CSML1000 Winter 2020, Group 8, Assignment2: Market Basket Prediction Model

Steven Wang, Tarun Bagga, Paul Doucet, Jerry Khidaroo, Nikola Stevanovic

2/15/2020

Contents

Load Libraries

```
# Load packages
# library('ggplot2') # visualization
# library('ggthemes') # visualization
# # library('scales') # visualization
# library('dplyr') # data manipulation
# library('caret')
# library('caret')
# library('readr')
# library('data.table')
# library('knitr')
# library('purrr') # ANN modeling functionality
# library('arules')

library(tidyverse)
library(readxl)
library(knitr)
```

```r
library(ggplot2)
library(lubridate)
library(arules)
library(arulesViz)
library(plyr)
```

1. Business Understanding

- Business Problem: The Business problem we are solving is how to increase the number of items ordered by a customer in each order submitted in the online store. To acheive this goal we want to build a machine learning model that predicts the most likely items a customer will add to their basket next based on their current selection and display those items as suggestions for quick access.
- Project Plan:
  - Select instacart dataset from kaggle to solve this unsupervised machine learning use case and in which we dont have any label or target provided and we are supose to create our own labels.
  - Load and get an understanding of the datasets, perform Exploratory Data Analysis on its features.
  - Transform the dataset as per the need to enable learning algorithms to be run on the data.
  - Identify the features of the dataset that are important in predicting the output recomendations.
  - Build and evaluate 2 Models by appling unsupervised machine learning algoritms to the dataset as appropiate and testing them.
  - Identify the best model to use for the project.
  - Build a shiny app that deploys the selected model with a user interface that allows end users to add products to a basket and get a list of recomended products.
  - Identify any ethical considerations that should be addressed at each stage of the process.
- Business Success Criteria: Success shall be achieved if we can facilitate customers to easily select items for their shopping cart with the help of artifical intelligence by modelling on their previous selections.
- Ethical Framework Questions:
  - How could your system negatively impact individuals? This is very important question to ask as we have to look at how our recommendations are not creating biases or undesired negative results for the customers. We could recommend a product to an actor which could be against their religious beliefs, could be illegal or dangerous to consume due to various factors such as dibates, allergens or pregnancy. We need to be cognizant of the fact that our association rules are ethically correct when created, if the customer does not have self control and keeps ordering an item which is not right for them, it potentially can hurt them financially and in other ways.
  - Who is most vulnerable and why? The most vulnerable would be actors who are not able to comprehend that certain recommendations are not good for them and they go ahead and buy these anyways.
  - How much error in predictions can your business accept for this use case? Since these recommendations can be dangerous and life threatening in some cases. The error can come from biased outcomes via the association rules which can heavily tilt the recommendation one way or the other.
  - Will you need to explain which input factors had the greatest influence on outputs? Since this is association rules based use case it is not that imperative to explain input factors affecting model.
  - Do you need PII or can you provide group-level data? The analysis requires customers personal data however any PII can be anonymised

2. Data Understanding

- Ethical Framework Questions:
  - Have you de-identified your data and taken measures to reduce the probability of reidentification? The data is de-identified.
  - Will socially sensitive features like gender or ethnic background influence outputs? No demographic data is present.
  - Are seemingly harmless features like location hiding proxies for socially sensitive features? No demographic data is present.

2.1 Get Data Files

- For this assignment we came across classic kaggle problem from instacart for predicting next item in the basket using market basket modelling. We looked at the data. The Dataset used is obtained from: https://www.kaggle.com/c/instacart-market-basket-analysis

2.2 Initial Data Collection Report:

- There are seven files provided as part of the dataset for this unsupervised learning model:

  1. aisles.csv:
  2. departmenst.csv:
  3. order_products___prior.csv:
  4. order_products___traing.csv:
  5. orders.csv
  6. products.csv
  7. sample_submission.csv (This file will not be used in the analysis since it is only relevant for submission to the kaggle competition where the data was sourced.).

  Each of the datset contains different type of data related to the online grocery mart with PK/FK relationships.

  Lets exploreand deep dive within the various datasets provided.

Products

For this section there are 2 CSV files provided, namely order_products_train and order_products_prior. These data inside the files specifies which products were purchased in each order. That is, order_products_prior contains previous order products for all customers and order_products_train contains the latest order products for some customers only.

Let us look the counts of records inside the files. Using the built-in R functions we can see that there are 1,384,617 products in the order_products_train file and 32,434,489 products in the order_products_prior file. Both the CSV files contain 4 features: The ID of the order (order_id) The ID of the product (product_id) The ordering of that product in the order (add_to_cart_order) Whether that product was reordered (reordered). Overall, there are 3,346,083 unique orders for 49,685 unique products.

Orders

Upon perusing the data for Instacart orders. The records inside the orders.csv file present a different tale as we see that there are 3,421,083 orders and 7 feature columns: The ID of the order (order_id) The ID of the customer (user_id) Which evaluation datasets that the order is in — prior, train, or test (eval_set) The

number of the order (order_number) The day of the week when that order occurred (order_dow) The hour of the day when that order occurred (order_hour_of_day) The number of days since the previous order (days_since_prior_order)

Departments and Aisles

Let's look at the most important departments, sorted by the number of products. The top 5 departments are Personal Care (6,563), Snacks (6,264), Pantry (5,371), Beverages (4,365), and Frozen (4,007).

Let's look at the most important aisles over all departments, as being sorted by the number of products. Ignoring the 'missing' values, we have the top 5 aisles being Candy Chocolate (1,258), Ice Cream (1,091), Vitamins Supplements (1,038), Yogurt (1,026), and Chips Pretzels (989).

2.3 Load and check data The links to the two big files are as follows: orders https://drive.google.com/open?id=1yuLqdtJKRBYs5seavGmjNvNfzh7_bQxs order products_prior https://drive.google.com/open?id=1uS7Kwwrr4AjzKvwbJSARNxWl1Ql7GRiP

```
orders = read.csv("./input/orders.csv")
aisles = read.csv("./input/aisles.csv")
departments = read.csv("./input/departments.csv")
products = read.csv("./input/products.csv")
order_products_prior = read.csv("./input/order_products__prior.csv")
order_products_train = read.csv("./input/order_products__train.csv")
```

Here we can check the Data visually by putting kable function around the head of datasets

```
# check data
kable(head(aisles))
```

aisle_id	aisle
1	prepared soups salads
2	specialty cheeses
3	energy granola bars
4	instant foods
5	marinades meat preparation
6	other

```
kable(departments)
```

department_id	department
1	frozen
2	other
3	bakery
4	produce
5	alcohol
6	international
7	beverages
8	pets
9	dry goods pasta

department_id	department
10	bulk
11	personal care
12	meat seafood
13	pantry
14	breakfast
15	canned goods
16	dairy eggs
17	household
18	babies
19	snacks
20	deli
21	missing

```
kable(head(products))
```

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

```
kable(head(orders))
```

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

```
kable(head(order_products_prior))
```

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

```
kable(head(order_products_train))
```

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
## 3 Data P	reparation, C	leaning and Manipula	tion

While looking for outliers and nulls we found out thet There were no null or empty values. This was alsoa consistent discovery for the variables in aisle, departments, Order_product_prior,order_product_train and products datasets. Although we did see that in Orders dataset we encountered some null values in days since prior order column and only 5% of the values were found to be missing. We have not imputed or rejected these since the count is very low to be a significant issue.

We also know from the data file descriptions from the source website that order_products_train has the most recent procuct orders by the user and order_products_prior contains all of the historical product orders.

3.1 Data Modification Due to data storage and processing constraints this analysis will use a random sample of the dataset available rather than all records.

Get a sample of 10000 'most recent' orders

```
# Set seed for reproducibility
set.seed(1234)
orders_train <- subset(orders, eval_set=='train')
orders_train_sample = orders_train[sample(nrow(orders_train), 10000), ]
#write.csv(orders_sample, "./input/orders_train_sample.csv", row.names = F)
```

merge order_products_prior with products and orders tables

```
# Merge products columns into order_products_prior
tmp <- merge(order_products_prior, products, by.x="product_id", by.y="product_id")

# Merge Orders columns into order_products_prior
orders_prior <- subset(orders, eval_set=='prior')
orders_products_prior_merged <- merge(tmp, orders_prior, by.x="order_id", by.y="order_id")
```

Filter orders_products_prior_merged based on user_ids in orders_train_sample

```
order_products_prior_sample = subset(orders_products_prior_merged, user_id %in% orders_train_sample$use
```

merge order_products_train with products and orders tables

```
# Merge products columns into order_products_train
tmp <- merge(order_products_train, products, by.x="product_id", by.y="product_id")

# Merge Orders columns into order_products_train
orders_products_train_merged <- merge(tmp, orders_train, by.x="order_id", by.y="order_id")
```

Filter order_product_train based on user_ids in orders_sample

```
order_products_train_sample = subset(orders_products_train_merged, user_id %in% orders_train_sample$use
```

Since we will be performing an unsupervised machine learning analysis on the dataset, records from order_products_prior_sample and order_products_train_sample can be combined.

