1 Chocolate Sandwich Cookies                                        61            19
2          2 All-Seasons Salt                                                 104            13
3          3 Robust Golden Unsweetened Oolong Tea                              94             7
4          4 Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce 38             1
5          5 Green Chile Anytime Sauce                                          5            13
5 rows
```

Aisles (134 rows)

Column Definition

- aisle_id : aisle identifier
- aisle: the name of the aisle

Data View



~~~
```{r}
head(aisles,5)
```
~~~

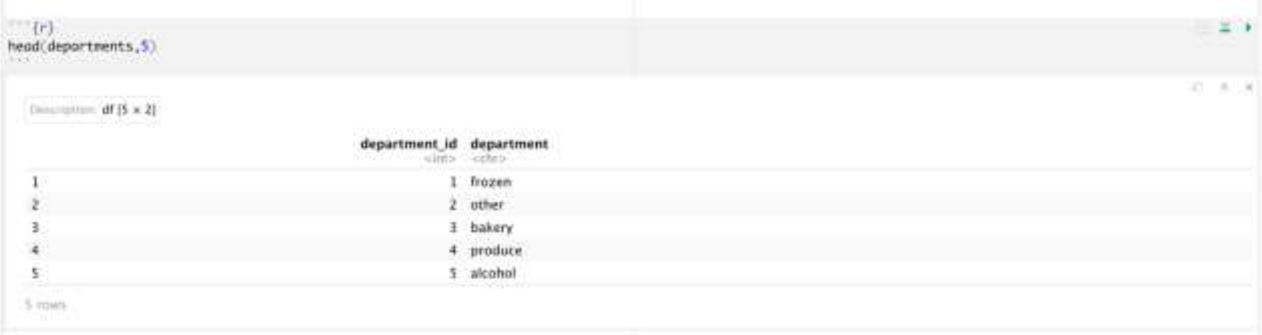
```
Description: df [5 × 2]

  aisle_id aisle
     <int> <chr>
1        1 prepared soups salads
2        2 specialty cheeses
3        3 energy granola bars
4        4 instant foods
5        5 marinades meat preparation
5 rows
```

Departments (21 rows)

Column Definition

- department_id : department identifier
- department: the name of the department

Data View



~~~
```{r}
head(departments,5)
```
~~~

Description: df [5 × 2]

	department_id <int>	department <chr>
1	1	frozen
2	2	other
3	3	bakery
4	4	produce
5	5	alcohol

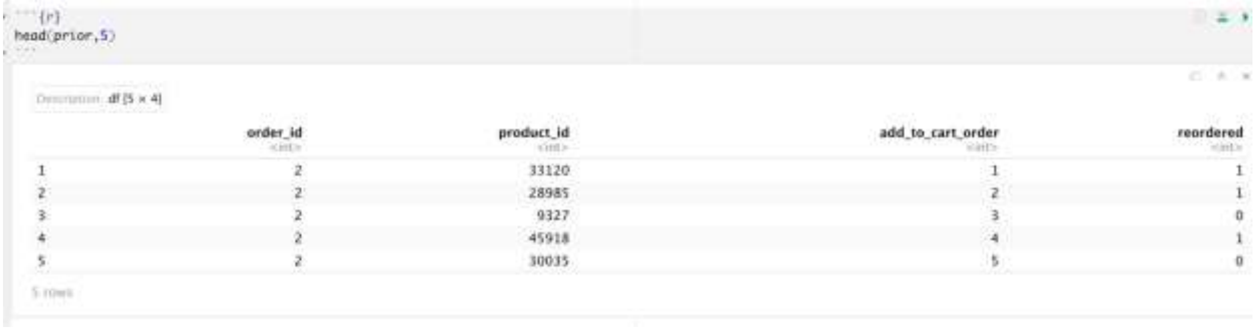
5 rows

Prior (30m+ rows)

Column Definition

- order_id : foreign key
- product_id : foreign key
- add_to_cart_order : order in which each product was added to cart
- reordered: 1 if this product has been ordered by this user in the past, 0 otherwise

Data View



~~~
```{r}
head(prior,5)
```
~~~

Description: df [5 × 4]

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

5 rows

As we can see, orders have all the relevant data for the particular order id, including the person who made the purchase, the date that it was made, the number of days since the last order, and more.

For each user, 4 and 100 of their orders are given, with the sequence of products purchased in each order. In this dataset, as previously said, a customer's 4 to 100 orders are given, and we need to forecast the things that will be ordered again. Therefore, the user's most recent order has been extracted and divided into train and test sets.

All the prior order information of the customer is present in order_products_prior dataset. We can also note that there is a column in orders data file called eval_set which tells us as to which of the three datasets (prior, train or test) the given row belongs to.

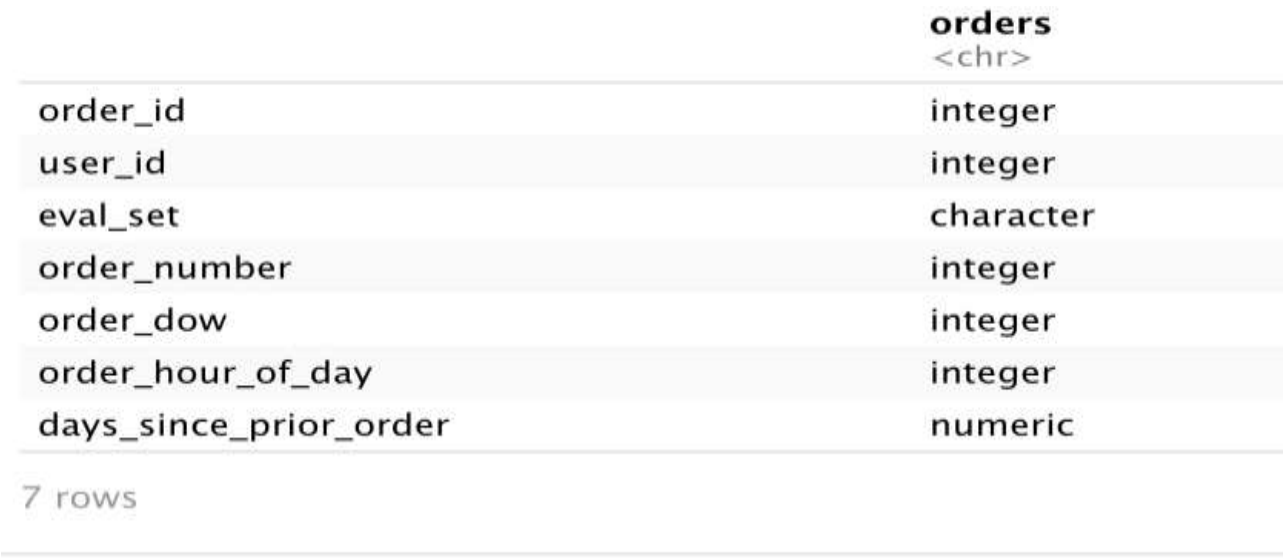
Prior and train dataset which has same column contains more thorough details on the products that were purchased in the specified order as well as the status of any subsequent orders.

Our dataset from Kaggle contains both training and test data information. The training dataset consists of 750,880 rows and the test dataset has 187,719 rows.

