NYU

instacart

Business Analytics Final Presentation

Raj Mehta | rcm445@nyu.edu

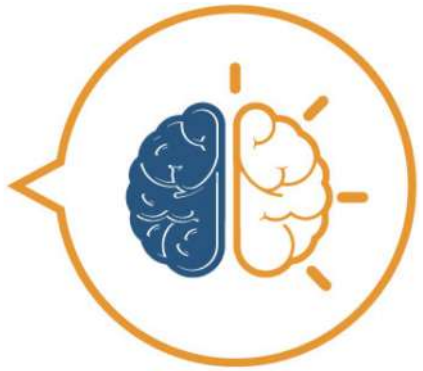
Agenda

- Our Goal
- Business Problem
- The Dataset
- Methodology Used
- Model Implementation
- Model Evaluation
- Business Implications

Who are we? And What is our Goal?

instacart

Business Intelligence Team



We are here on a mission to simplify weekly tasks for customers!

To enhance customer experience

To increase Instacart revenue

What is the company about?

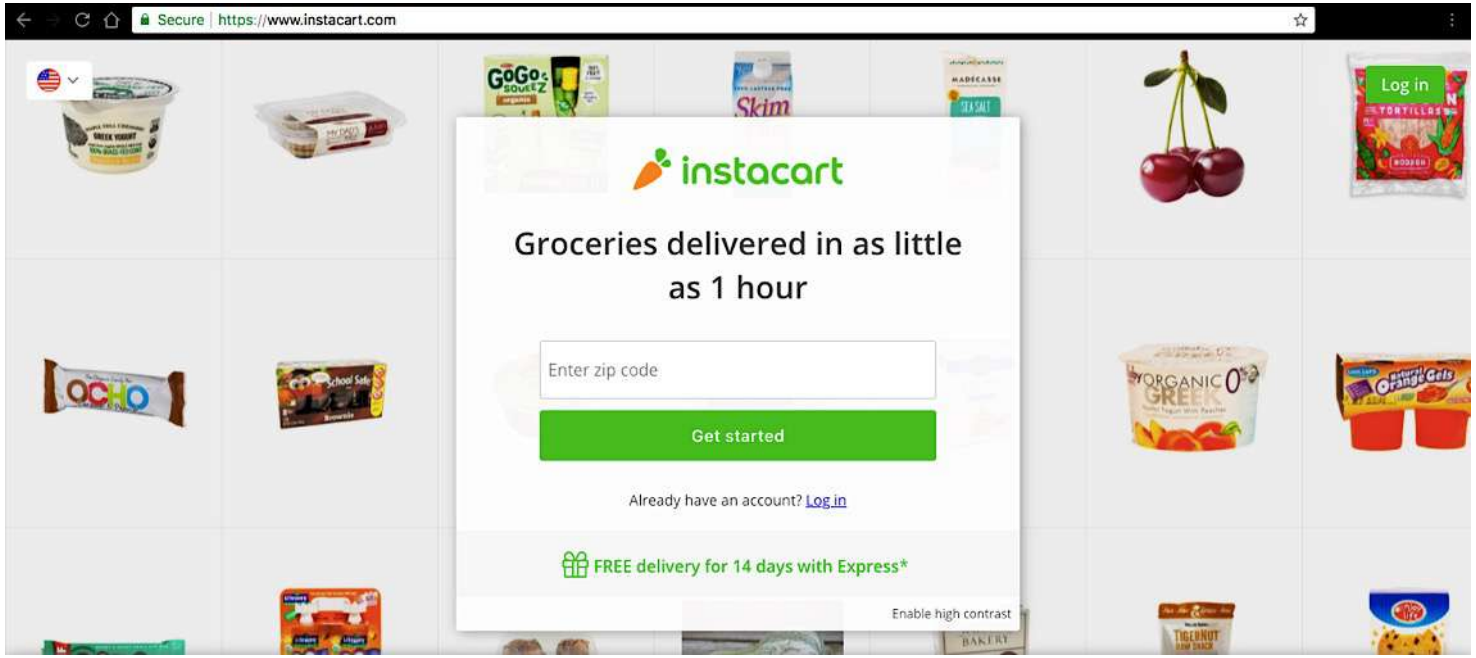
instacart

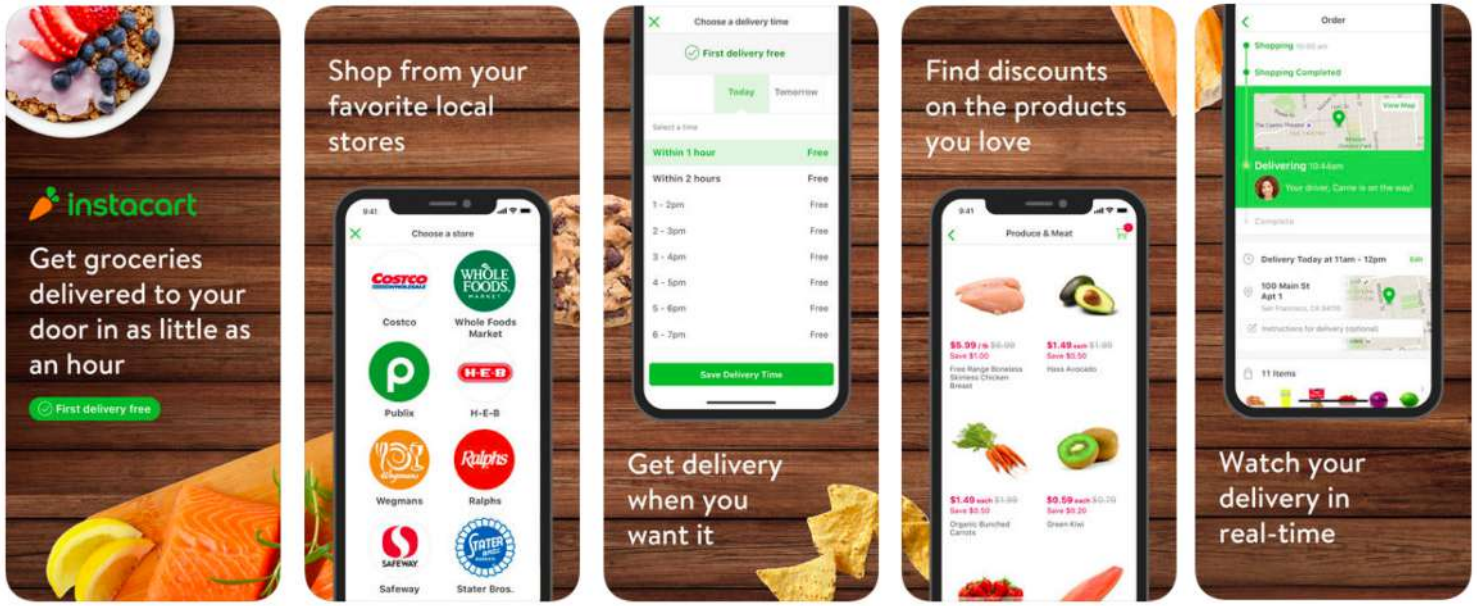
Products you love

From Stores you Love

Delivered to your Doorstep

How does the company work?





Problem we are trying to solve for Instacart

How to improve Customer Experience?

What are the gaps?

- Customer Retention
- Customers Attraction
- Instacart Profitable Relationships

Who's getting the benefits?

- Customers
- Partners
- Shoppers
- Instacart

Data Selection

Size: 3 Million Transactional Orders
Source: Company Website with 3 Million Instacart Orders, Open Sourced
Structure: 6 Tables

Departments [1:21]
Aisles [1:134]

Products [1:49688]
Orders [1:3421083]

Prior Orders [1:32434489]
Train Orders [1:1384617]

aisles [1:134]		
🔑	aisle_id	integer
	aisle	varchar

products [1:49688]		
🔑	product_id	integer
	product_name	varchar
🔑	aisle_id	integer
🔑	department_id	integer

departments [1:21]		
🔑	department_id	integer
	department	varchar

order_products__prior [1:32434489]		
🔑	order_id	integer
🔑	product_id	integer
	add_to_cart_order	integer
	reordered	integer

order_products__train [1:1384617]		
🔑	order_id	integer
🔑	product_id	integer
	add_to_cart_order	integer
	reordered	integer

orders [1:49688]		
🔑	order_id	integer
	user_id	integer
	eval_set	varchar
	order_num	integer
	order_dow	integer
	order_hour_of_day	integer
	day_since_prior_order	integer

Data Analysis & Proposed Methodology

- Aisle
- Products
- Orders
- Departments
- Order Products Prior
- Order Products Train