Join order_products_prior_sample and order_products_train_sample into a single dataset

```
order_products_sample_combined <- rbind(order_products_prior_sample, order_products_train_sample)

# Lets look at the merged sample dataset we will proceed with in our analysis
kable(head(order_products_sample_combined, 12))
```

	order_id	product_id	add_to_cart_order	reordered	product_name
234	26	25890	4	0	Boneless Skinless Chicken Breasts
235	26	40545	7	0	Berry Medley
236	26	21903	6	0	Organic Baby Spinach
237	26	47766	8	1	Organic Avocado
238	26	46206	3	0	Red Grapefruit
239	26	24852	2	1	Banana
240	26	35951	1	0	Organic Unsweetened Almond Milk
241	26	33120	5	0	Organic Egg Whites
739	87	40338	9	1	Leafy Green Romaine Lettuce
740	87	2707	7	1	Completeâ„¢ ActionPacsâ„¢ Fresh Scent Dishwas
741	87	4105	13	0	Ion4 Mountain Berry Blast
742	87	31748	3	0	Cantaloupe Grapefruit Sparkling Water
#### 3	.2 Feature	Engineering			

- For the Market Basket analysis we will need 3 data columns in order to build a transaction file:

- A Customer identity field, an Transaction field, and a Product Name field

- From looking at the features we have orders$user_idforCustomeridentity, orders$order_id for Transaction, and products$product_name for Product Name.

```
# Select the 3 columns we need to create the transactions dataset and inspect
mydata <- subset(order_products_sample_combined, select = c(user_id, order_id, product_name))
kable(head(mydata, 12))
```

	user_id	order_id	product_name
234	153404	26	Boneless Skinless Chicken Breasts
235	153404	26	Berry Medley
236	153404	26	Organic Baby Spinach
237	153404	26	Organic Avocado
238	153404	26	Red Grapefruit
239	153404	26	Banana
240	153404	26	Organic Unsweetened Almond Milk
241	153404	26	Organic Egg Whites
739	155476	87	Leafy Green Romaine Lettuce
740	155476	87	Completeâ„¢ ActionPacsâ„¢ Fresh Scent Dishwasher Detergent

	user_id	order_id	product_name
741	155476	87	Ion4 Mountain Berry Blast
742	155476	87	Cantaloupe Grapefruit Sparkling Water

These 2 files will now be saved to disk for future re-loading during project development

```
# Save files
write.csv(order_products_sample_combined, "./input/order_products_sample_combined.csv", row.names = F)
write.csv(mydata, "./input/mydata.csv", row.names = F)
```

Create Transaction Table

```
# Create Transaction Table
detach("package:plyr", unload=TRUE)

retail_sorted <- mydata[complete.cases(mydata),]
retail_sorted <- retail_sorted[order(retail_sorted$user_id),]
library(plyr)
itemList <- ddply(retail_sorted,c("user_id","order_id"),
                  function(df1)paste(df1$product_name,
                                     collapse = ","))

# Remove user and order id's from the list
itemList$user_id <- NULL
itemList$order_id <- NULL
colnames(itemList) <- c("Items")
```

```
# Save Transaction File
write.csv(itemList,"./input/InstaCart_MBA.csv", quote = FALSE, row.names = TRUE)
```

- Clear up some memory

```
rm(orders)
rm(order_products_prior)
rm(order_products_train)
rm(orders_products_train_merged)
rm(orders_train)
rm(orders_train_sample)
rm(order_products_prior_sample)
rm(order_products_train_sample)
rm(orders_products_prior_merged)
rm(orders_prior)
rm(order_products_sample_combined)
rm(mydata)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)   max used   (Mb)
## Ncells  4715880 251.9   40961174 2187.6   67138536 3585.6
## Vcells 56043327 427.6  857005007 6538.5 1067587244 8145.1
```

Once the analysis data files are saved the data preparation steps above can be commented out and we can just:

Load the saved files

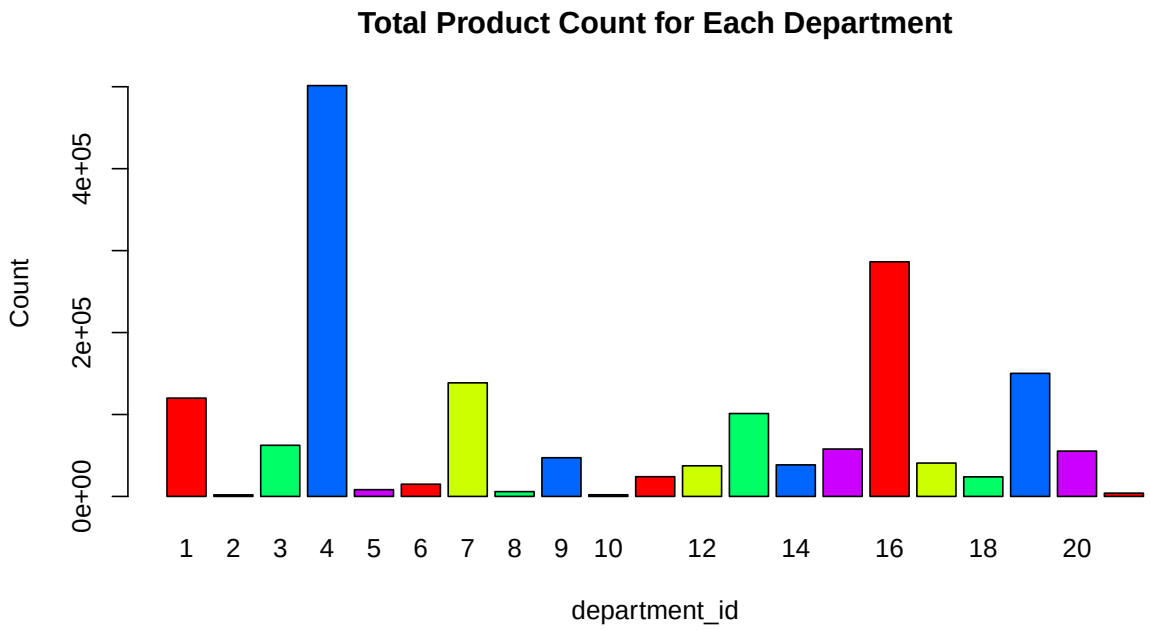
```
order_products_sample_combined <- read.csv("./input/order_products_sample_combined.csv")
mydata <- read.csv("./input/mydata.csv")
```

4. Data Understanding (Continued)

Back to some more detailed examination of the data now that we have our analysis dataset in place

Take a look at data distribution by Dept. id.

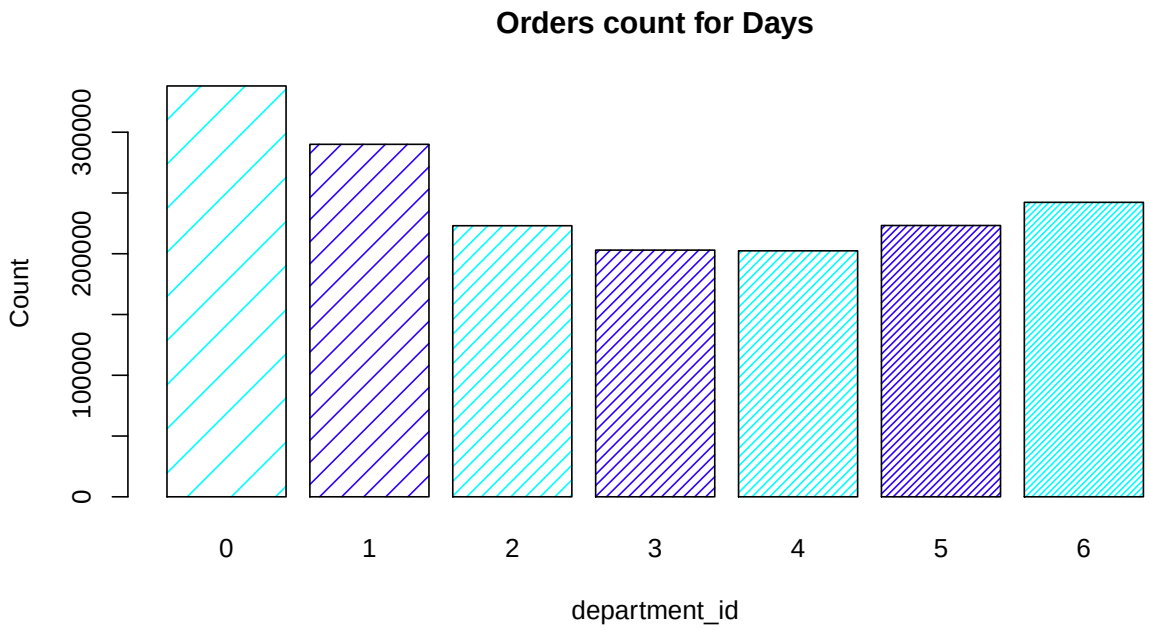
```
a=table(order_products_sample_combined$department_id)
barplot(a,main="Total Product Count for Each Department",
        ylab="Count",
        xlab="department_id",
        col=rainbow(5),
        #legend=rownames(a)
        )
```