Data Source: Instacart Market Basket Analysis | Kaggle

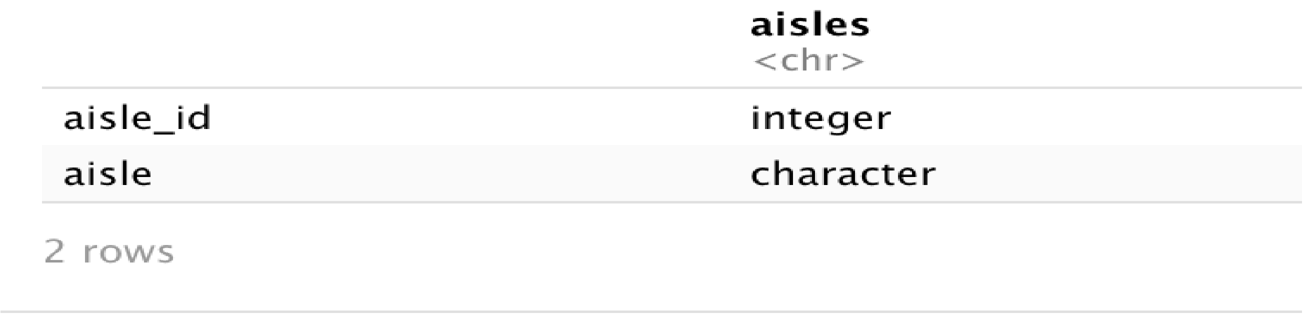
Data Processing

We perform various exploratory data analyses to understand the data. For better processing of data, we convert various character variables to Factors and numeric values to numeric [6]. The factor conversion is done on orders, products, aisles, and department data sets. So, the final data set types of each of the data set is as below



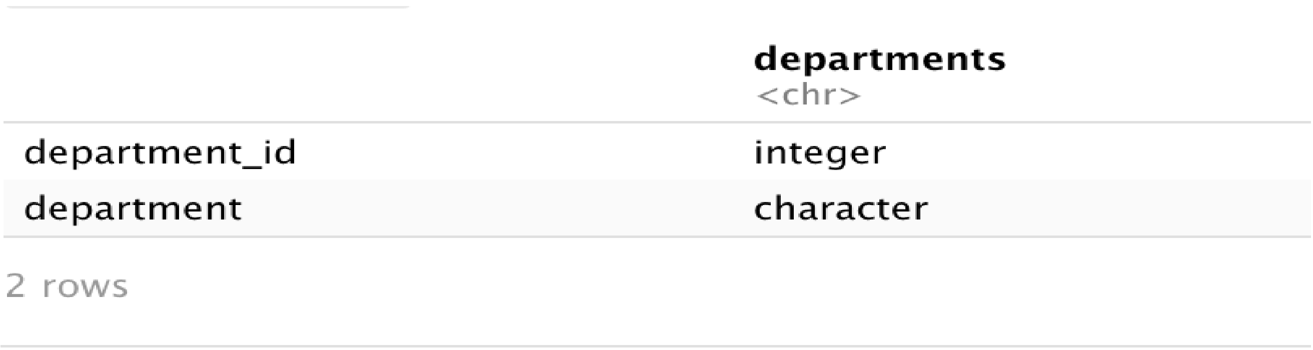
	orders <chr>
order_id	integer
user_id	integer
eval_set	character
order_number	integer
order_dow	integer
order_hour_of_day	integer
days_since_prior_order	numeric

7 rows



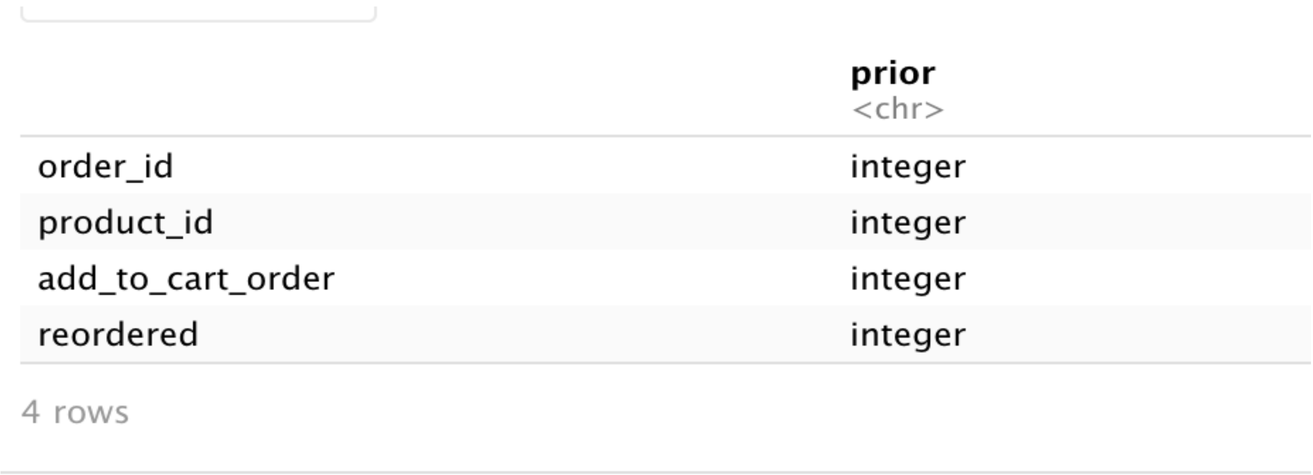
	aisles <chr>
aisle_id	integer
aisle	character

2 rows



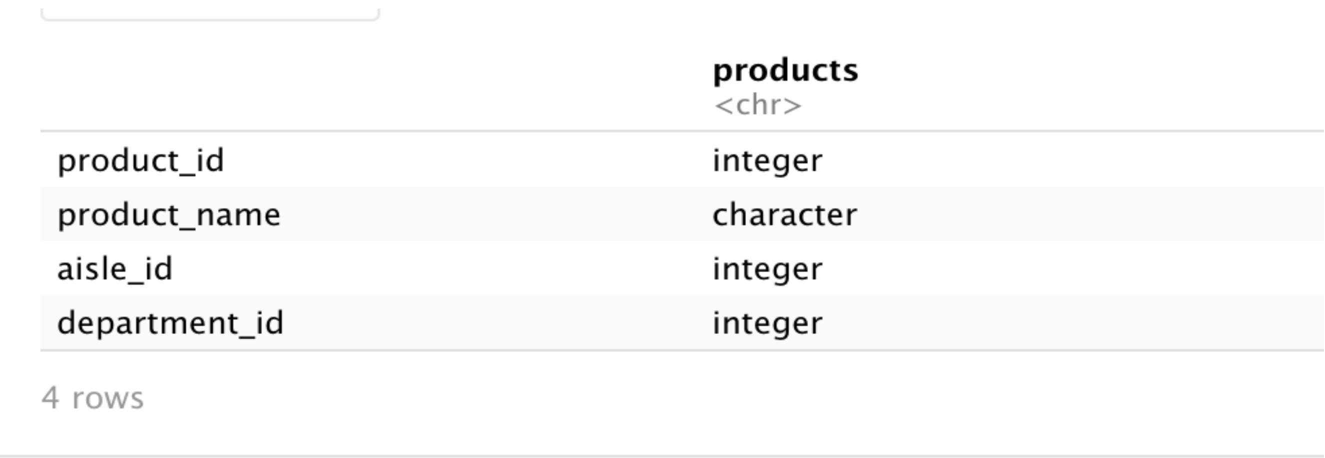
	departments <chr>
department_id	integer
department	character

2 rows



	prior <chr>
order_id	integer
product_id	integer
add_to_cart_order	integer
reordered	integer

4 rows

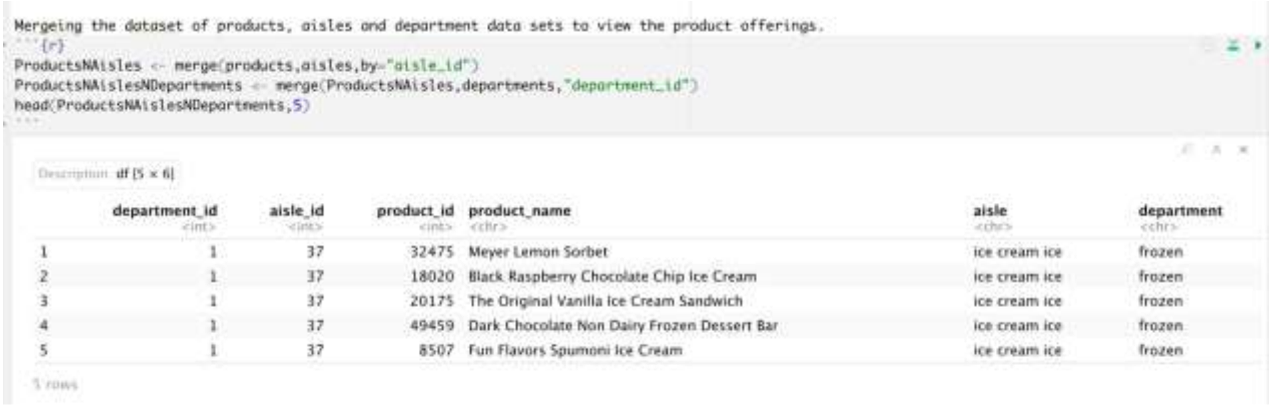


	products <chr>
product_id	integer
product_name	character
aisle_id	integer
department_id	integer

4 rows

Exploratory Data Analysis

We cleaned and enhanced the dataset from its concept to show several various aspects. To start off we are merging the datasets of products, Aisles and Departments data to find the exact product offerings. After merging the data below is the results. The Merged products, Aisles and Departments data has 49688 Rows and 6 Columns.