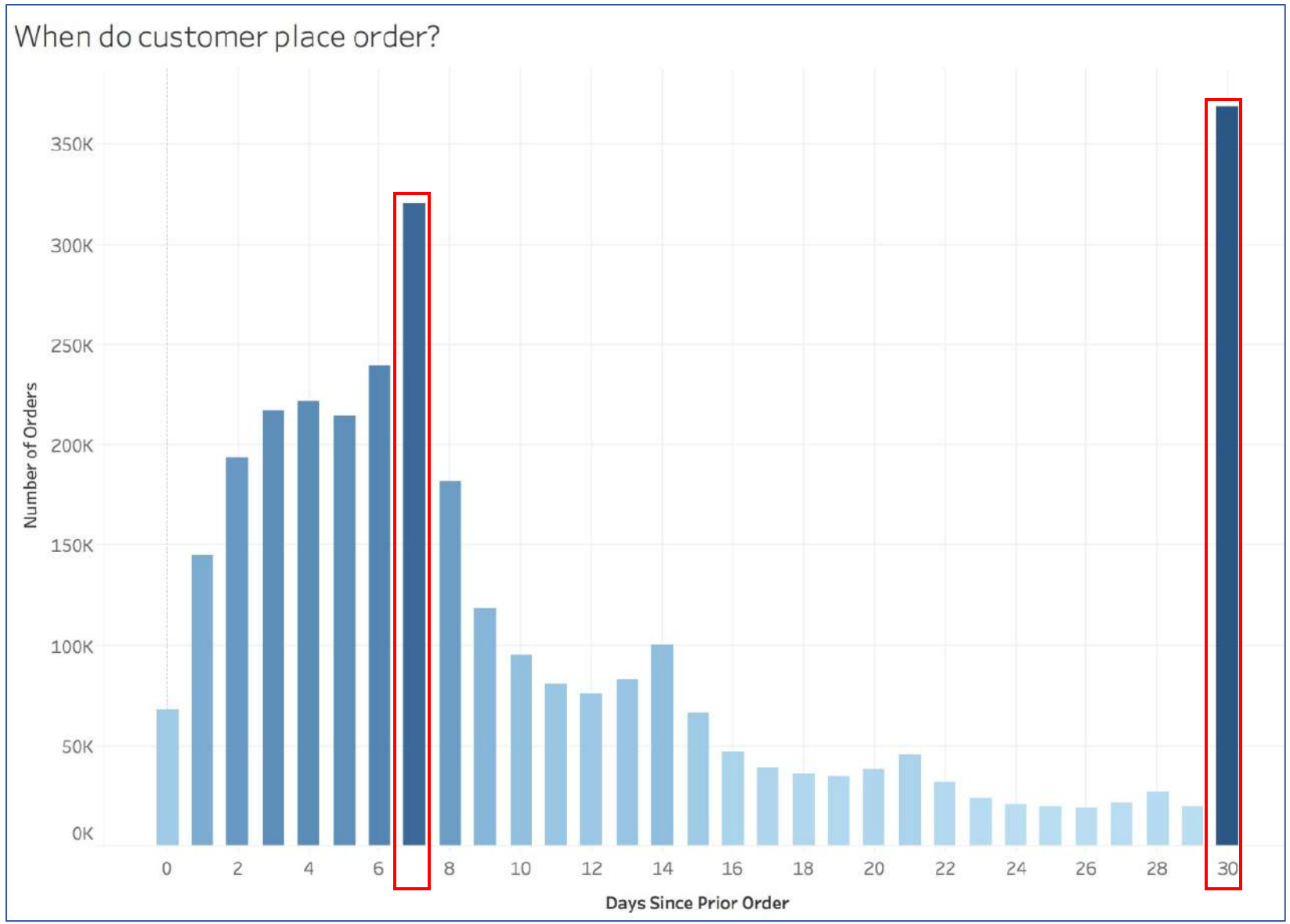
Exploratory Analysis

Data Pre-Processing

Predictive Analysis

Market Basket Analysis (MBA)

Exploratory Analysis

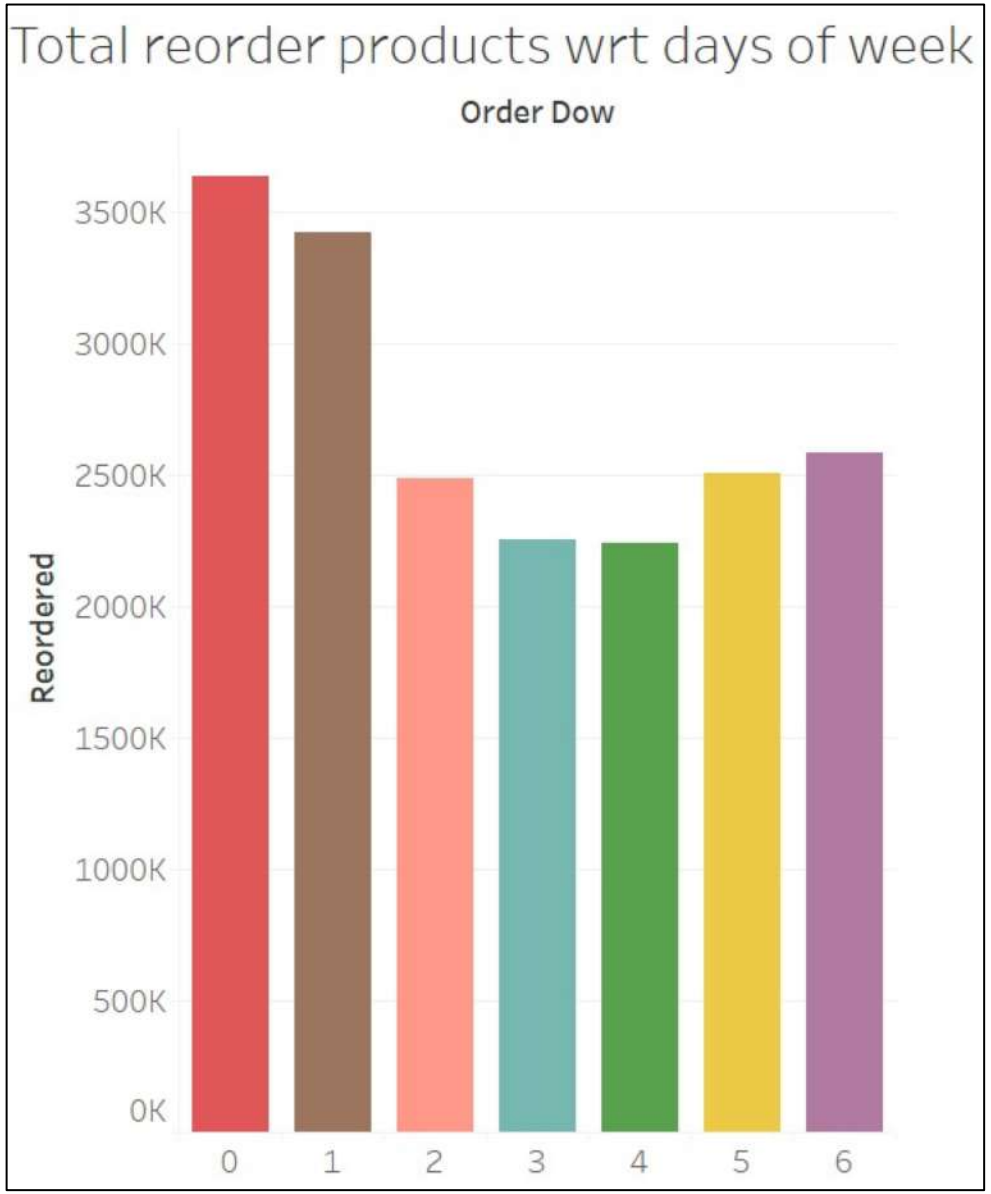


Days Since Prior Order!

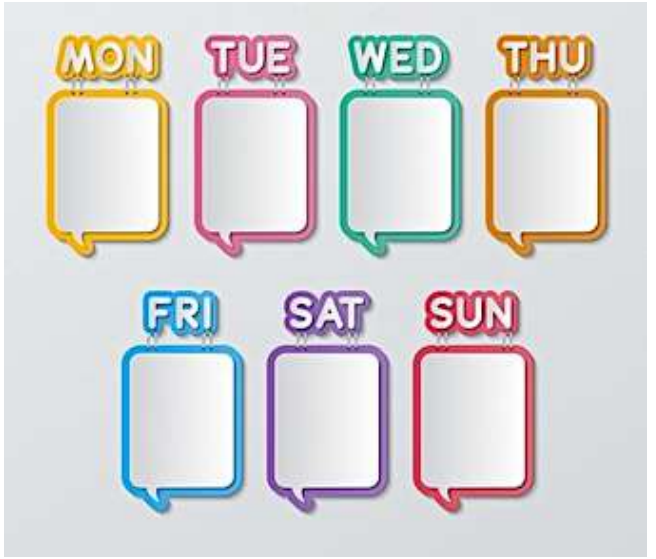
Order after:

- 7 Days i.e. Weekly
- 30 Days i.e. Monthly

Exploratory Analysis



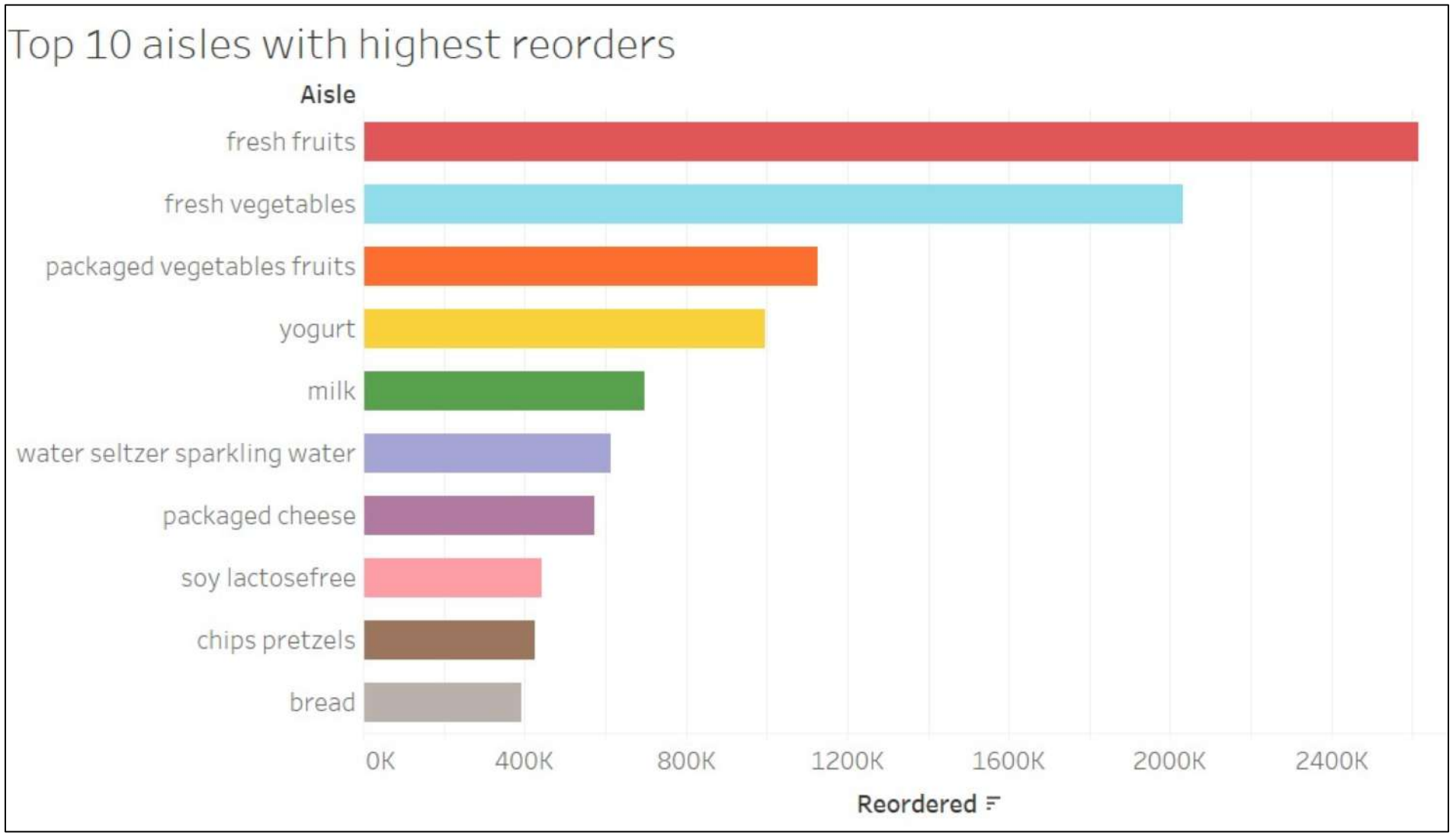
What day of the week are people likely to place an order?



Maximum Orders placed on:

- Day 0 of the week
- Day 1 of the week

Exploratory Analysis



Which Aisle is the favorite?

- Fresh Fruit
- Fresh Vegetables

Data Pre-Processing

Feature Engineering

Feature Identification

Joining Data

~28 New Features

Train Data: 8M

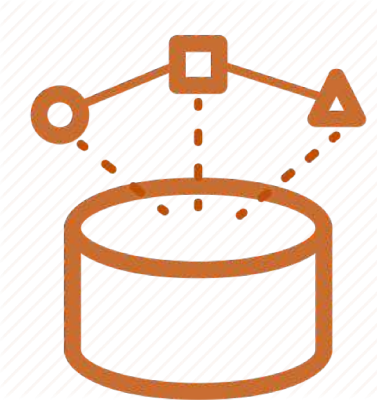
For Instance:

People Reorder

- Weekly
- Monthly

Maximum Orders placed on

- Day 0 of the week
- Day 1 of the week



Data Pre-Processing

New Features created:

Number of reorders on a particular Day

```
#Summing according to the day of the week the reorder number
prod_dow_0 = sum(reordered & order_dow == 0),
prod_dow_1 = sum(reordered & order_dow == 1),
prod_dow_2 = sum(reordered & order_dow == 2),
prod_dow_3 = sum(reordered & order_dow == 3),
prod_dow_4 = sum(reordered & order_dow == 4),
prod_dow_5 = sum(reordered & order_dow == 5),
prod_dow_6 = sum(reordered & order_dow == 6),
#Reorder number weekly, biweekly and monthly
prod_reweekly = sum(reordered & days_since_prior_order <= 6, NA, na.rm = TRUE),
prod_rebiweekly = sum(reordered & (days_since_prior_order >= 7 & days_since_prior_order <= 14), NA, na.rm = TRUE),
prod_remonthly = sum(reordered & (days_since_prior_order >= 15 & days_since_prior_order <= 31), NA, na.rm = TRUE)
```

Reorder Probability

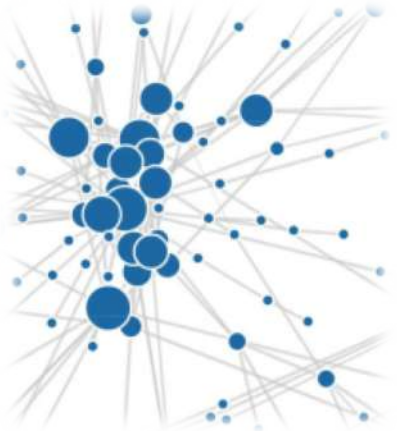
```
prd$prod_reorder_probability <- (prd$prod_reorder_probability_21 +
                                 prd$prod_reorder_probability_32) / 2

#How many times the product was reordered wrt the first one
#prd$prod_reorder_times <- prd$prod_reorders / prd$prod_first_orders
#Probabilty of the reodering of the product overall
prd$prod_reorder_ratio <- prd$prod_reorders / prd$prod_orders
```

Predictive Analysis

```
model_xgb <- xgboost(data = as.matrix(subtrain1),
                     label = as.matrix(as.numeric(subtrain$reordered)),
                     objective = "reg:logistic",
                     eval_metric = "logloss",
                     nrounds = 30
                     eta = 0.2,
                     max_depth = 6,
                     min_child_weight = 10,
                     gamma = 0.70,
                     subsample = 0.76,
                     colsample_bytree = 0.95,
                     alpha = 2e-05,
                     lambda = 10)
```

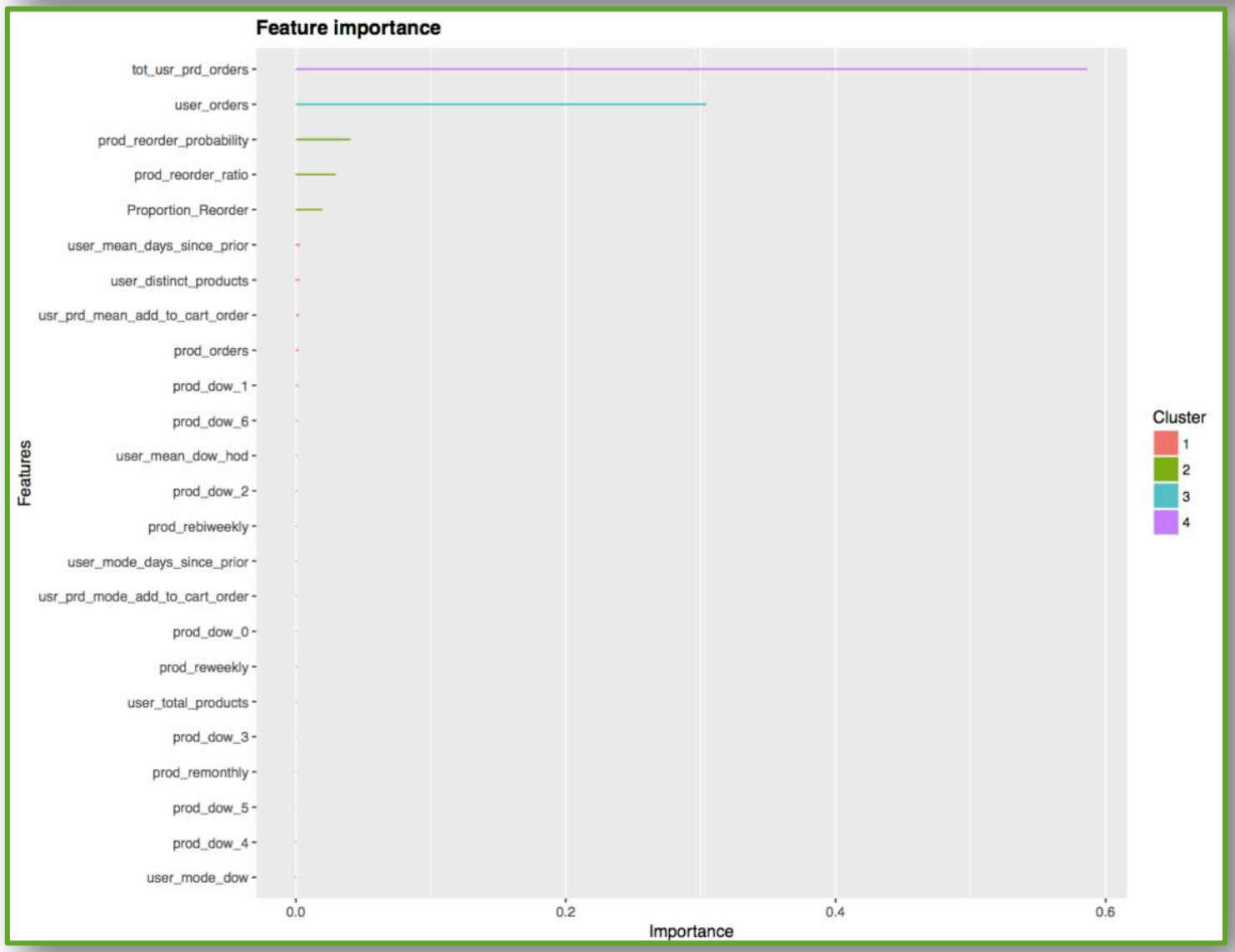
```
[21]    train-logloss:0.262702
[22]    train-logloss:0.262461
[23]    train-logloss:0.262279
[24]    train-logloss:0.262108
[25]    train-logloss:0.261981
[26]    train-logloss:0.261884
[27]    train-logloss:0.261792
[28]    train-logloss:0.261715
[29]    train-logloss:0.261633
[30]    train-logloss:0.261569
```