Take a look at data distribution by Day of Week.

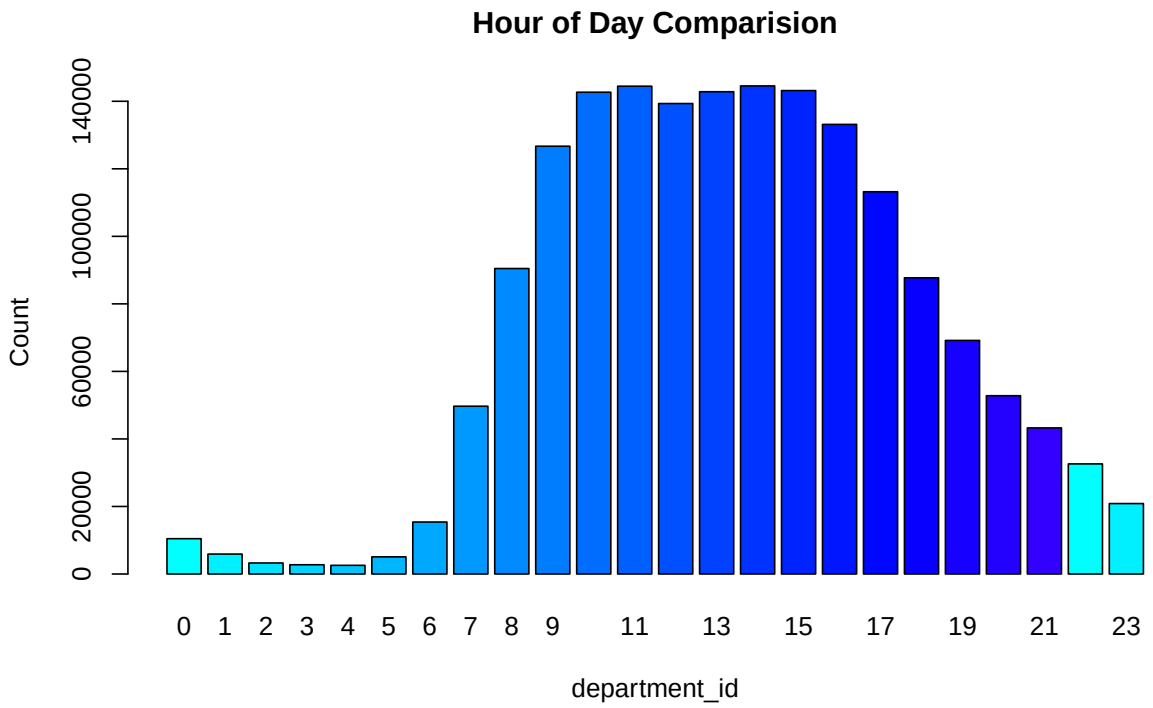
```
density1 <- seq(5,35,length.out=7)
a=table(order_products_sample_combined$order_dow)
wth <- c(rep(5, each=7))
barplot(a,main="Orders count for Days",
        ylab="Count",
        xlab="department_id",
```

```
        col=rainbow(2, start = 0.5, end = 0.7),
        density=density1,
        #legend=rownames(a),
        width = wth)
```



This plot displays the count of orders per sat of the week ###### Take a look at data distribution by Hour of Day.

```
a=table(order_products_sample_combined$order_hour_of_day)
barplot(a,main="Hour of Day Comparision",
        ylab="Count",
        xlab="department_id",
        col=rainbow(22, start = 0.5, end = 0.7),
        #legend=rownames(a),
        #width = rep(2, each=24)
        )
```




```
# tmp <- retail %>%
#   group_by(product_id, product_name) %>%
#   summarize(count = n()) %>%
#   arrange(desc(count))
# tmp <- head(tmp, n=10)
# tmp
# tmp %>%
#   ggplot(aes(x=reorder(product_name,count), y=count))+
#   geom_bar(stat="identity",fill="indian red")+
#   coord_flip()
```

5. Data Modeling

- Ethical Framework Questions:
  - Does your use case require a more interpretable algorithm? Our algorithm uses predictive measures such as lift and confidence to help the users in gauging the intepretibility of the results and the model.
  - Should you be optimizing for a different outcome than accuracy to make your outcomes fairer? Since there is no demographic data, fairness would be hard to determine, the customer data is already masked.
  - Is it possible that a malicious actor has compromised training data and created misleading results? No. The data is from a reputable source such as Kaggle and it is a well know use case.

```r
# Load Transaction data
suppressWarnings(
tr <- read.transactions('./input/InstaCart_MBA.csv', format = 'basket', sep=',')
)
```

5.1 Data Modeling - Apriori 5.1.1 Build Model - Apriori

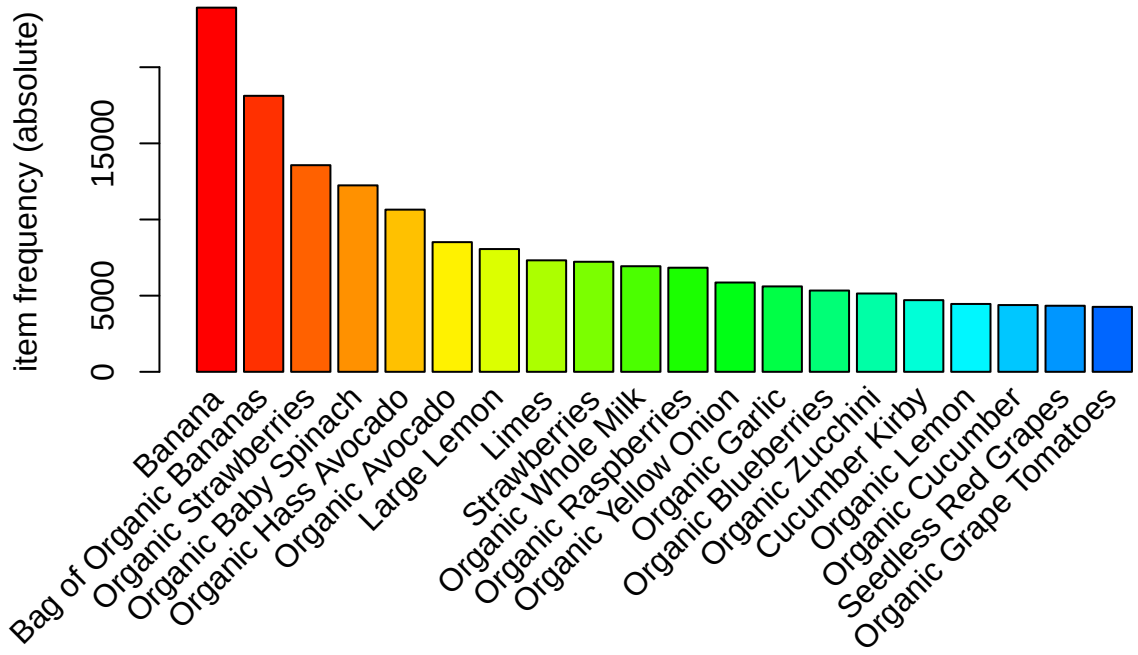
```r
tr
```

```
## transactions in sparse format with
##  168855 transactions (rows) and
##  218157 items (columns)
```

```r
summary(tr)
```

```
## transactions as itemMatrix in sparse format with
##  168855 rows (elements/itemsets/transactions) and
##  218157 columns (items) and a density of 5.038013e-05
##
## most frequent items:
##                 Banana Bag of Organic Bananas   Organic Strawberries
##                  23912                  18122                  13568
##   Organic Baby Spinach   Organic Hass Avocado                (Other)
##                  12242                  10650                1777354
##
## element (itemset/transaction) length distribution:
## sizes
##     1     2     3     4     5     6     7     8     9    10    11    12    13
##     1  9446 10644 11247 11757 11978 11738 11206 10146  9410  8489  7505  6551
##    14    15    16    17    18    19    20    21    22    23    24    25    26
##  6010  5343  4724  4164  3656  3272  2890  2437  2240  1902  1695  1453  1271
##    27    28    29    30    31    32    33    34    35    36    37    38    39
##  1089   918   827   727   619   498   471   378   326   254   224   204   178
##    40    41    42    43    44    45    46    47    48    49    50    51    52
##   152   115    97    80    71    82    43    51    28    32    17    29    17
##    53    54    55    56    57    58    59    60    61    62    63    64    65
##    19    15    13    16    11     9     8     5     3     4     6     3     4
##    66    67    68    69    70    71    73    74    75    77    78    79    82
##     8     4     2     3     4     1     1     4     1     4     1     1     1
##    96    99
##     1     1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    5.00    9.00   10.99   15.00   99.00
##
## includes extended item information - examples:
##                          labels
## 1                            #2
## 2            #2 Coffee Filters
## 3 #2 Cone White Coffee Filters
```

```
itemFrequencyPlot(tr, topN=20, type='absolute', col=rainbow(20, start = 0, end = 0.6))
```




```
# Training Apriori on the dataset
rules <- apriori(tr, parameter = list(supp=0.001, conf=0.5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 168
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[218157 item(s), 168855 transaction(s)] done [0.92s].
## sorting and recoding items ... [1779 item(s)] done [0.04s].
## creating transaction tree ... done [0.08s].
## checking subsets of size 1 2 3 4 done [0.06s].
## writing ... [283 rule(s)] done [0.00s].
```

```
## creating S4 object  ... done [0.03s].
```

```
rules <- sort(rules, by='confidence', decreasing = TRUE)
summary(rules)
```