Mergeing the dataset of products, aisles and department data sets to view the product offerings.

~~~
```{r}
ProductsNAisles <- merge(products,aisles,by="aisle_id")
ProductsNAislesNDepartments <- merge(ProductsNAisles,departments,"department_id")
head(ProductsNAislesNDepartments,5)
```
~~~

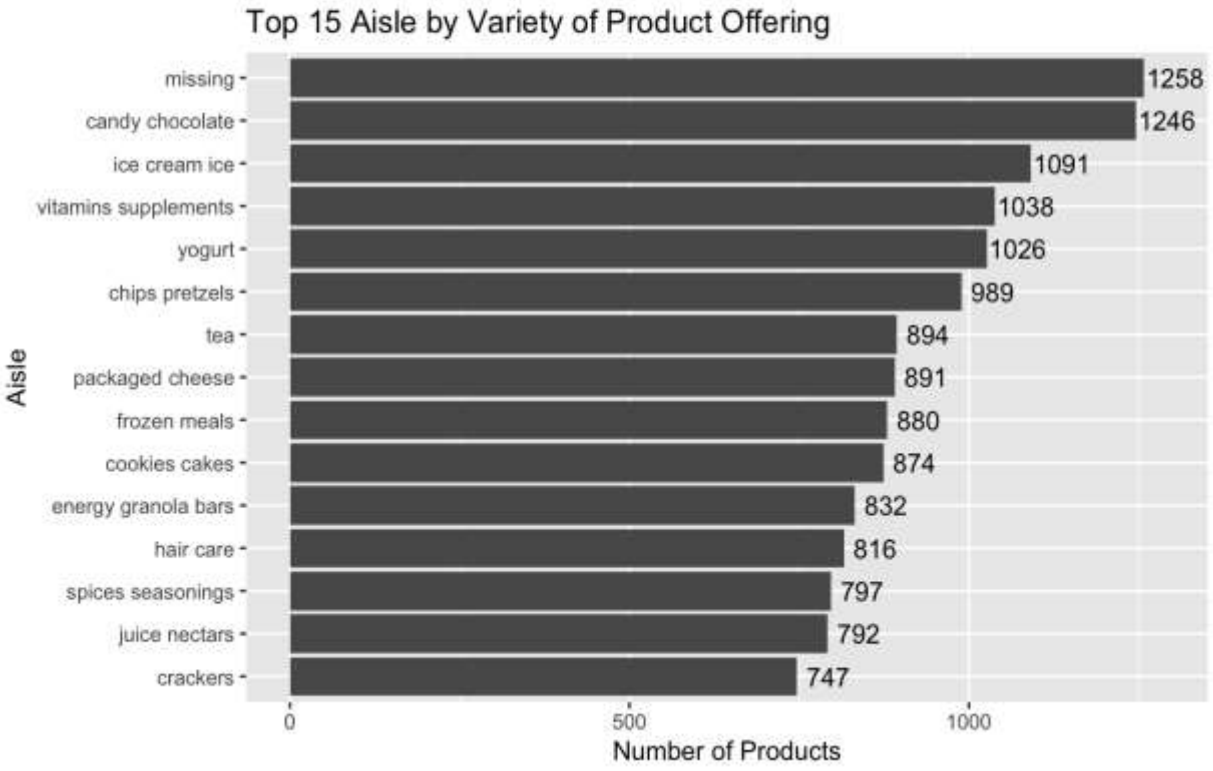
Description: df [5 × 6]

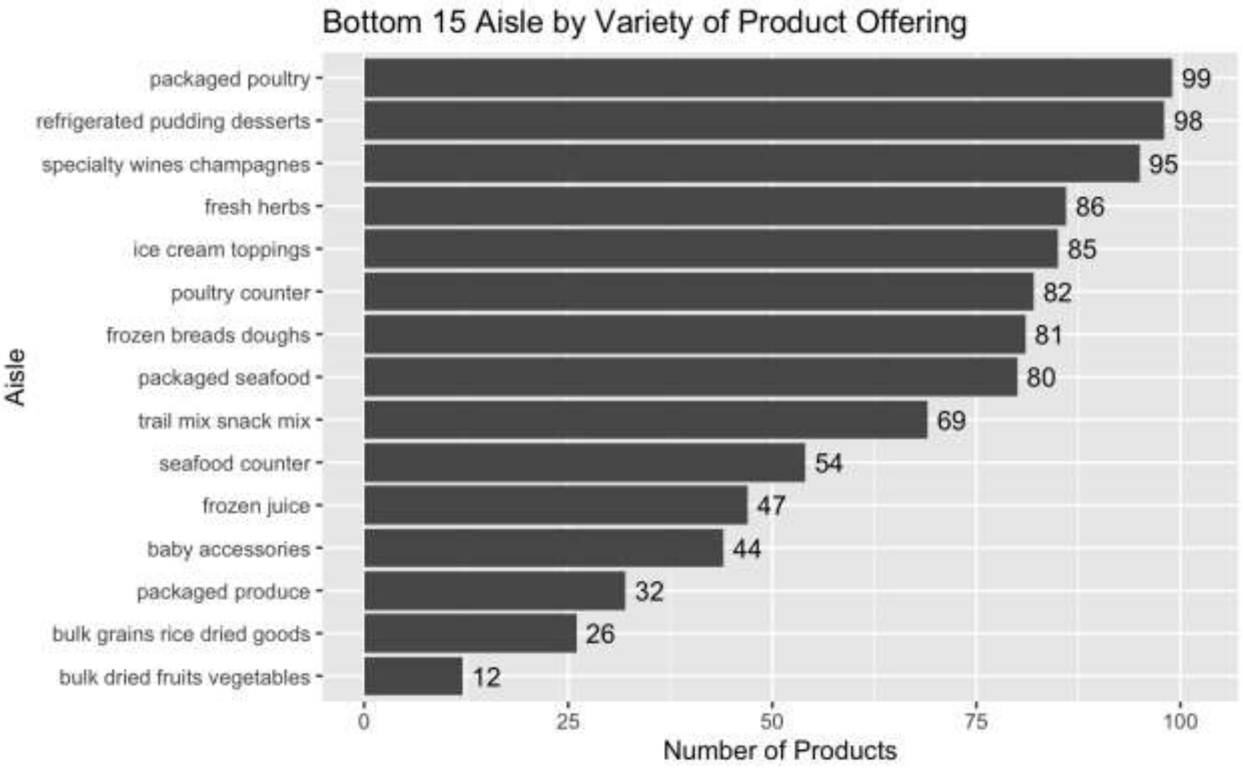
	department_id <int>	aisle_id <int>	product_id <int>	product_name <chr>	aisle <chr>	department <chr>
1	1	37	32475	Meyer Lemon Sorbet	ice cream ice	frozen
2	1	37	18020	Black Raspberry Chocolate Chip Ice Cream	ice cream ice	frozen
3	1	37	20175	The Original Vanilla Ice Cream Sandwich	ice cream ice	frozen
4	1	37	49459	Dark Chocolate Non Dairy Frozen Dessert Bar	ice cream ice	frozen
5	1	37	8507	Fun Flavors Spumoni Ice Cream	ice cream ice	frozen

5 rows

We further explore the data to find the product offerings of the Instacart data.