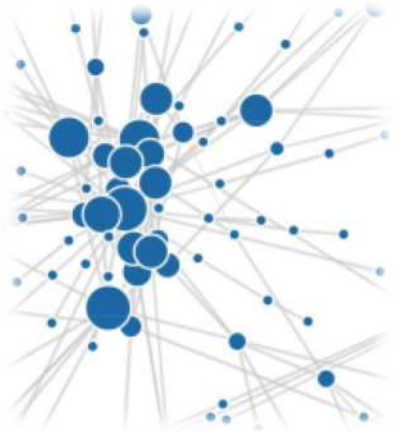
Accuracy = 73.84%



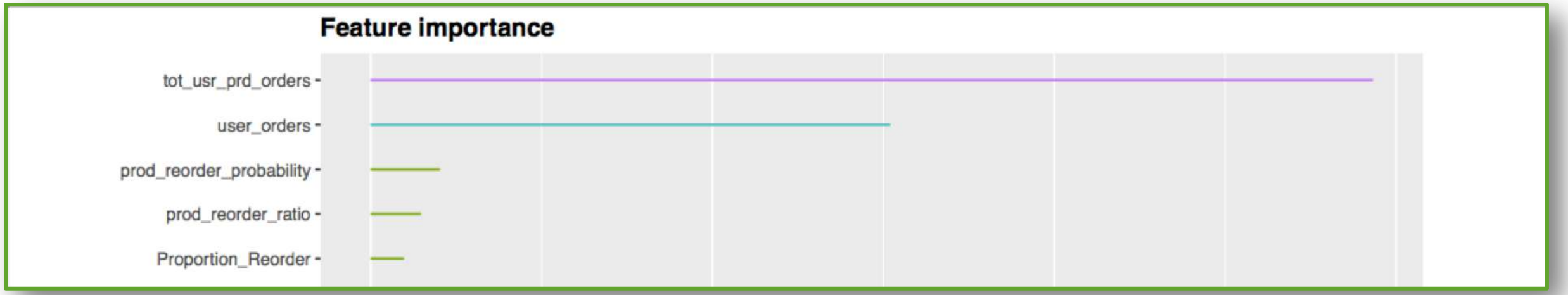
Predictive Analysis

```
importance <- xgb.importance(feature_names = colnames(subtrain %>% select(-reordered)),model = model_xgb)
xgb.ggplot.importance(importance, top_n = 15)
```





Predictive Analysis



Our Top 5 Features

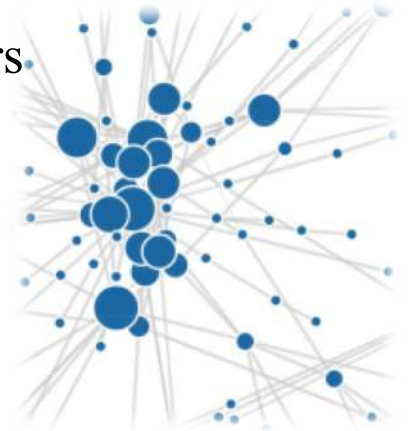
tot_usr_prd_orders: Number of times a product is ordered by a particular user

user_orders: Maximum Orders made by a user

prod_reorder_probability: Probability of a product being reordered with respect to prior orders

prod_reorder_ratio: A product being reordered with respect to all the orders of that product

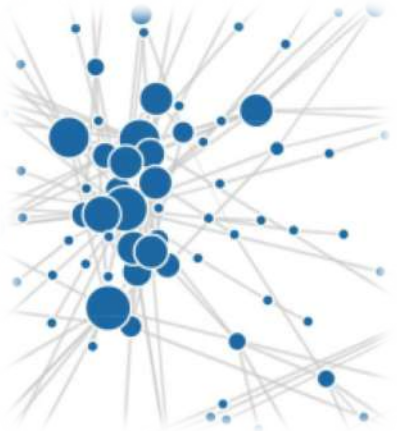
Proportion_Reorder: Probability of a user reordering



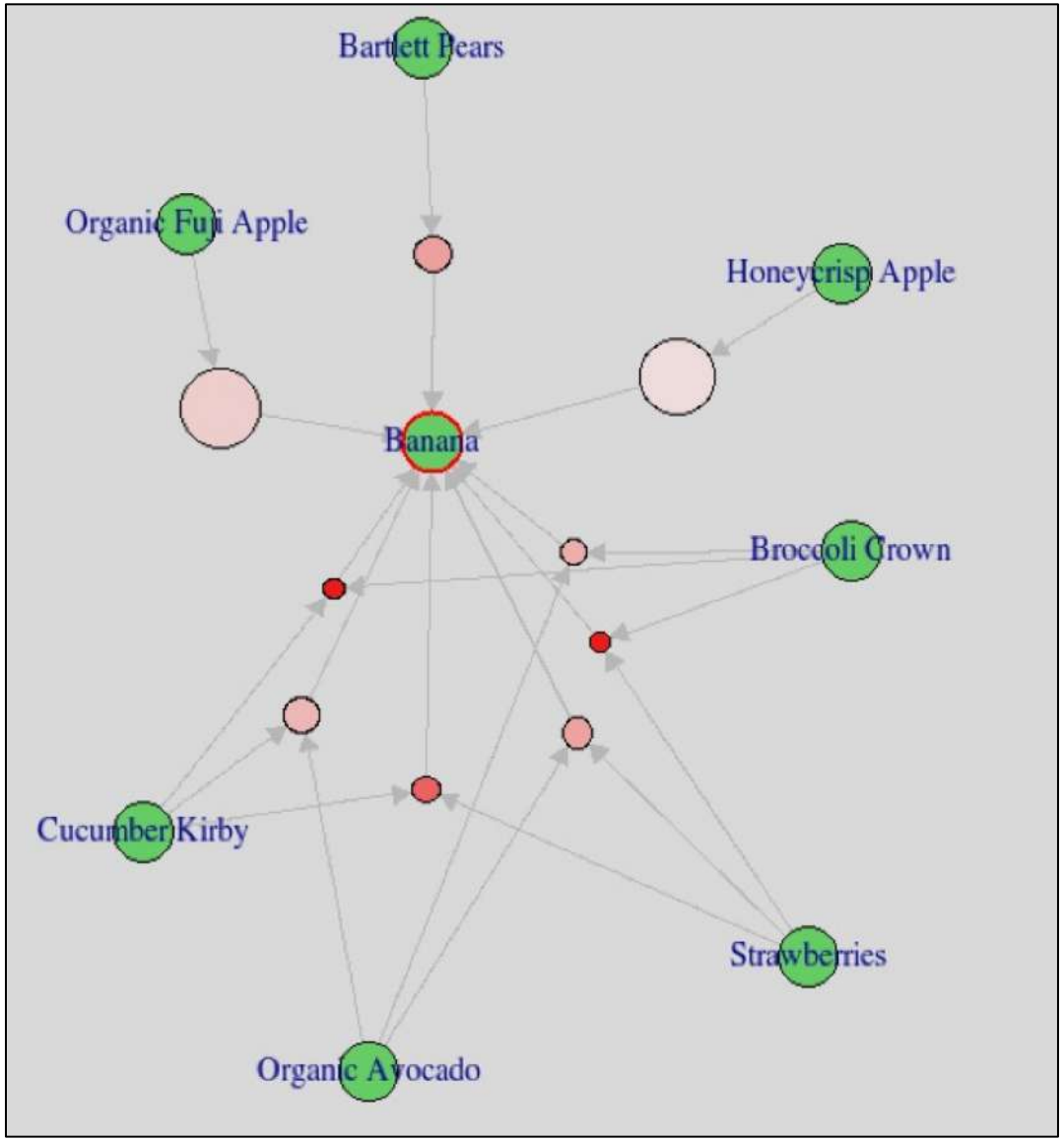
Predictive Analysis

	order_id	product_name
1	2774568	Granny Smith Apples
2	2774568	Strawberries
3	2774568	Unsweetened Chocolate Almond Breeze Almond Milk
4	2774568	Garlic Couscous
5	2774568	Organic Baby Spinach
6	2774568	Organic Whole String Cheese
7	2774568	Vanilla Unsweetened Almond Milk
8	2774568	Organic Peeled Whole Baby Carrots
9	2774568	Organic Avocado
10	1528013	Organic Baby Spinach
11	1528013	Ground Turkey Breast