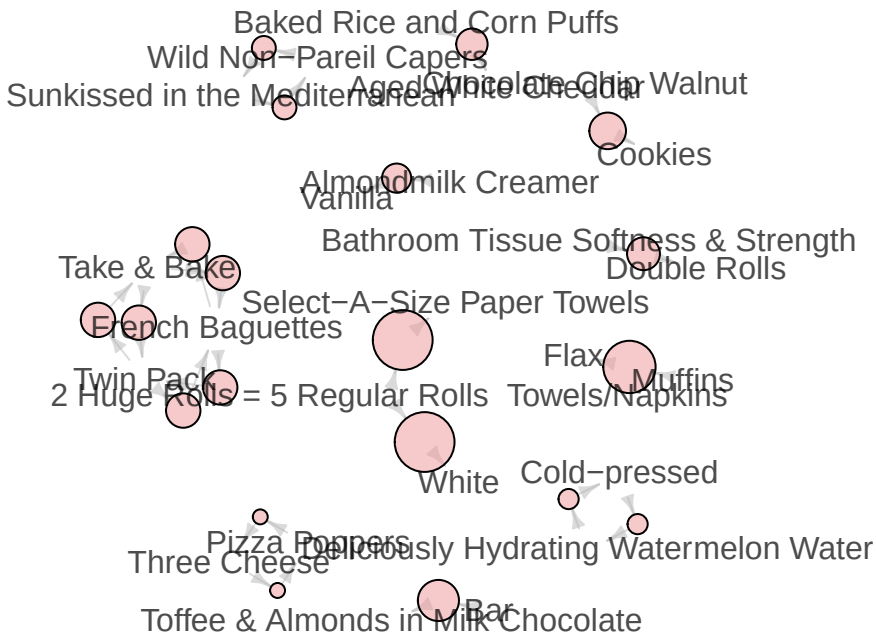
```
## set of 283 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4
## 150 120  13
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   2.516   3.000   4.000
##
## summary of quality measures:
##     support           confidence          lift            count
##  Min.   :0.001001   Min.   :0.5000   Min.   :  3.531   Min.   : 169.0
##  1st Qu.:0.001261   1st Qu.:0.8108   1st Qu.:109.546   1st Qu.: 213.0
##  Median :0.001469   Median :1.0000   Median :302.066   Median : 248.0
##  Mean   :0.002108   Mean   :0.9050   Mean   :344.026   Mean   : 355.9
##  3rd Qu.:0.002357   3rd Qu.:1.0000   3rd Qu.:516.376   3rd Qu.: 398.0
##  Max.   :0.010778   Max.   :1.0000   Max.   :999.142   Max.   :1820.0
##
## mining info:
##  data ntransactions support confidence
##    tr        168855   0.001        0.5
```

```
inspect(rules[1:20])
```

```
##      lhs                                              rhs
## [1]  {Almondmilk Creamer}                          => {Vanilla}                                0.0
## [2]  {Cold-pressed}                                => {Deliciously Hydrating Watermelon Water} 0.0
## [3]  {Deliciously Hydrating Watermelon Water}      => {Cold-pressed}                           0.0
## [4]  {Bathroom Tissue Softness & Strength}         => {Double Rolls}                           0.0
## [5]  {2 Huge Rolls = 5 Regular Rolls  Towels/Napkins} => {Select-A-Size Paper Towels}          0.0
## [6]  {2 Huge Rolls = 5 Regular Rolls  Towels/Napkins} => {White}                               0.0
## [7]  {Three Cheese}                                => {Pizza Poppers}                          0.0
## [8]  {Pizza Poppers}                               => {Three Cheese}                           0.0
## [9]  {Baked Rice and Corn Puffs}                   => {Aged White Cheddar}                     0.0
## [10] {Sunkissed in the Mediterranean}              => {Wild Non-Pareil Capers}                 0.0
## [11] {Wild Non-Pareil Capers}                      => {Sunkissed in the Mediterranean}         0.0
## [12] {French Baguettes}                            => {Twin Pack}                              0.0
## [13] {Twin Pack}                                   => {French Baguettes}                       0.0
## [14] {French Baguettes}                            => {Take & Bake}                            0.0
## [15] {Take & Bake}                                 => {French Baguettes}                       0.0
## [16] {Twin Pack}                                   => {Take & Bake}                            0.0
## [17] {Take & Bake}                                 => {Twin Pack}                              0.0
## [18] {Muffins}                                     => {Flax}                                   0.0
## [19] {Bar}                                         => {Toffee & Almonds in Milk Chocolate}     0.0
## [20] {Chocolate Chip Walnut}                       => {Cookies}                                0.0
```