Top 15 and Bottom 15 Aisle by Variety of Product Offering





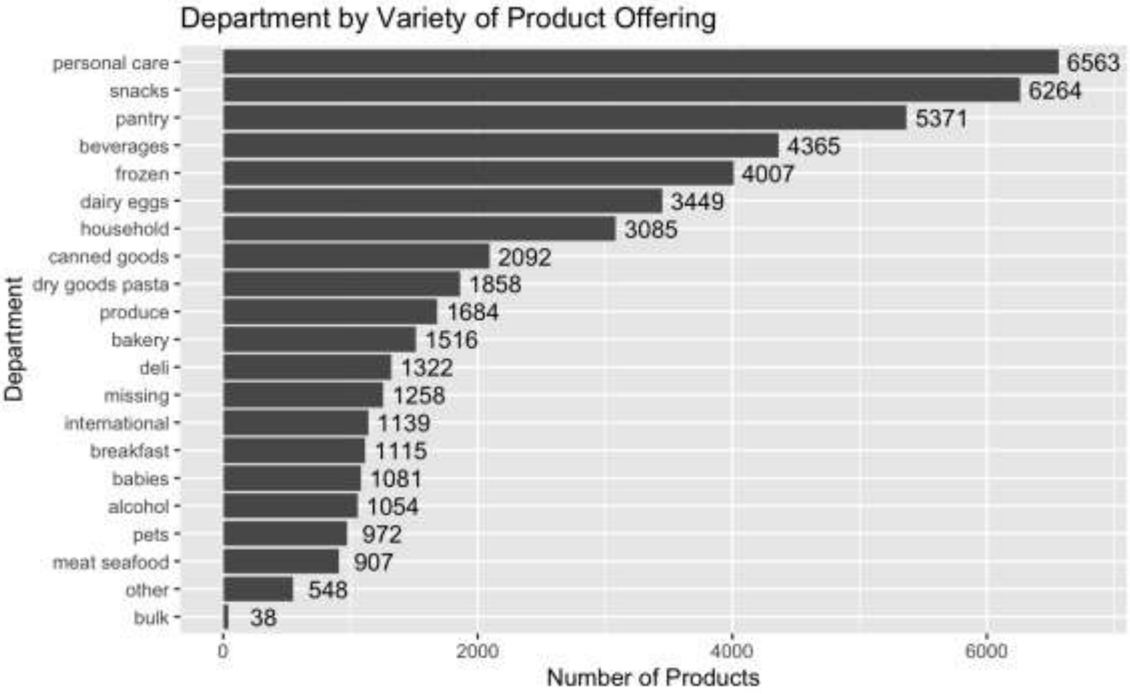
The aisles with the most diversity of products are the ones with sweets, chocolate, and ice cream, whereas the aisles with the least variety are those with trash bag liners, frozen desserts, and Indian meals.

In the first 15 Aisle we can see a variety of products that Instacart offers. Not every product is full in the aisles. There are products in the aisle with different availability. Also, there are some missing items in the aisle which means we don’t know where the product is located or the product_id does not match the aisle_id.

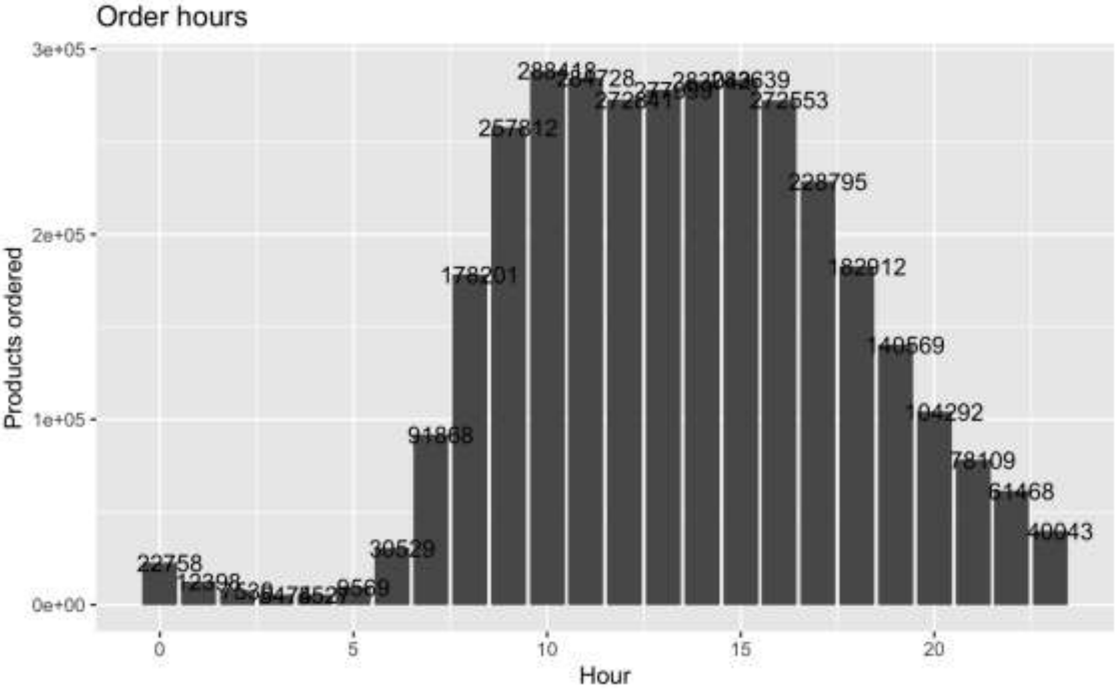
In case of bottom 15 Aisle, we can see a variety of products that Instacart offers. These aisles show the availability of each product. It shows that bulk products like dried fruits, vegetables and dries grain rice are most ordered and purchased than the wines and poultry products. No item is mis-matched in this set of data so there are no missing items.

Department by Variety of Product Offering

The above graph shows the different Departments that are available in the Instacart application. The departments that are in bulk are ordered and purchased most by the customers. This maybe because of the price. Also, it shows that personal care and snacks are the least selected departments in Instacart.