Submission File with Order ID and Products Reordered



Market Basket Analysis



Products

Association Rule

Low Confidence

Moderate Confidence

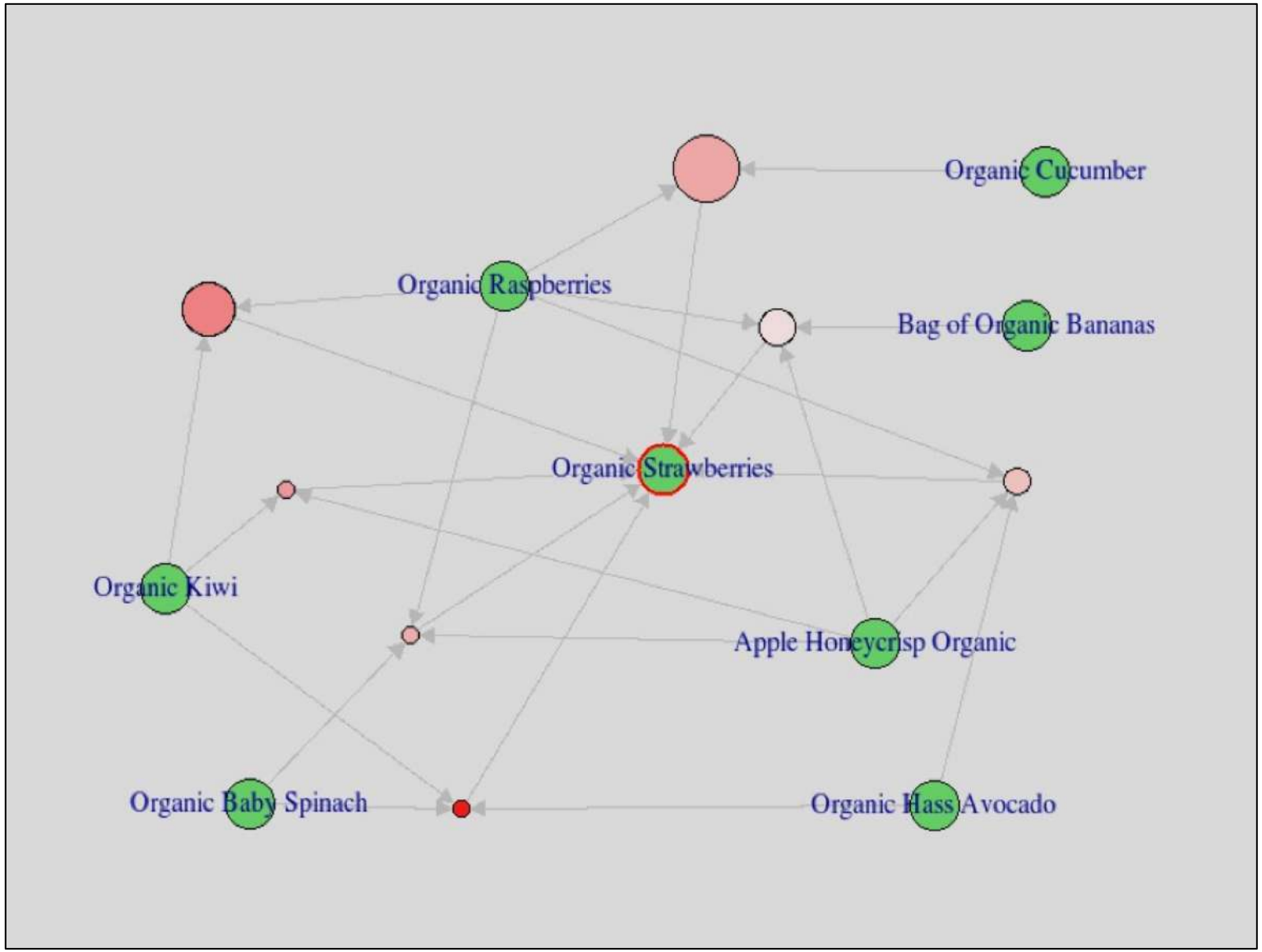
High Confidence

Low Support

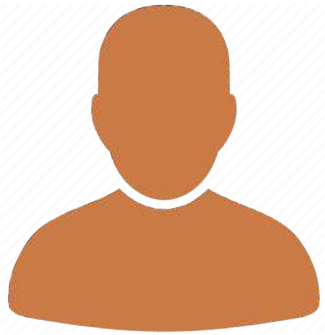
Moderate Support

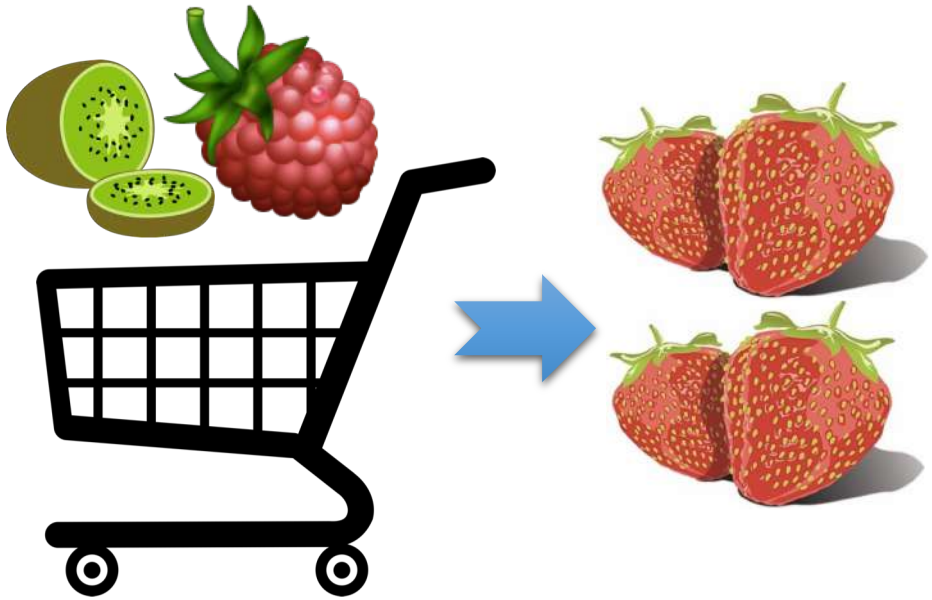
High Support

Market Basket Analysis



- Fresh Fruit
- Fresh Vegetables





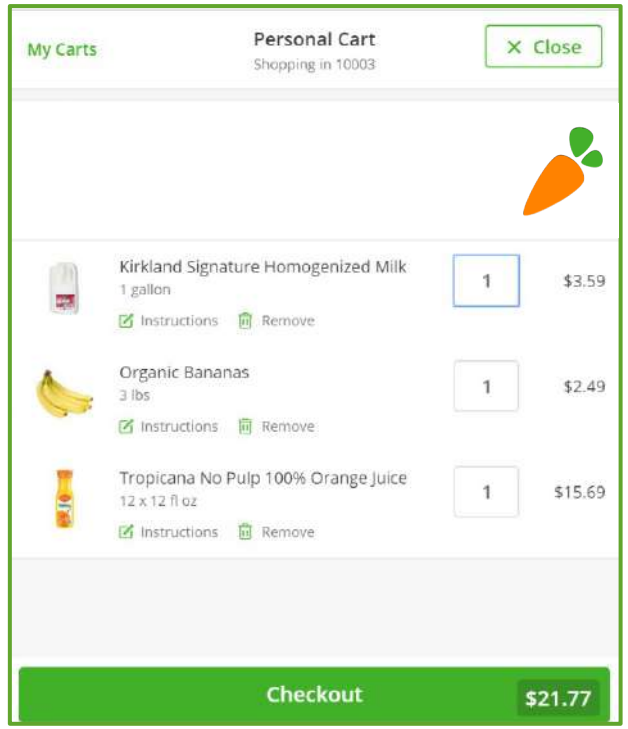
Business Implications based on:

- Feature Importance Matrix
- Market Basket Analysis

The Auto-Carting Feature

Customer Personalization

Auto-Cart

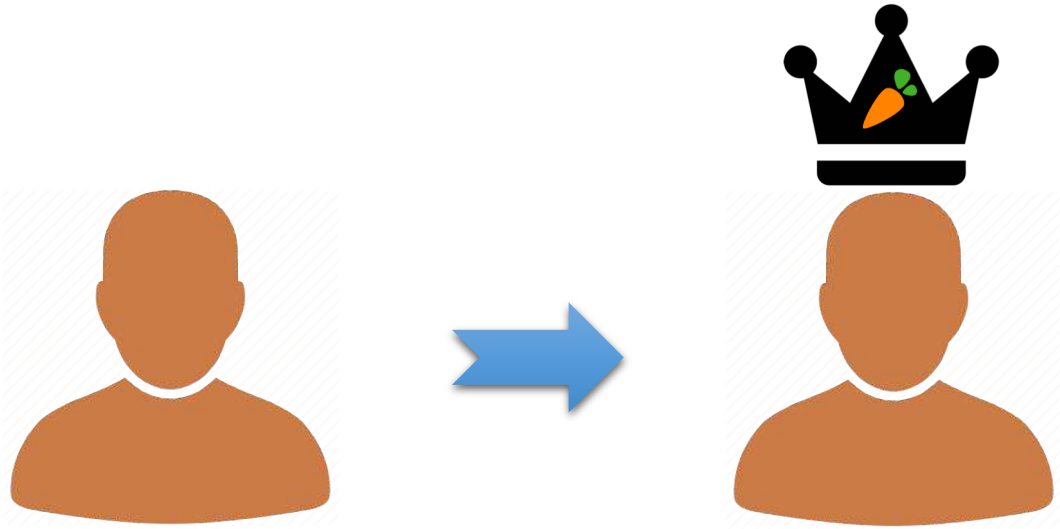


- **Number of times a Product is ordered by a particular User**
- **A product being reordered**
- **Reorder frequency**

Auto-Delivery

Shop More, Save More!

Customer Retention



Insta Membership

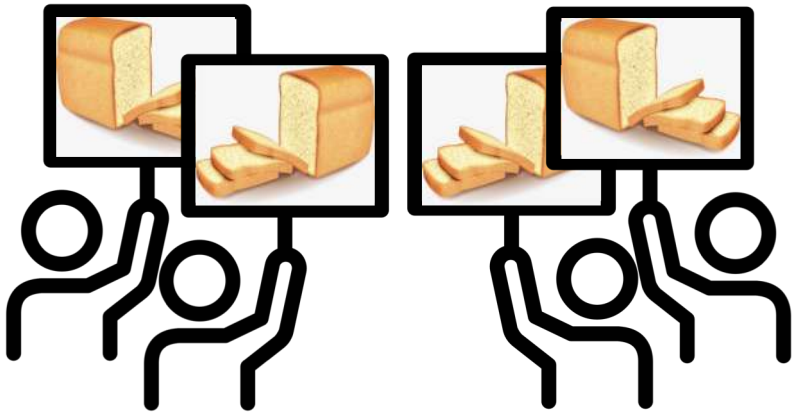
- Unlimited Free Delivery
- "Cart-King" Discounts
- Priority Delivery

- **Maximum Orders made by a user**
- **User reordering capacity**

Business Implication 3

You say it, We have it!