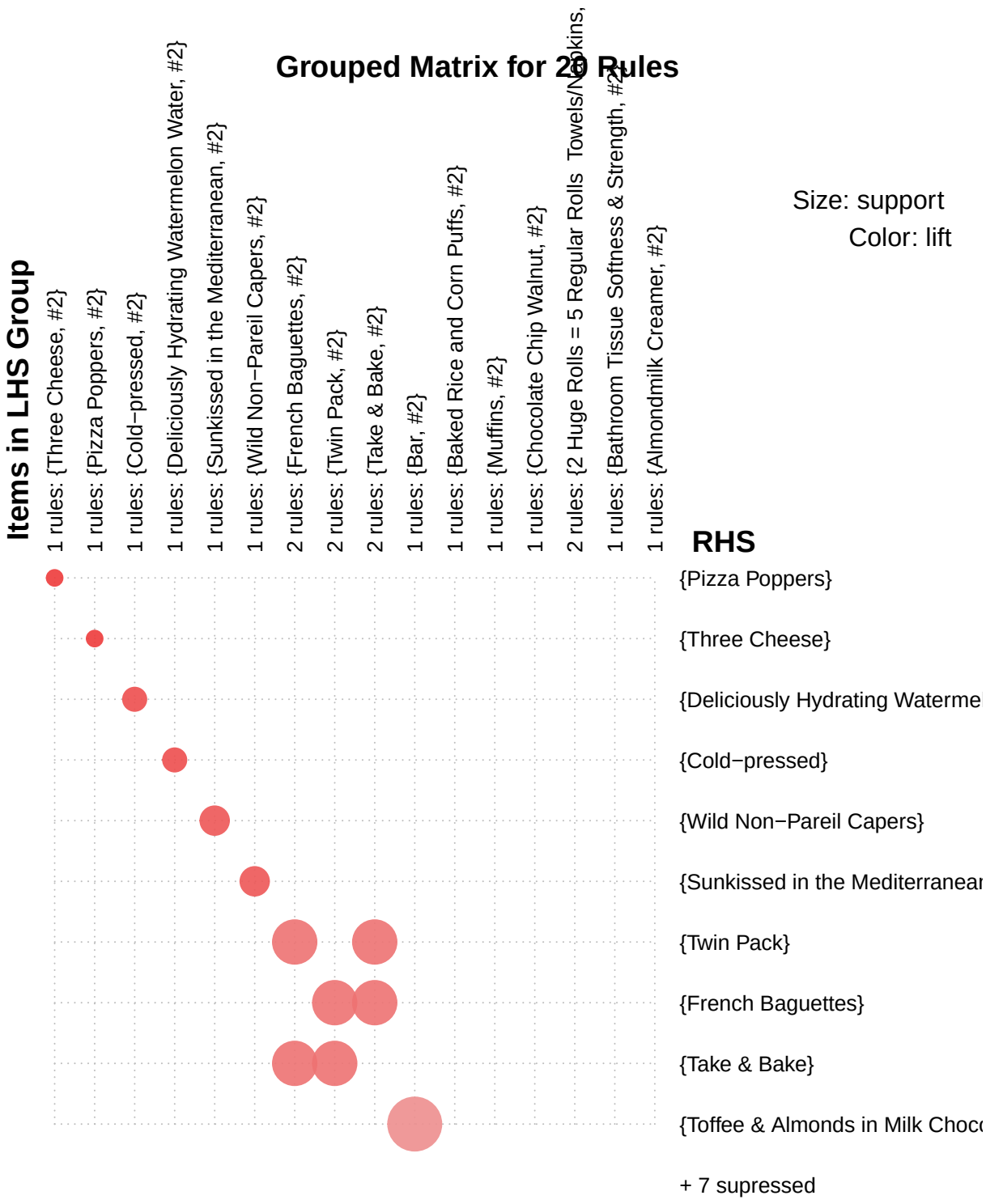
5.2 Data Evaluation - Apriori 5.2.1 Test the Model

```
topRules <- rules[1:20]
# Visualising the results
plot(topRules, method="graph", shading=NA)
```



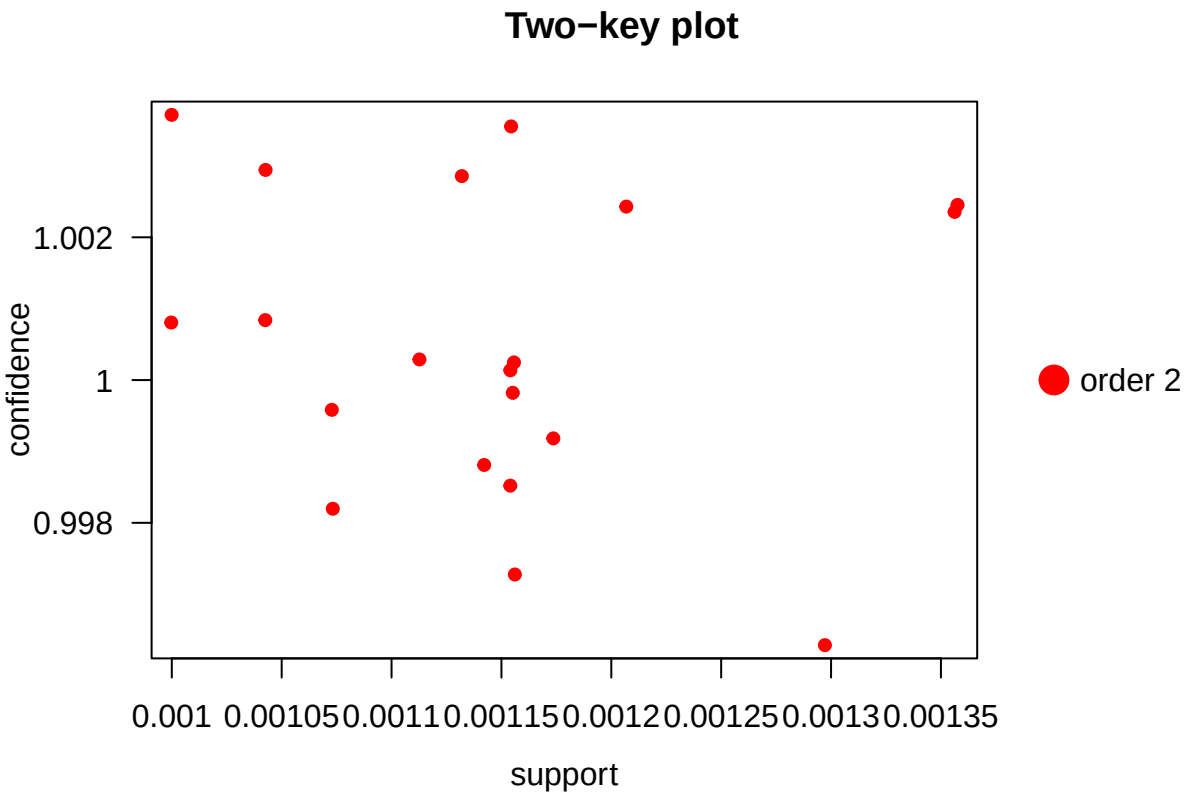
The plot shows top 20 rules of the model

```
plot(topRules, method = "grouped")
```

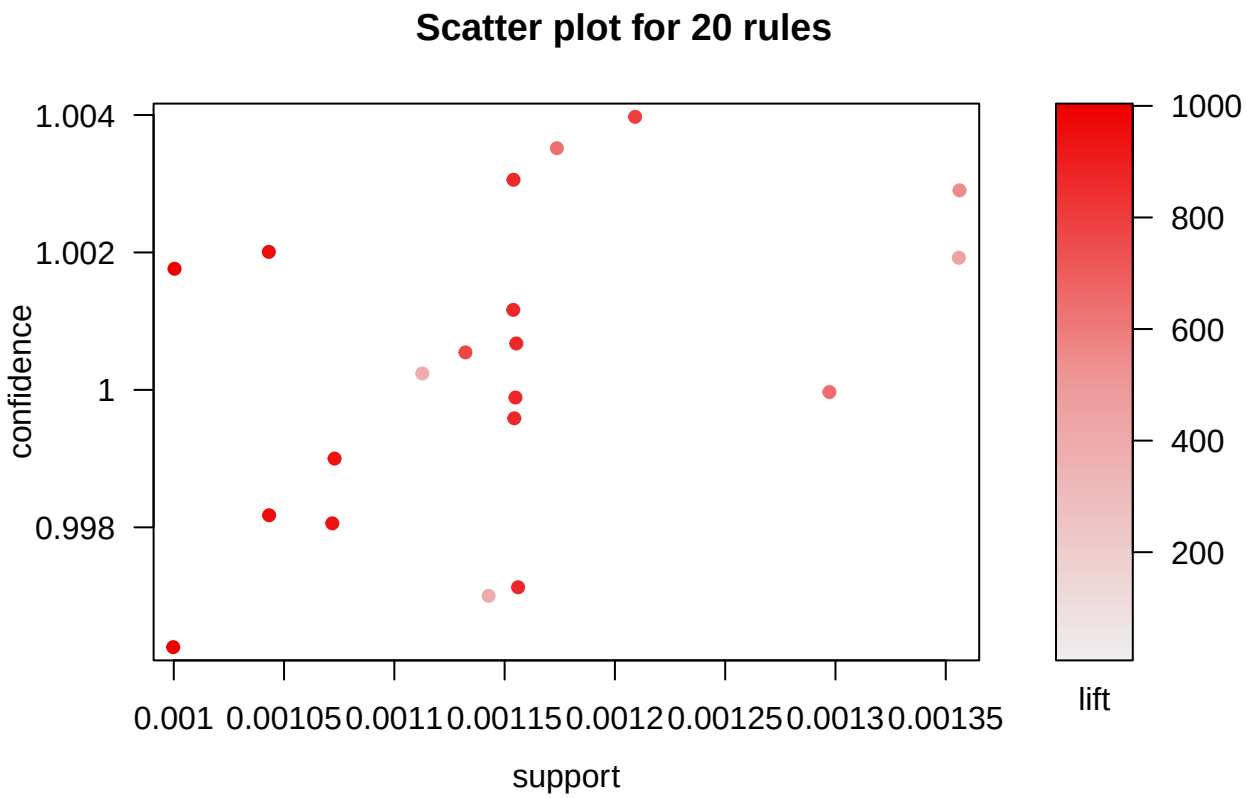


This plot is grouping the top 20 rules together.