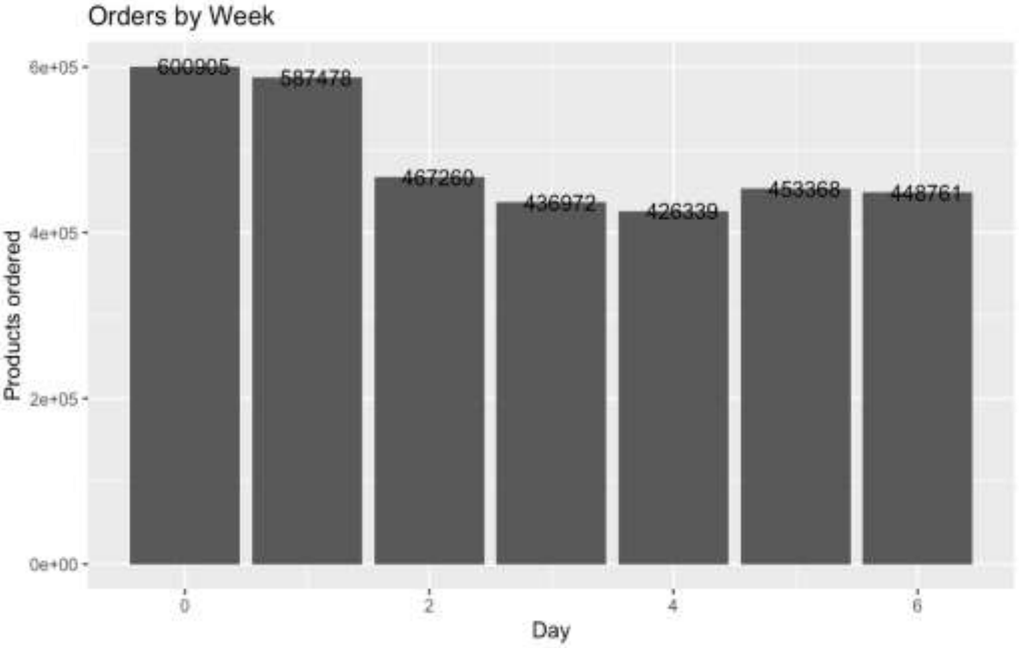


Orders by hour



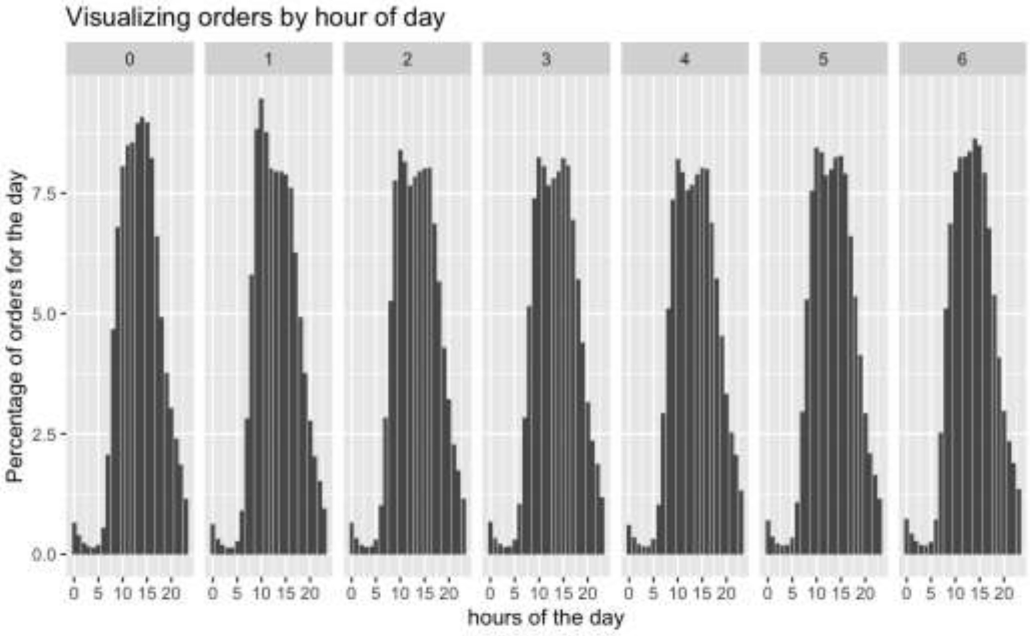
Next, we analyze the products ordered based on the hours. In this analysis we are going to compare the number of products that are ordered per hour by customers in Instacart. It clearly shows that for the first 5 to 6 hours the number of products ordered are very less. The products ordered between 7 to 20 hours are more and then decreases after 20 hours. This explains that middle of the day is more active than the opening and closing of the shop

Orders by Week



The above resulted graph shows products that are ordered by week. It shows 7 days in a week and products ordered in that week. From the result it is clear that start of the week the product ordered are more and by the middle of the week the ordering decreases and gets stable by the end of the week.

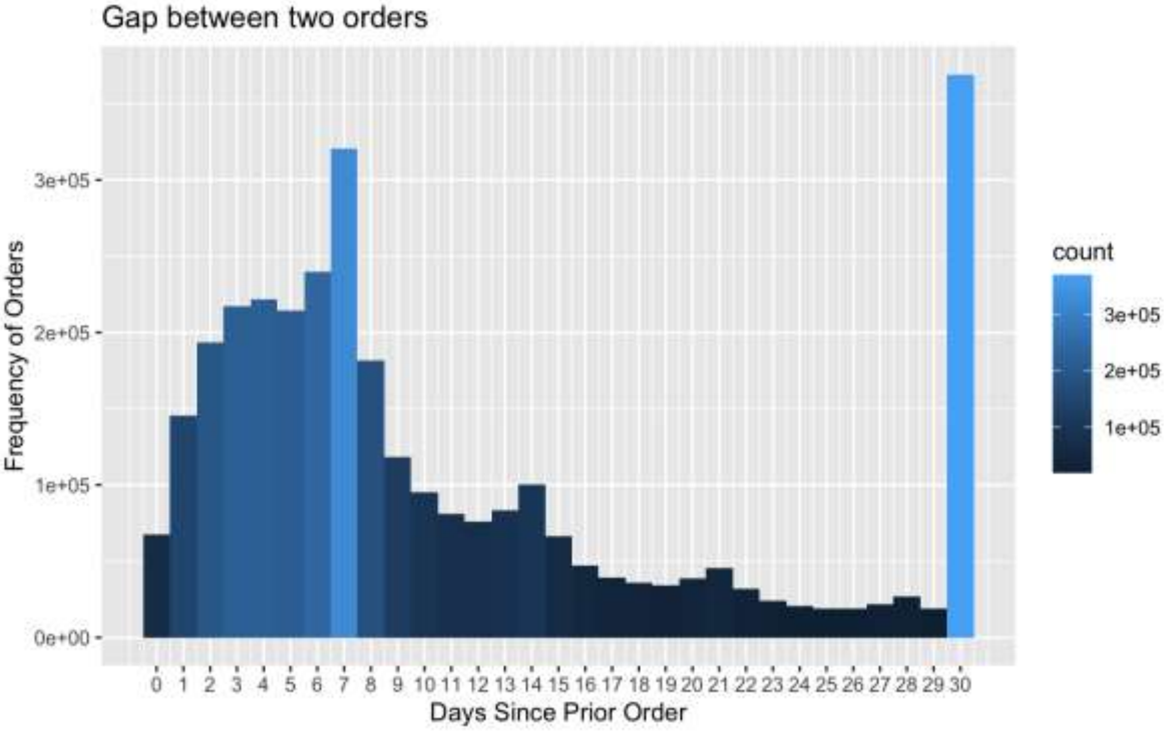
Orders by Every day Every hour



The next analysis is between products ordered by every hour of the day. The above graph shows the product order percentage vs hours in a day. Each hour has a different order pattens. The percentage of products ordered by the middle of the hour reaches maximum at each hour. By this we can analyze the peak ordering time in an hour.

Days Since Prior Order Analysis

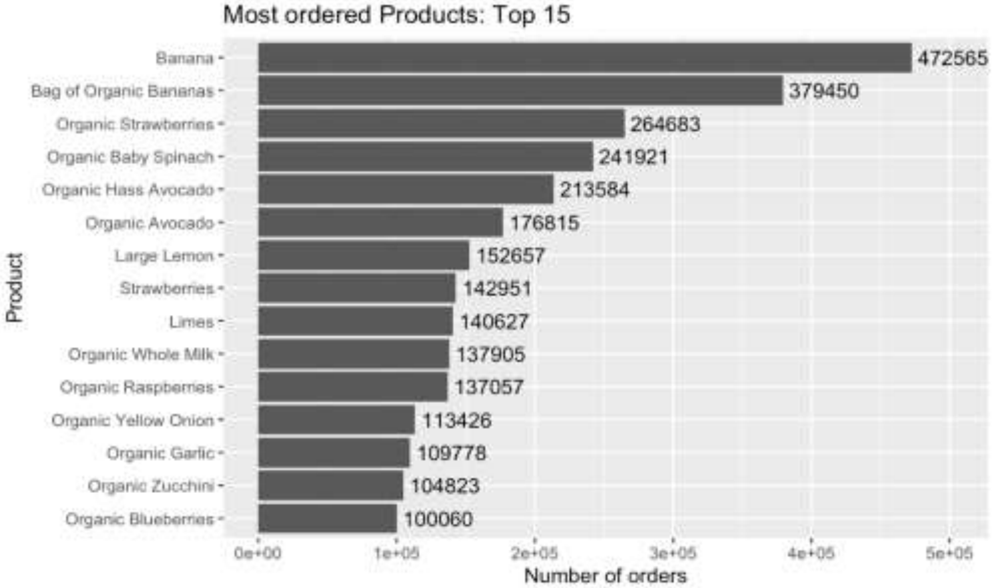
For each user, the space between two orders is provided to us. Two sorts of persons are revealed as we plot it! One places a new order every month, the other every week. This is based on the 30th day and 7th day peaks.