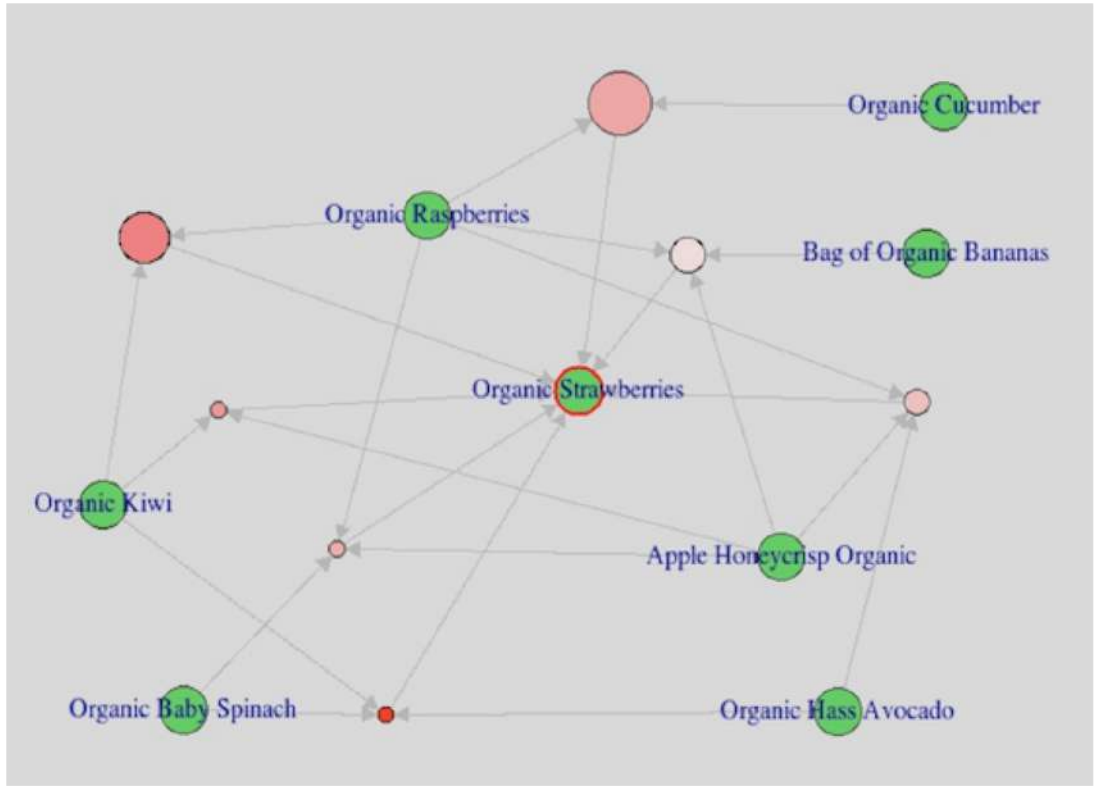
Customer Satisfaction



- Maximum time a product is being reordered
- Number of times a Product is ordered by a particular User

Made for Each Other Products!

Customer Task Simplicity

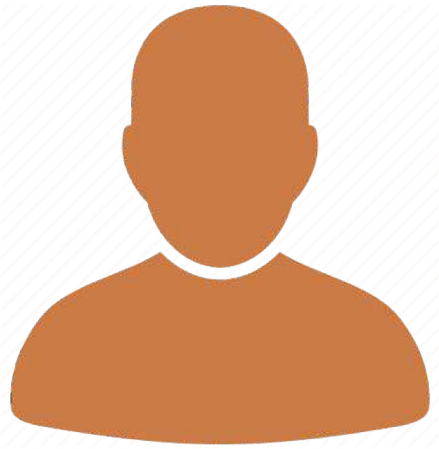


Effective Search System

Recommendations of Associated & Complementary Products

More than Just Deals!

Customer Attraction



Insta Bonanza

- Promotional Offers
- Cash Back Offers
- Referral Bonus

- Best day on which maximum orders are made
- Best time of the week
- Specific User order

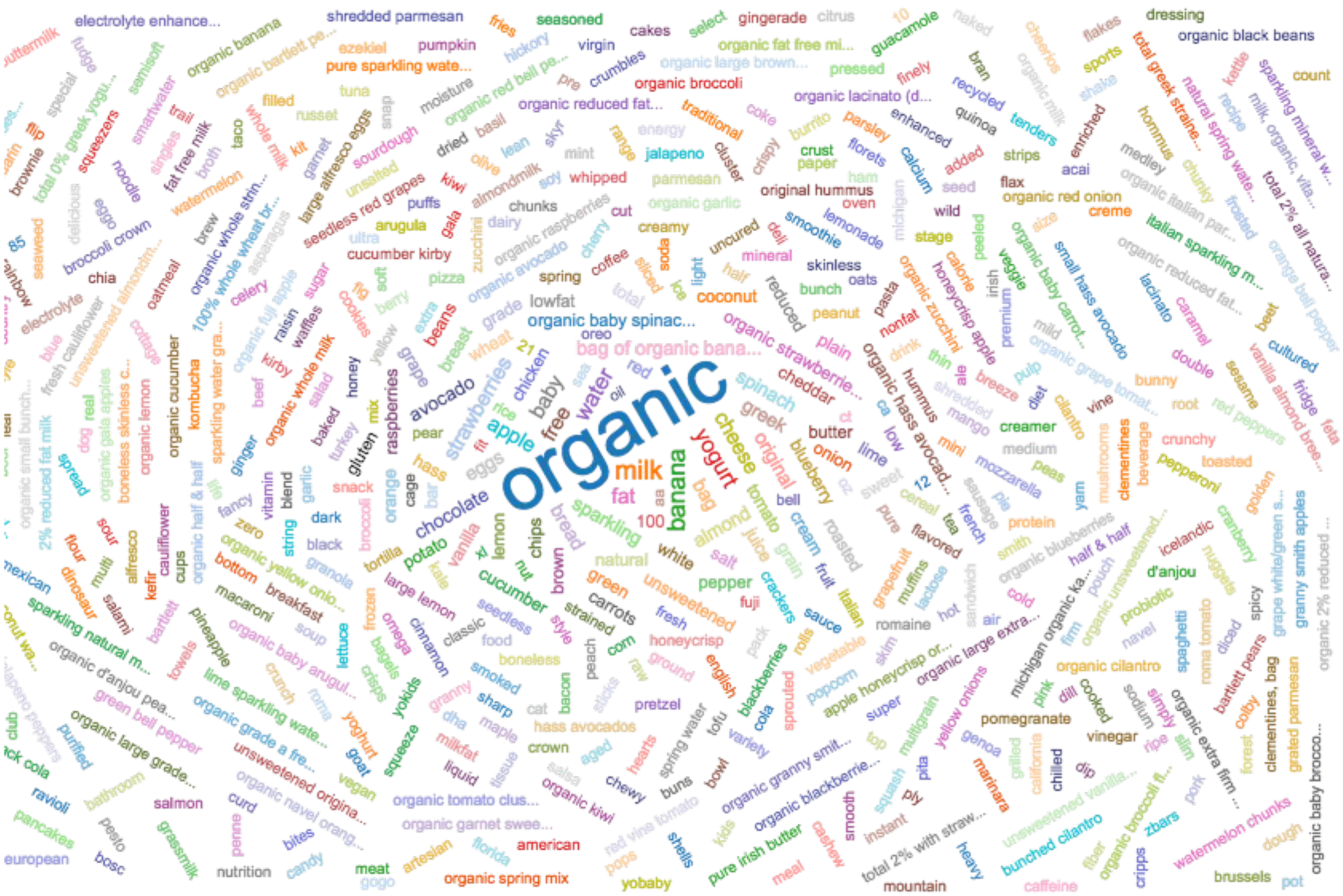
Healthy is the New Wealthy!

Explore New Market

Department
&
Products

↓

Organic Market
&
Farm Market



NYU

Thank You

instacart

Business Intelligence Team

Appendix: Data Description

Dataset	Number of Rows
aisles.csv	134

Column Name	Column Description
aisle_id	Unique ID for aisle
aisle	Name of the aisle

aisle_id	aisle
1	prepared soups salads
2	specialty cheeses
3	energy granola bars
4	instant foods
5	marinades meat preparation
6	other
7	packaged meat
8	bakery desserts
⋮	⋮
126	feminine care
127	body lotions soap
128	tortillas flat bread
129	frozen appetizers sides
130	hot cereal pancake mixes
131	dry pasta
132	beauty
133	muscles joints pain relief
134	specialty wines champagnes

```
> str(aisles)
Classes 'data.table' and 'data.frame':  134 obs. of  2 variables:
 $ aisle_id: int  1 2 3 4 5 6 7 8 9 10 ...
 $ aisle   : chr  "prepared soups salads" "specialty cheeses" "energy granola bars" "instant foods" ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Appendix: Data Description

Dataset	Number of Rows
products.csv	49688

Column Name	Column Description
product_id	An unique ID for a product
product_name	Name of the product corresponding to the ID
aisle_id	An unique ID for aisle
department_id	Department ID in which the product is placed

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich Cookies	61	19
2	All-Seasons Salt	104	13
3	Robust Golden Unsweetened Oolong Tea	94	7
4	Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce	38	1
5	Green Chile Anytime Sauce	5	13
6	Dry Nose Oil	11	11
7	Pure Coconut Water With Orange	98	7
8	Cut Russet Potatoes Steam N' Mash	116	1
⋮			
49682	California Limeade	98	7
49683	Cucumber Kirby	83	4
49684	Vodka, Triple Distilled, Twist of Vanilla	124	5
49685	En Croute Roast Hazelnut Cranberry	42	1
49686	Artisan Baguette	112	3
49687	Smartblend Healthy Metabolism Dry Cat Food	41	8
49688	Fresh Foaming Cleanser	73	11