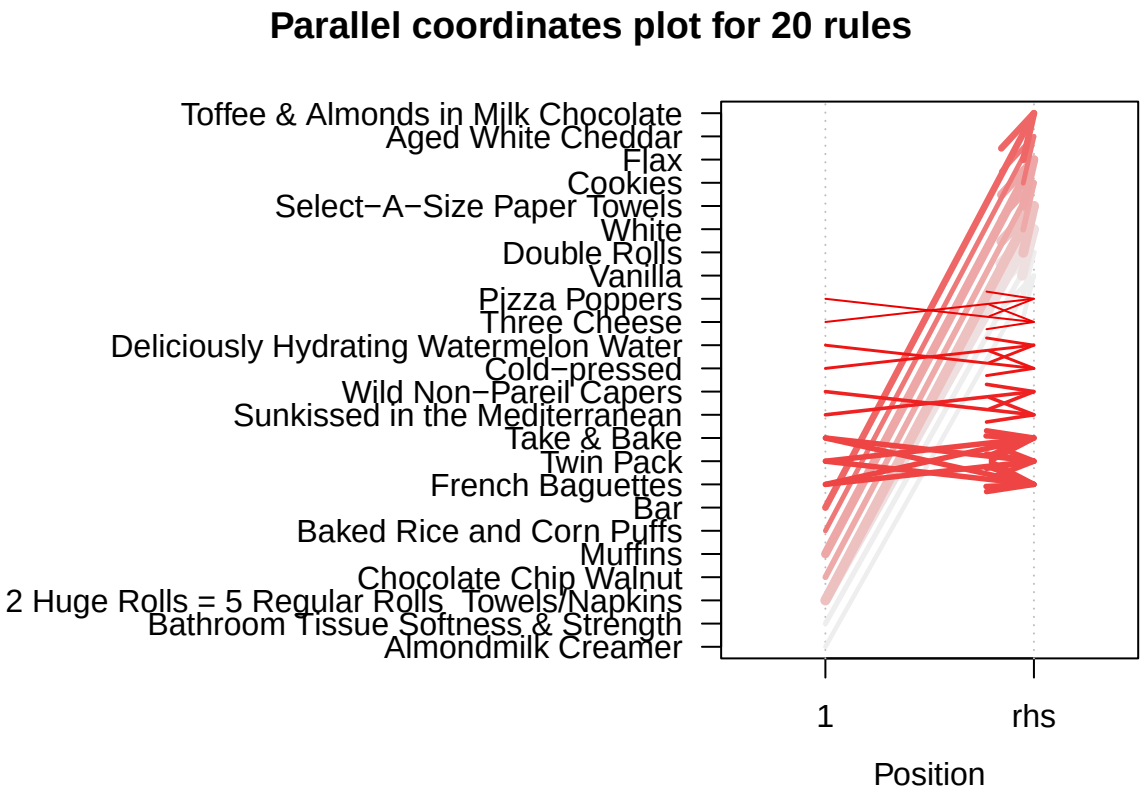
```
plot(topRules, method = "two-key plot")
```




```
plot(topRules, method = "scatterplot")
```




```
plot(topRules, method = "paracoord")
```



Plot 1 and Plot 2 are just scatterplot for first 20 rules which provides lift vs confidence. The Plot 3 above uses apriori model to showcase how products on the lhs are associated with products on the right. The rhs is same as lhs.

5.2.2 Perform a test prediction

```
grocery_item = "Milk"
rules <- apriori(tr, parameter = list(supp=0.001, conf=0.15),
                 appearance = list(default="rhs", lhs=grocery_item),
                 control = list (verbose=F))
rules_conf <- sort (rules, by="confidence", decreasing=TRUE) # 'high-confidence' rules.
result = inspect(head(rules_conf))
```

```
##     lhs       rhs                      support     confidence lift      count
## [1] {Milk} => {Vitamin D}              0.007852892 0.6707132  84.961199 1326
## [2] {Milk} => {Organic}                0.005928163 0.5063227  24.240182 1001
## [3] {Milk} => {Bag of Organic Bananas} 0.002842676 0.2427921   2.262259  480
## [4] {Milk} => {2% Milkfat}             0.002830831 0.2417805  35.749424  478
## [5] {Milk} => {Reduced Fat}            0.002357052 0.2013151  83.112629  398
## [6] {Milk} => {Banana}                 0.001847739 0.1578149   1.114412  312
```

```
result$rhs
```

```
## [1] {Vitamin D}              {Organic}                {Bag of Organic Bananas}
## [4] {2% Milkfat}             {Reduced Fat}            {Banana}
## 6 Levels: {2% Milkfat} {Bag of Organic Bananas} {Banana} ... {Vitamin D}
```

```
# Training Eclat on the dataset
eclat_itemsets = eclat(tr, parameter = list(support = 0.001, minlen = 2))
```

5.3 Data Modeling - Eclat

```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen            target   ext
##     FALSE   0.001      2     10 frequent itemsets FALSE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 168
##
## create itemset ...
## set transactions ...[218157 item(s), 168855 transaction(s)] done [0.96s].
## sorting and recoding items ... [1779 item(s)] done [0.04s].
## creating sparse bit matrix ... [1779 row(s), 168855 column(s)] done [0.02s].
## writing  ... [2184 set(s)] done [2.94s].
## Creating S4 object  ... done [0.00s].
```

```
eclat_itemsets <- sort(eclat_itemsets, by='count', decreasing = TRUE)
summary(eclat_itemsets)
```

```
## set of 2184 itemsets
##
## most frequent items:
##                 Banana Bag of Organic Bananas   Organic Strawberries
##                    346                    260                    235
##   Organic Baby Spinach   Organic Hass Avocado                 (Other)
##                    231                    185                   3374
##
## element (itemset/transaction) length distribution:sizes
##    2    3    4
## 1926  253    5
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    2.00    2.00    2.12    2.00    4.00
##
## summary of quality measures:
##     support             count
##  Min.   :0.001001   Min.   : 169.0
##  1st Qu.:0.001173   1st Qu.: 198.0
##  Median :0.001445   Median : 244.0
##  Mean   :0.001948   Mean   : 328.9
##  3rd Qu.:0.002091   3rd Qu.: 353.0
##  Max.   :0.018282   Max.   :3087.0
##
```

```
## includes transaction ID lists: FALSE
##
## mining info:
##  data ntransactions support
##    tr        168855   0.001
```

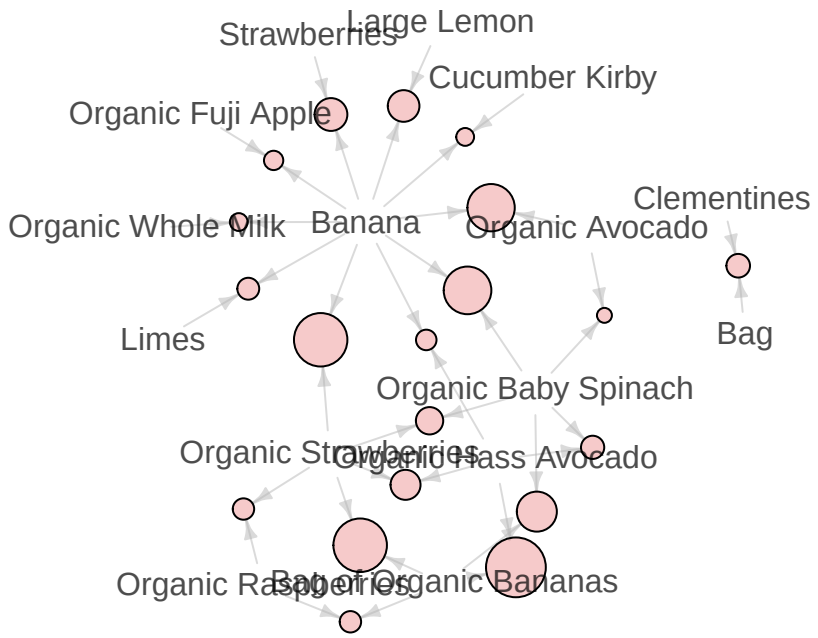
```
inspect(sort(eclat_itemsets, by = 'count')[1:20])
```

```
##      items                                          support     count
## [1]  {Bag of Organic Bananas,Organic Hass Avocado} 0.018281958 3087
## [2]  {Bag of Organic Bananas,Organic Strawberries} 0.016979065 2867
## [3]  {Banana,Organic Strawberries}                 0.016961298 2864
## [4]  {Banana,Organic Baby Spinach}                 0.015765005 2662
## [5]  {Banana,Organic Avocado}                      0.015693939 2650
## [6]  {Bag of Organic Bananas,Organic Baby Spinach} 0.014177845 2394
## [7]  {Banana,Strawberries}                         0.012685440 2142
## [8]  {Banana,Large Lemon}                          0.012442628 2101
## [9]  {Organic Hass Avocado,Organic Strawberries}   0.012099138 2043
## [10] {Organic Baby Spinach,Organic Strawberries}   0.011548370 1950
## [11] {Bag,Clementines}                             0.010778479 1820
## [12] {Organic Baby Spinach,Organic Hass Avocado}   0.010695567 1806
## [13] {Banana,Limes}                                0.010417222 1759
## [14] {Bag of Organic Bananas,Organic Raspberries}  0.010316544 1742
## [15] {Organic Raspberries,Organic Strawberries}    0.010310622 1741
## [16] {Banana,Organic Hass Avocado}                 0.010132954 1711
## [17] {Banana,Organic Fuji Apple}                   0.009884220 1669
## [18] {Banana,Cucumber Kirby}                       0.009576264 1617
## [19] {Banana,Organic Whole Milk}                   0.009570341 1616
## [20] {Organic Avocado,Organic Baby Spinach}        0.008978117 1516
```