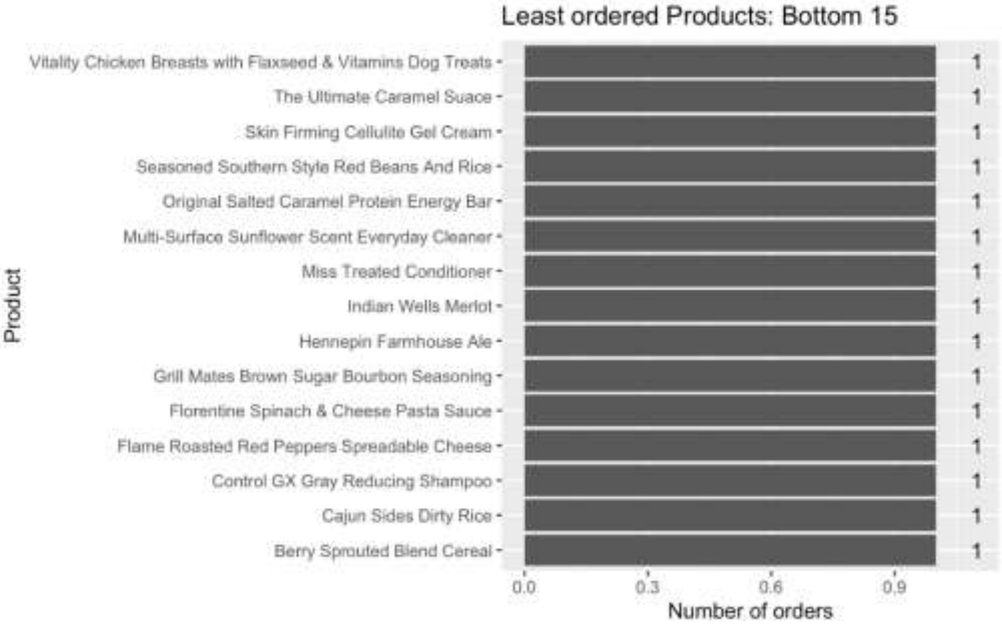
People reorder on average 11% monthly and 9% weekly. This demonstrates that there are some people who restock their food every month and others who restock them weekly. The frequency of NA denotes the total number of distinct users and their initial order. There is a continuous spike in orders from day 1 to day 6, shows that some people are frequent buyers with short window of restocking.

Prior Table Analysis

We now will perform prior table analysis to see the top products ordered prior and the least ordered products prior.



From the graph we can see the result that the most ordered product is Banana and Organic Banana which tops in the top 15 products. The graph represents the history of orders by all the users and their most preferred product. We can predict their next purchase or order by looking the history of their order using prior table.



The above graph shows the top 15 least ordered products from the prior table. This is obtained by analyzing the history of products ordered by the customers. It clearly shows that the frequency of the above products ordered by the customers are less. This is obtained by analyzing the prior table.

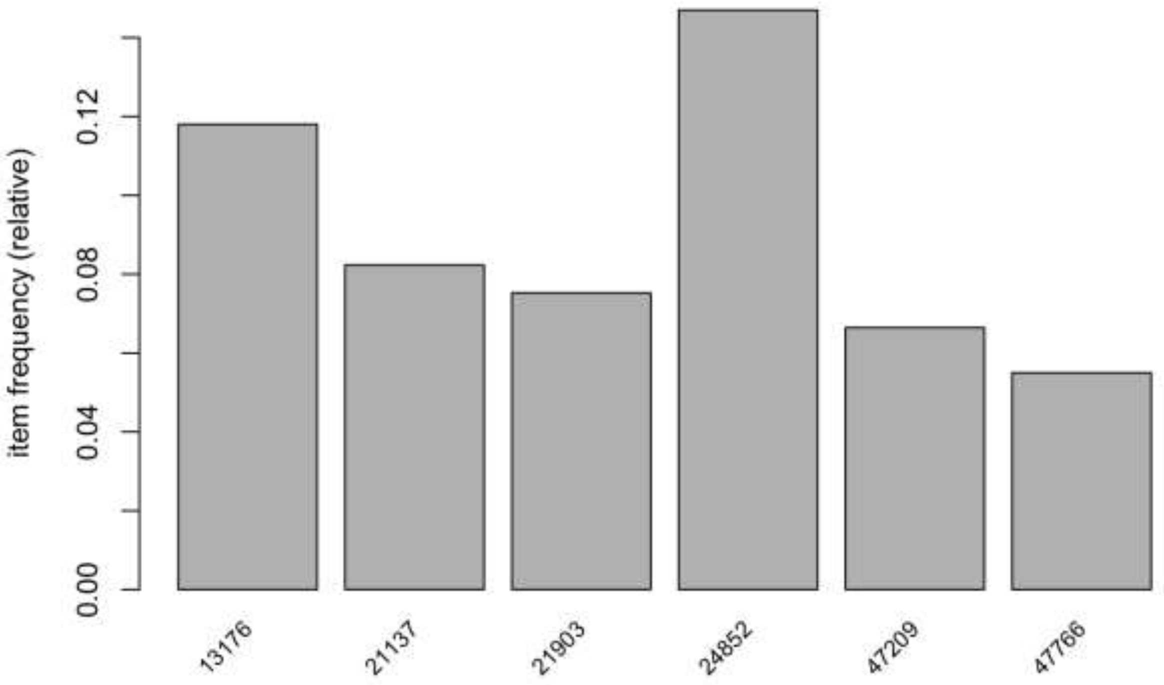
Model Selection and Analysis

There are several uses for market basket analysis, such as cross-selling, product placement, affinity marketing, fraud detection, and consumer behavior [1][3]. We will use the Apriori algorithm for mining association rules and make a comparison with the Frequent Pattern Growth Algorithm[10][11]. The user id and product id serve as the keys of a denormalized structure that is formed after the features are created. The issue then transforms into a classification problem that requires a classifier algorithm to be used to solve. XGBoost is the classifier that we have selected.

Market Basket Analysis

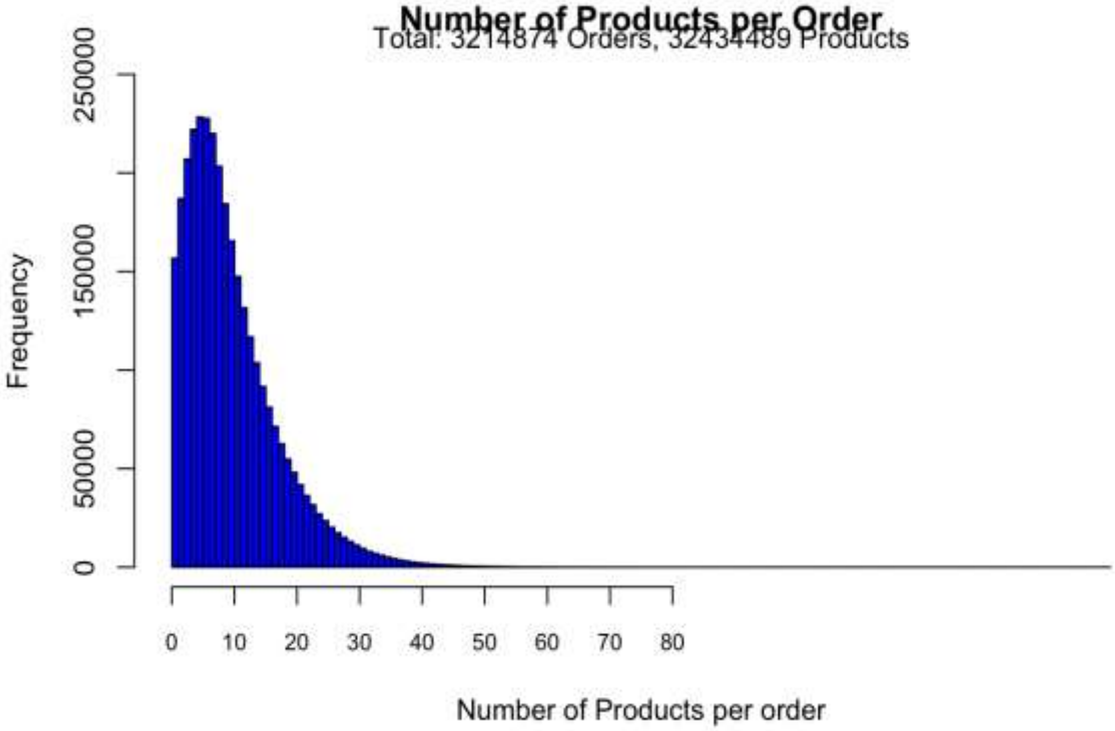
Now we will perform Market Basket Analysis using Apriori algorithms to predict grouped items, transactions, frequency, rules and define meaning full definitions[11]. Before we proceed further, we Please refer to the Appendix for various important definitions associated with the Market basket analysis. We converted the prior table into transactions to perform analysis to get the frequency items and plots and from there we would derive rules with support and confidences[10].