```
> str(products)
Classes 'data.table' and 'data.frame':  49688 obs. of  4 variables:
 $ product_id   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ product_name : chr  "Chocolate Sandwich Cookies" "All-Seasons Salt" "Robust Golden Unsweetened Oolong Tea" "Smart Ones Classic Favori
tes Mini Rigatoni With Vodka Cream Sauce" ...
 $ aisle_id     : int  61 104 94 38 5 11 98 116 120 115 ...
 $ department_id: int  19 13 7 1 13 11 7 1 16 7 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Appendix: Data Description

Dataset	Number of Rows
departments.csv	21

Column Name	Column Description
department_id	Unique ID for department
department	Name of the Department. For example: Frozen, Bakery, etc.

department_id	department
1	frozen
2	other
3	bakery
4	produce
5	alcohol
6	international
7	beverages
8	pets
⋮	⋮
16	dairy eggs
17	household
18	babies
19	snacks
20	deli
21	missing

```
> str(departments)
Classes 'data.table' and 'data.frame':  21 obs. of  2 variables:
 $ department_id: int  1 2 3 4 5 6 7 8 9 10 ...
 $ department   : chr  "frozen" "other" "bakery" "produce" ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Appendix: Data Description

Dataset	Number of Rows
orders.csv	3421083

Column Name	Column Description
order_id	An unique ID for each order
user_id	An unique ID for the user that placed the corresponding order
eval_set	Which evaluation set the order belongs in
order_num	Order number for the user
order_dow	The day of week when the order was placed
order_hour_of_day	Hour of day when the order was placed
day_since_prior_order	Day since the last placed order (upto 30 days)

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
2539329	1	prior	1	2	8	NA
2398795	1	prior	2	3	7	15
473747	1	prior	3	3	12	21
2254736	1	prior	4	4	7	29
431534	1	prior	5	4	15	28
3367565	1	prior	6	2	7	19
550135	1	prior	7	1	9	20
3108588	1	prior	8	1	14	14
2295261	1	prior	9	1	16	0
2550362	1	prior	10	4	8	30
1187899	1	train	11	4	8	14
2168274	2	prior	1	2	11	NA
1501582	2	prior	2	5	10	10
⋮						
2719402	63100	prior	2	0	8	14
2855250	63100	prior	3	2	13	16
2355832	63100	prior	4	5	16	30
2362552	63100	train	5	1	13	30

```
> str(orders)
Classes 'data.table' and 'data.frame':  3421083 obs. of  7 variables:
 $ order_id              : int  2539329 2398795 473747 2254736 431534 3367565 550135 3108588 2295261 2550362 ...
 $ user_id               : int  1 1 1 1 1 1 1 1 1 1 ...
 $ eval_set              : chr  "prior" "prior" "prior" "prior" ...
 $ order_number          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ order_dow             : int  2 3 3 4 4 2 1 1 1 4 ...
 $ order_hour_of_day     : int  8 7 12 7 15 7 9 14 16 8 ...
 $ days_since_prior_order: num  NA 15 21 29 28 19 20 14 0 30 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Appendix: Data Description

Dataset	Number of Rows
order_products__prior.csv	32434489

Column Name	Column Description
order_id	An unique ID for prior order
product_id	An unique ID for a product
add_to_cart_order	Order in which each product was added to cart
reordered	1 if this product has been ordered by the user in the past, 0 otherwise

order_id	product_id	add_to_cart_order	reordered
2	33120	1	1
2	28985	2	1
2	9327	3	0
2	45918	4	1
2	30035	5	0
2	17794	6	1
2	40141	7	1
2	1819	8	1
2	43668	9	0
⋮			
110719	35140	2	0
110720	28849	1	1
110720	35594	2	0
110720	16185	3	1

```
> str(orders_prior)
Classes 'data.table' and 'data.frame':  32434489 obs. of  4 variables:
 $ order_id         : int  2 2 2 2 2 2 2 2 2 3 ...
 $ product_id       : int  33120 28985 9327 45918 30035 17794 40141 1819 43668 33754 ...
 $ add_to_cart_order: int  1 2 3 4 5 6 7 8 9 1 ...
 $ reordered        : int  1 1 0 1 0 1 1 1 0 1 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Appendix: Data Description

Dataset	Number of Rows
order_products__train.csv	1384617

Column Name	Column Description
order_id	An unique ID for most recent order
product_id	An unique ID for a product
add_to_cart_order	Order in which each product was added to cart
reordered	1 if this product has been ordered by the user in the past, 0 otherwise

order_id	product_id	add_to_cart_order	reordered
1	49302	1	1
1	11109	2	1
1	10246	3	0
1	49683	4	0
1	43633	5	1
1	13176	6	0
1	47209	7	0
1	22035	8	1
36	39612	1	0
36	19660	2	1
⋮			
2593147	34969	4	1
2593147	33055	5	0
2593147	16185	6	1
2593147	7364	7	0

```
> str(orders_train)
Classes 'data.table' and 'data.frame':  1384617 obs. of  4 variables:
 $ order_id         : int  1 1 1 1 1 1 1 1 36 36 ...
 $ product_id       : int  49302 11109 10246 49683 43633 13176 47209 22035 39612 19660 ...
 $ add_to_cart_order: int  1 2 3 4 5 6 7 8 1 2 ...
 $ reordered        : int  1 1 0 0 1 0 0 1 0 1 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Team 5 Pitch - Instacart

Problems we are trying to solve for Instacart?

1. How to improve customer experience?
2. How to retain the existing customers?
3. How to invest and collaborate with specific market based on most popular departments?

Dataset

Size: 3 Million Transactional Orders
Source: https://www.instacart.com/datasets/grocery-shopping-2017
Structure: 6 Tables

Departments [1:21]
Aisles [1:134]
Products [1:49688]
Orders [1:3421083]
Prior Orders [1:32434489]
Train Orders [1:1384617]

Methods

- Data Exploratory Analysis and Data Preprocessing
- Modeling and Predictive Analysis using Random Forest and XGBoost
- Market Basket Analysis (MBA)

Appendix: Problems we are trying to solve for Instacart?

1. How to invest and collaborate with new and existing profitable Partners?

By evaluating the customer purchasing behaviors from past purchasing data, we could identify:

- Identify the existing profitable partners
- Explore new profitable partners

2. How to improve customer experience?

By evaluating customer reordering habits from the past purchasing data, we could:

- Employing effective search system: While searching, a customer will get similar items based on their previous order(s)
- Customer Personalization: Recommendation based on customers' previous order(s)
- Customer Task Simplicity: Using Market Basket Analysis, Recommendations of associated products and/or complement products based on on their current order

3. How to manage inventory?

By predicting recurring orders from the past data we could:

- Manage a balance between the supply and demand
- Fulfil customer satisfaction by having appropriate amount of stock of products with high demand

4. How to attract new customers and retain the existing customers?

Finding some relation between customer purchase and the time, we could identify the best time to roll out the promotional offers and discounts to pull the new customers and embrace the existing ones

Problems we are trying to solve for Instacart?

How to invest and collaborate with new and existing profitable Partners?

By evaluating the customer purchasing behaviors from past purchasing data, we could identify:

Identify the existing profitable partners	Explore new profitable partners

How to improve customer experience?

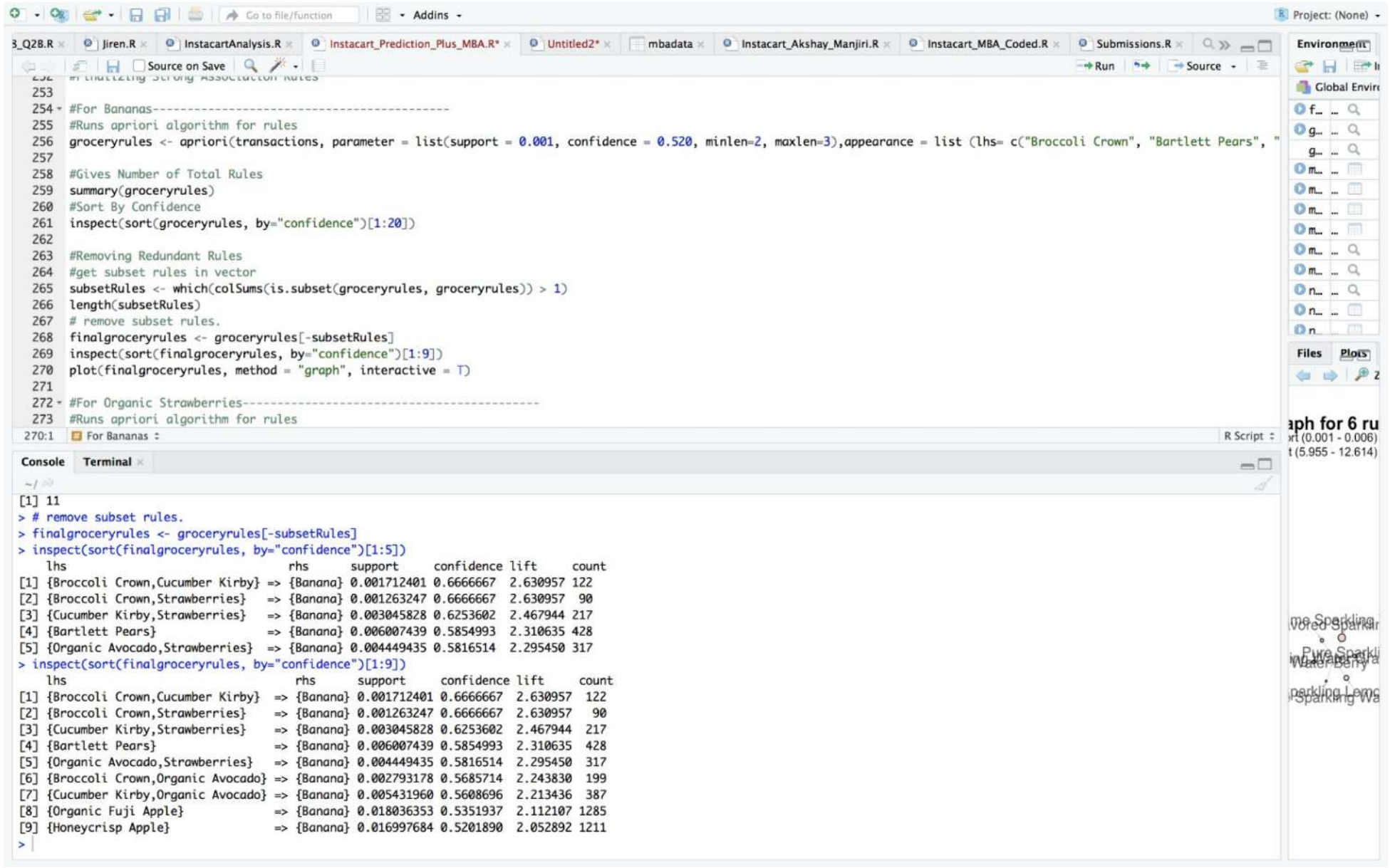
By evaluating customer reordering habits and recurring orders from the past purchasing data, we could:

Employing effective search system	Customer Task Simplicity
Customer Personalization (Auto-Cart)	Fulfil customer satisfaction

How to attract new customers and retain the existing customers?

Best Time to Roll-Out the Promotional Offers and Discounts

Appendix: MBA Statistics



```
#For Bananas----------------------------------------
#Runs apriori algorithm for rules
groceryrules <- apriori(transactions, parameter = list(support = 0.001, confidence = 0.520, minlen=2, maxlen=3),appearance = list (lhs= c("Broccoli Crown", "Bartlett Pears", "

#Gives Number of Total Rules
summary(groceryrules)
#Sort By Confidence
inspect(sort(groceryrules, by="confidence")[1:20])

#Removing Redundant Rules
#get subset rules in vector
subsetRules <- which(colSums(is.subset(groceryrules, groceryrules)) > 1)
length(subsetRules)
# remove subset rules.
finalgroceryrules <- groceryrules[-subsetRules]
inspect(sort(finalgroceryrules, by="confidence")[1:9])
plot(finalgroceryrules, method = "graph", interactive = T)

#For Organic Strawberries----------------------------------------
#Runs apriori algorithm for rules
```

```
[1] 11
> # remove subset rules.
> finalgroceryrules <- groceryrules[-subsetRules]
> inspect(sort(finalgroceryrules, by="confidence")[1:5])
    lhs                                rhs      support     confidence lift     count
[1] {Broccoli Crown,Cucumber Kirby} => {Banana} 0.001712401 0.6666667  2.630957 122
[2] {Broccoli Crown,Strawberries}   => {Banana} 0.001263247 0.6666667  2.630957  90
[3] {Cucumber Kirby,Strawberries}   => {Banana} 0.003045828 0.6253602  2.467944 217
[4] {Bartlett Pears}                => {Banana} 0.006007439 0.5854993  2.310635 428
[5] {Organic Avocado,Strawberries}  => {Banana} 0.004449435 0.5816514  2.295450 317
> inspect(sort(finalgroceryrules, by="confidence")[1:9])
    lhs                                 rhs      support     confidence lift     count
[1] {Broccoli Crown,Cucumber Kirby}  => {Banana} 0.001712401 0.6666667  2.630957  122
[2] {Broccoli Crown,Strawberries}    => {Banana} 0.001263247 0.6666667  2.630957   90
[3] {Cucumber Kirby,Strawberries}    => {Banana} 0.003045828 0.6253602  2.467944  217
[4] {Bartlett Pears}                 => {Banana} 0.006007439 0.5854993  2.310635  428
[5] {Organic Avocado,Strawberries}   => {Banana} 0.004449435 0.5816514  2.295450  317
[6] {Broccoli Crown,Organic Avocado} => {Banana} 0.002793178 0.5685714  2.243830  199
[7] {Cucumber Kirby,Organic Avocado} => {Banana} 0.005431960 0.5608696  2.213436  387
[8] {Organic Fuji Apple}             => {Banana} 0.018036353 0.5351937  2.112107 1285
[9] {Honeycrisp Apple}               => {Banana} 0.016997684 0.5201890  2.052892 1211
>
```

Appendix: MBA Statistics

```
#For Organic Strawberries----------------------------------------
#Runs apriori algorithm for rules
groceryrules <- apriori(transactions, parameter = list(support = 0.001, confidence = 0.580, minlen=2, maxlen=4),appearance = list (lhs = c("Bag of Organic Bananas", "Organic R

#Gives Number of Total Rules
summary(groceryrules)
#Sort By Confidence
inspect(sort(groceryrules, by="confidence")[1:10])

#Removing Redundant Rules
#get subset rules in vector
subsetRules <- which(colSums(is.subset(groceryrules, groceryrules)) > 1) # get subset rules in vector
length(subsetRules)
# remove subset rules.
finalgroceryrules <- groceryrules[-subsetRules]
inspect(sort(finalgroceryrules, by="confidence")[1:7])
plot(finalgroceryrules, method = "graph", interactive = T)

#For Organic Bag of Bananas----------------------------------------
#Runs apriori algorithm for rules
```

```
> subsetRules <- which(colSums(is.subset(groceryrules, groceryrules)) > 1) # get subset rules in vector
> length(subsetRules)
[1] 4
> # remove subset rules.
> finalgroceryrules <- groceryrules[-subsetRules]
> inspect(sort(finalgroceryrules, by="confidence")[1:5])
    lhs                                                    rhs                    support     confidence lift     count
[1] {Organic Baby Spinach,Organic Hass Avocado,Organic Kiwi}           => {Organic Strawberries} 0.001024633 0.6460177  5.182472  73
[2] {Organic Kiwi,Organic Raspberries}                                 => {Organic Strawberries} 0.002273844 0.6206897  4.979285 162
[3] {Apple Honeycrisp Organic,Organic Kiwi}                            => {Organic Strawberries} 0.001052705 0.6147541  4.931669  75
[4] {Organic Cucumber,Organic Raspberries}                             => {Organic Strawberries} 0.002765106 0.6080247  4.877685 197
[5] {Apple Honeycrisp Organic,Organic Baby Spinach,Organic Raspberries} => {Organic Strawberries} 0.001010597 0.6050420  4.853757  72
> inspect(sort(finalgroceryrules, by="confidence")[1:7])
    lhs                                                    rhs                    support     confidence lift     count
[1] {Organic Baby Spinach,Organic Hass Avocado,Organic Kiwi}           => {Organic Strawberries} 0.001024633 0.6460177  5.182472  73
[2] {Organic Kiwi,Organic Raspberries}                                 => {Organic Strawberries} 0.002273844 0.6206897  4.979285 162
[3] {Apple Honeycrisp Organic,Organic Kiwi}                            => {Organic Strawberries} 0.001052705 0.6147541  4.931669  75
[4] {Organic Cucumber,Organic Raspberries}                             => {Organic Strawberries} 0.002765106 0.6080247  4.877685 197
[5] {Apple Honeycrisp Organic,Organic Baby Spinach,Organic Raspberries}   => {Organic Strawberries} 0.001010597 0.6050420  4.853757  72
[6] {Apple Honeycrisp Organic,Organic Hass Avocado,Organic Raspberries}   => {Organic Strawberries} 0.001347463 0.5962733  4.783413  96
[7] {Apple Honeycrisp Organic,Bag of Organic Bananas,Organic Raspberries} => {Organic Strawberries} 0.001698365 0.5845411  4.689295 121
>
```

Appendix: MBA Statistics

```
newsubmission1 <- subset(newsubmission, select=c(order_id, product_name))
write.csv(newsubmission1, file = "Downloads/instacart/Manjiri_ICPN_output.csv", row.names = F)

#Install arules and use for MBA
require(arules)
require("arulesViz")
manjiridata <- fread("Downloads/instacart/Manjiri_ICPN_output.csv")

#Remove OrderID for MBA
maniiridata <- maniiridata %>% select(-order_id)
```

```
> summary(transactions)
transactions as itemMatrix in sparse format with
 71245 rows (elements/itemsets/transactions) and
 22129 columns (items) and a density of 0.0003589908

most frequent items:
                Banana Bag of Organic Bananas   Organic Strawberries   Organic Baby Spinach   Organic Hass Avocado                (Other)
                 18053                 13566                   8881                   8641                   6554                 510283

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35
6470 7467 7062 6689 5862 5189 4481 3901 3429 2935 2413 2033 1795 1541 1337 1141  954  822  702  679  499  479  396  387  318  306  255  210  204  183  133   97  115  104   78
  36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56   57   58   59   61   62   63   64   65   66   69   70   71   77   78
  81   68   60   48   44   44   29   27   27   18   17   13   14   12   11    9    6    5    8    8    4    3    4    3    3    2    1    1    1    1    1    1    1    1    1
  81   82
   1    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   6.000   7.944  10.000  82.000

includes extended item information - examples:
                                                labels
1          (70% Juice!) Mountain Raspberry Juice Squeeze
2                    \\""Constant Comment\\"" Black Tea
3 \\""Constant Comment\\"" Decaffeinated Black Tea Blend

includes extended transaction information - examples:
  transactionID
1       1000014
2       1000078
3       1000126
>
```