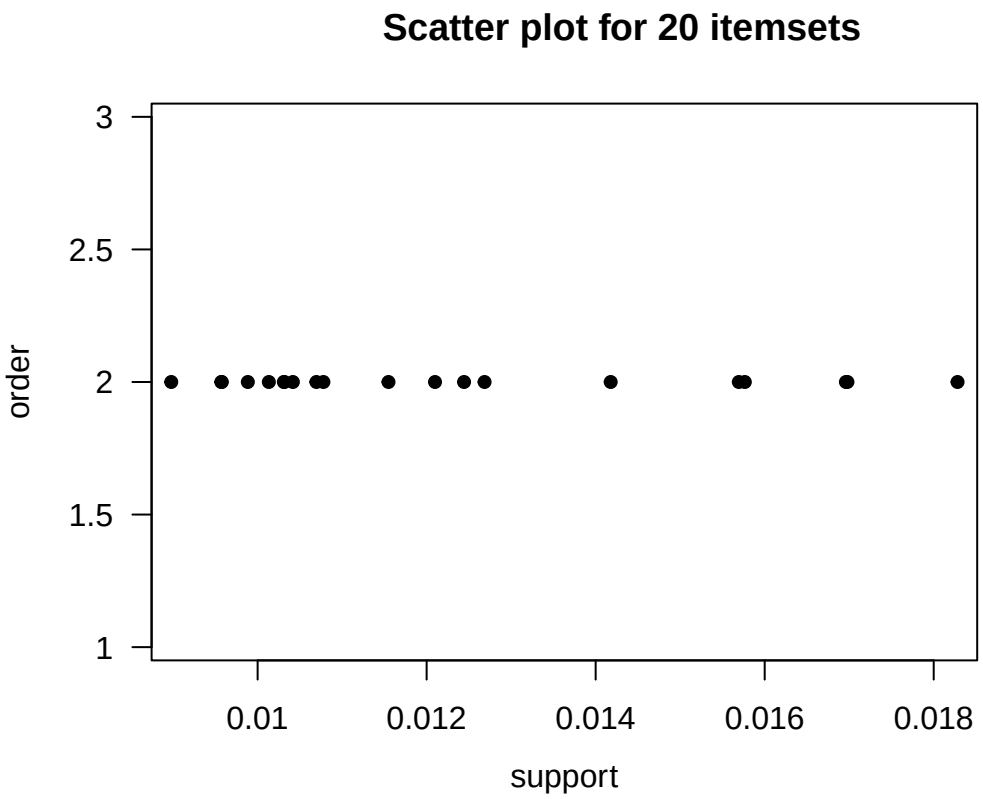
```
eclat_topItemSets <- eclat_itemsets[1:20]
```

```
# Visualising the results of the Eclat Analysis
plot(eclat_topItemSets, method="graph")
```

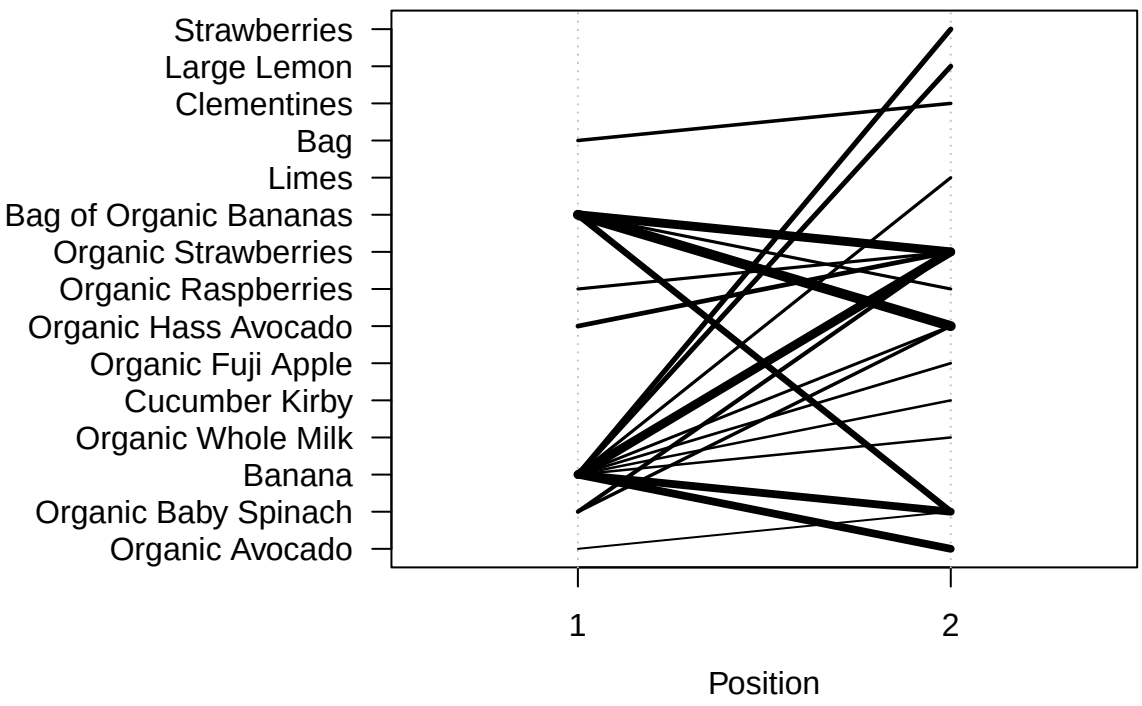
5.4 Data Evaluation - Eclat




```
plot(eclat_topItemSets, method = "scatterplot")
```




```
plot(eclat_topItemSets, method = "paracoord")
```