On converting to transactions there are over 3 million+ transactions and 50 K columns. we performed products that occurred most frequently in the transactions with a support of at least 0.05.

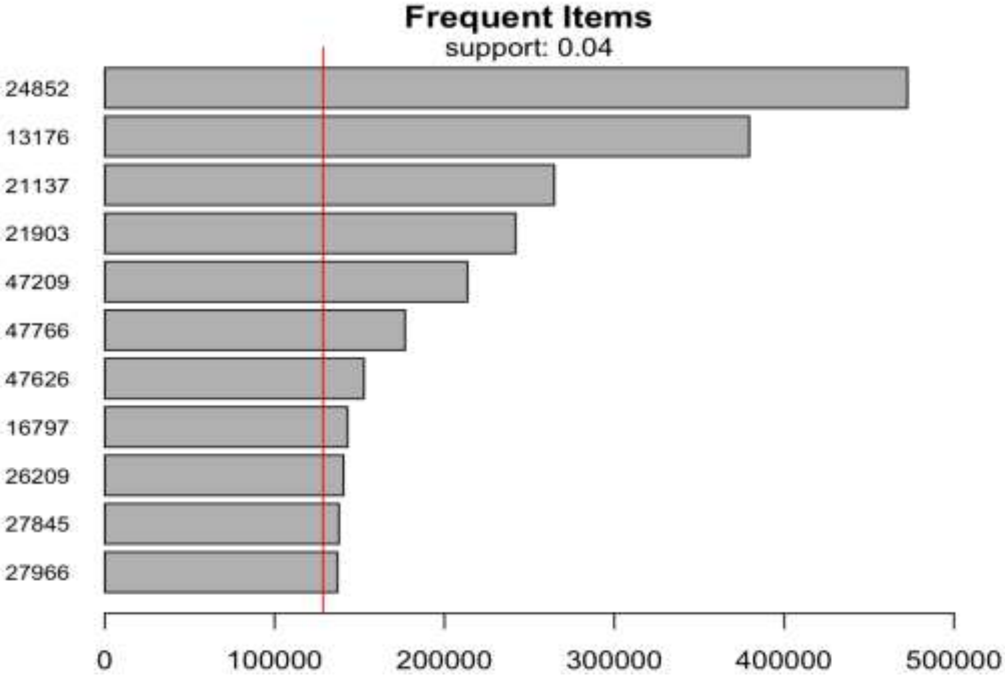


The productsID in the x-axis were the most frequently brought products.

Below is the graph where we examined the number of products per order.

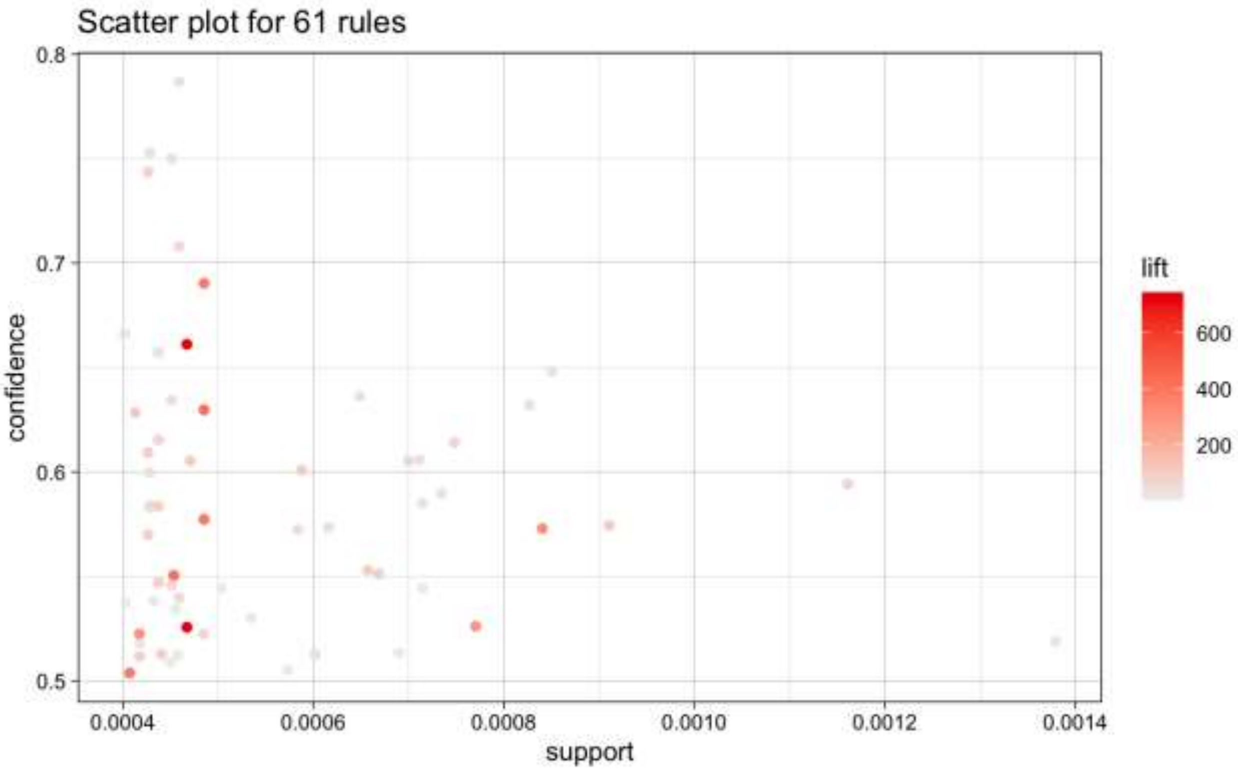


We see that a person usually buys around 1 - 40 products per orders in general. We then further examined the number of frequent items brought with a better support we choose the number 0.04 which is about 4%.



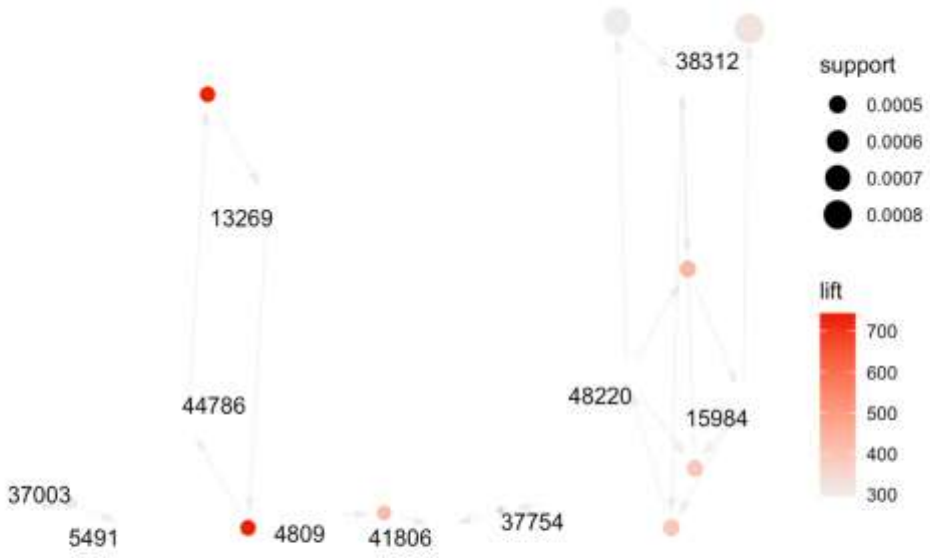
We find 11 products to occur when the support is set at 0.04. This means these products are found in 4% of the total transactions which is approximately about 140k.