Plot 1 gives graph for association of products for Plot 1 Plot 2 are just scatterplot for first 20 rules which provides lift vs confidence. The Plot 3 above uses apriori model to showcase how products on the lhs are associated with products on the right. The rhs is same as lhs. ###### 5.4.1 Perform a test prediction When we test our model for prediction we see the following results

```
grocery_item = "Garlic"
# eclat_itemsets = eclat(tr, parameter = list(support = 0.001, minlen = 2),
#                  control = list (verbose=F))

## Create rules from the itemsets
eclat_rules <- ruleInduction(eclat_itemsets, tr, confidence = .9)
summary(eclat_rules)
```

```
## set of 189 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3  4
## 98 80 11
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    2.00    2.00    2.54    3.00    4.00
##
## summary of quality measures:
##     support           confidence          lift          itemset
##  Min.   :0.001001   Min.   :0.9039   Min.   : 47.5   Min.   :  24.0
```

```
##  1st Qu.:0.001261   1st Qu.:1.0000   1st Qu.:187.6   1st Qu.: 452.0
##  Median :0.001481   Median :1.0000   Median :376.9   Median :1026.0
##  Mean   :0.002088   Mean   :0.9930   Mean   :393.7   Mean   : 937.9
##  3rd Qu.:0.002357   3rd Qu.:1.0000   3rd Qu.:550.0   3rd Qu.:1416.0
##  Max.   :0.008356   Max.   :1.0000   Max.   :999.1   Max.   :2169.0
##
## mining info:
##  data ntransactions support confidence
##    tr        168855   0.001        0.9
```

```
kable(inspect(eclat_rules[1:20]))
```

```
##      lhs                                        rhs                                support confidenc
## [1]  {Organic Red Radish}                    => {Bunch}                          0.008356282  1.000000
## [2]  {Vitamin D}                             => {Milk}                           0.007852892  0.994748
## [3]  {Organic,
##       Vitamin D}                             => {Milk}                           0.005495840  1.000000
## [4]  {Milk,
##       Organic}                               => {Vitamin D}                      0.005495840  0.927071
## [5]  {YoKids Squeezers Organic Low-Fat Yogurt} => {Strawberry}                   0.005341861  1.000000
## [6]  {Bibb) Lettuce}                         => {Butter}                         0.004352847  1.000000
## [7]  {Super Spinach! Baby Spinach,
##       Sweet Baby Kale}                       => {Baby Bok Choy}                  0.004157413  1.000000
## [8]  {Baby Bok Choy,
##       Sweet Baby Kale}                       => {Super Spinach! Baby Spinach}    0.004157413  1.000000
## [9]  {Baby Bok Choy,
##       Super Spinach! Baby Spinach}           => {Sweet Baby Kale}                0.004157413  1.000000
## [10] {Sweet Baby Kale}                       => {Baby Bok Choy}                  0.004157413  1.000000
## [11] {Sweet Baby Kale}                       => {Super Spinach! Baby Spinach}    0.004157413  1.000000
## [12] {Super Spinach! Baby Spinach}           => {Sweet Baby Kale}                0.004157413  1.000000
## [13] {Super Spinach! Baby Spinach}           => {Baby Bok Choy}                  0.004157413  1.000000
## [14] {Organic Milk Reduced Fat}              => {2% Milkfat}                     0.004127802  1.000000
## [15] {Butter,
##       Organic Butterhead (Boston)}           => {Bibb) Lettuce}                  0.004115957  1.000000
## [16] {Bibb) Lettuce,
##       Organic Butterhead (Boston)}           => {Butter}                         0.004115957  1.000000
## [17] {Bibb) Lettuce,
##       Butter}                                => {Organic Butterhead (Boston)}    0.004115957  0.945578
## [18] {Organic Butterhead (Boston)}           => {Butter}                         0.004115957  1.000000
## [19] {Organic Butterhead (Boston)}           => {Bibb) Lettuce}                  0.004115957  1.000000
## [20] {Bibb) Lettuce}                         => {Organic Butterhead (Boston)}    0.004115957  0.945578

## Warning in kable_markdown(x, padding = padding, ...): The table should have a
## header (column names)
```

|| || || ||

```
result_eclat <- subset(eclat_rules, subset = lhs %in% "Milk")

rules_conf <- sort (result_eclat, by="confidence", decreasing=TRUE) # 'high-confidence' rules.
result = inspect(head(rules_conf))
```

```
##     lhs                                        rhs          support
## [1] {Milk,Reduced Fat}                      => {2% Milkfat} 0.002357052
## [2] {Milk,Organic,Organic Strawberries}     => {Vitamin D}  0.001000859
## [3] {Bag of Organic Bananas,Milk,Organic}   => {Vitamin D}  0.001687839
## [4] {Milk,Organic}                          => {Vitamin D}  0.005495840
##     confidence lift     itemset
## [1] 1.0000000  147.8590  452
## [2] 0.9657143  122.3298 2169
## [3] 0.9563758  121.1469  803
## [4] 0.9270729  117.4350   62
```

```
result$rhs
```

```
## [1] {2% Milkfat} {Vitamin D}  {Vitamin D}  {Vitamin D}
## Levels: {2% Milkfat} {Vitamin D}
```

6. Final Model Analysis and Selection

6.1 Model Comparison We have used the following two models to do prediction for the instacart shopping use case.The following Machine Learning Algorithms were used in this analysis: Apriori - This is an algorithm that is used for market basket modelling. It uses techniques such as association rule learning over datasets along with item set mining. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the datsets Eclat - This modelling technique is another one in Association rule mining algorithms. This algorithm does Mining frequent itemsets using the vertical data format. (Eclat) is a method that transforms a given data set of transactions in the horizontal data format of TID-itemset into the vertical data format of item-TID_set. It mines the transformed data set by TID_set intersections based on the Apriori property and additional optimization techniques such as diffset.

While Using associative rule methodology in apriori, the model predicts for us next product sequence the customer can potentially purchase. Since it is difficult to understand nuances of the model, The results can be interpreted using Lift, support and confidence. These three terms can be defined as follows:

Support – It is defined as the percentage of transactions that comprise all of the items in a dataset. The more the support value the more frequently the product occurs. High support values are preferred for ample amount of future transactions.

Confidence - It is the probability that a transaction that contains the items on the left hand side of the also contains the item on the right hand side. The more the confidence value, the greater the likelihood that the product on the right hand side will be purchased.

Lift - The formula for Lift is always considered as the ratio of Confidence to Expected Confidence. It is the probability of all of the products in a rule occurring together by the product of the probabilities of the items on the left and right hand side occurring as if there was no association between them.

Source: https://www.lexjansen.com/sesug/2019/SESUG2019_Paper-252_Final_PDF.pdf

6.3 Selected Model: Based on the apriori associative rule methodology the models we are able to predict the reorder of products, some of the recommendations have been made.The apriori model gives highest lift and confidence and resulting in high amount of accuracy hence we choose Apriori model versus the Eclat Model.

```
#kable(inspect(rules[1:10]))
```

7. Deployment

7.1 Shiny App Url: https://csml1000-group8.shinyapps.io/Assignment2/

7.2 Summary Explanation If we at looking at how the company can benifit from the associative rules for grouping of products together we would be heading towrds at the promotional and marketing departments. It will be highly desirable to run promotional and marketing campaigns with the help of the model for the prediction of the next product. The pomotions can be in such a way that Customers can be provided additional offers or discounts on bundling the products together for a lesser price and customize the products based on the association rules.Based on the reordering model the model can ensure that products can be added to the cart automatically based on the customer preferences.We would recommend the company to add the items directly to the customer's shopping cart or to provide as suggestive list when they make their purchase in order to enhance the customer experience.

- Limitations of our analysis:
  - Due to processing and resource limitations we used a random sample of aproximately 10% of the original dataset
  - The analysis is based on data provided by InstaCart and may inherit any biases that exists in their customer base relative to the general population.
- Further steps:
  - The analysis can be expanded to include all of the original data as well as any other similar sources that may be available. We also wanted to tackle the Ethical framework for the model by adding the exclusion rules along with our association rules. The exclusion rules would allow us to protect actors from buying certain products which would either be underiable, illegal or dangerous for the actors.
- Ethical Framework Questions:
  - Can a malicious actor infer information about individuals from your system? No. There is no PII present.
  - Are you able to identify anomalous activity on your system that might indicate a security breach? This would need to be considered for each specific deployment.
  - Do you have a plan to monitor for poor performance on individuals or subgroups? N/A since No demographic data is present.
  - Do you have a plan to log and store historical predictions if a consumer requests access in the future? N/A since No demographic data is present.
  - Have you documented model retraining cycles and can you confirm that a subject's data has been removed from models? N/A since No demographic data is present.

References

Yihui Xie, J. J. Allaire, Garrett Grolemund, 2019, R Markdown: The Definitive Guide https://bookdown.org/yihui/rmarkdown/markdown-syntax.html

Jonathan McPherson, 2016, R Notebooks https://blog.rstudio.com/2016/10/05/r-notebooks

Adam Kardash, Patricia Kosseim, 2018, Responsible AI in Consumer Enterprise, integrate.ai

Roberto J. Bayardo Jr, Rakesh Agrawal, Proc. of the Fifth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, 145-154, 1999. Mining the Most Interesting Rules, https://www.bayardo.org/ps/kdd99.pdf on Feb. 28, 2020

Gopi Subramanian, R Data Analysis Projects, 2017, https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788621878/1/ch01lvl1sec10/association-rule-mining on Feb. 28, 2020

Aravind Dhanabal,2019, Market Basket Analysis on Instacart https://www.lexjansen.com/sesug/2019/SESUG2019_Paper-252_Final_PDF.pdf on Feb. 28, 2020