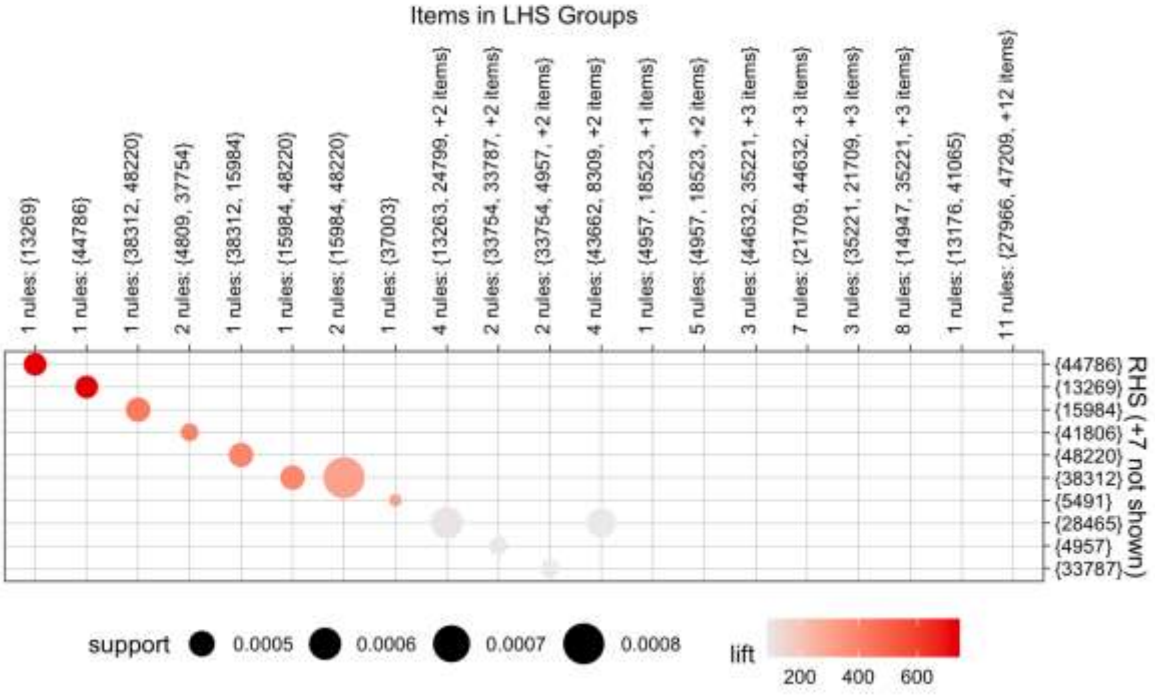
We decided on a lower support value of 0.0004 since we want to produce 2 items and 3 product combinations. This means that the product gets sold 1200 times out of 3 million transactions, or around 0.04% of all transactions.



From the above we can see that rules with good lift are between the confidence of 50-70% and a support of 0.0004 -0.0005

Below we can map the rules and its confidence and support accordingly. We see that the order where product 13269 has a strong lift with product 4809





The support, confidence, and lift three criteria are depicted in the above figure. The level of confidence is set at 50%. A list of 61 rules is provided. We order them according to the lift value, which determines the effectiveness of the rule, and create our product combinations as a result.

ML Model to predict the next buy

As mentioned, before we are going to use the XGBoost classifier to solve the problem of predicting the next buy[9]. To make the prediction we construct the Product and User metrics to make our prediction, our metrics are as below

Product metrics grouped by product IDs

1. Total Orders,
2. Total Order ratio for each product
3. Mean of add to cart
4. Total times the product was reordered.

User metrics grouped by User IDs

1. Total Orders
2. Mean day of the week
3. Mean hour of the day
4. Mean day of the week and hour of the day
5. Days since the prior order

We then use the above metrics to create the combination metrics to perform our classification So hence we group by the above two datasets by user_id and product_id to create the below metrics

1. Total orders
2. Mean add to cart
3. Total reorders.

The above metrics and scores are converted into matrix and then we split the data to test and train and then perform the training of the model with the below hyperparameters.

Hyperparameter table

eta	0.1
max_depth	6
min_child_weight	10
gamma	0.7
subsample	0.77
colsample_bytree	0.95
alpha	0.00002
lambda	10

The model created is then used to predict the next orders where if the product is ordered (a probability greater than 21%) then we consider that product is reordered in a particular order.
In the final evaluation of the product we achieved F1 score of 0.404 for a million transactions which is good.

Conclusion and Future Work

In this experiment, we explored large amounts of data, and performed exploratory data analysis which one of the key concepts to derive mining, predict and understand results. We performed market basket analysis on the data by manipulating and merging several datasets and derived several rules which would help the customer to produce different offer sections which would give them a boost in the revenue generation. The data from web analytics shows how users act and how they are encouraged to behave by earlier website design decisions. By employing the results of the association analysis, business decision-making can be influenced. Using the MBA, we can add suggestions to product pages and product cart pages and utilize a list of rules. The restrictions that apply to each product with a high lift where the recommended product has a high margin should be considered. It has the potential to significantly increase profit. Also, using the XGBoost classifier we can predict the products that would be brought by the user in near future and using these analytics the company could perform curated results and thus increasing the business revenue.

Future work can include prediction based on neural nets, deep learning and using different metrics to predict the next buy. Also, Collaborative filtering can be used to suggest products to customers.

References

1. Fachrul Kurniawan, Binti Umayah, Jihad Hammad, Supeno Mardi Susiki Nugroho and Mochammad Hariadi, “Market Basket Analysis to Identify Customer Behaviors by Way of Transaction Data” Knowledge Engineering and Data Science (KEDS) -Vol 1, No 1, January 2018, pp. 20–25

2. Manpreet Kaur ,Shivani Kang. “Market Basket Analysis: Identify the changing trends of market data using association rule mining”, International Conference on Computational Modeling and Security (CMS 2016), Procedia Computer Science 85 (2016) 78 – 85

3. A. Herman, L.E. Forcum, Joo Harry. Using Market Basket Analysis in Management Research, Journal of Management, 39 (7) (2013), pp. 1799-1824

Website Links:

4. Market Basket Analysis - an overview | ScienceDirect Topics
5. Association Rules in Data Mining | Learn the Algorithms, Types, and Uses (educba.com)
6. Data Extraction | Data Cleaning | Data Manipulation in R | Intellipaat
7. What Is Data Analysis? Methods, Techniques, Types & How-To (datapine.com)
8. Exploratory Data Analysis: Useful R Functions for Exploring a Data Frame | R-bloggers
9. Beginner’s Guide to XGBoost for Classification Problems | Towards Data Science
10. Association Rules in R - Analytics Tuts (analytics-tuts.com)
11. R Market Basket Analysis using Apriori Examples | DataCamp

Appendix

Tools Used:

Software Packages: RStudio, R.

Development: GitHub, Notion.

Project Planning: Excel, Notion Kanban boards.

Libraries: data.table, dplyr, ggplot2, knitr, stringr, DT, magrittr, grid, gridExtra, sqldf, Matrix, arules, tidyr, arulesViz, methods data.table, xgboost.

Documentation: Microsoft Word, Notion

Source Code:

We used GitHub for collaborate of our project with our team members.

GitHub Link: https://github.com/AstroLeague/CSP571_Project