Table of Contents

1. Introduction

Data Science and Analytics is nowadays widely used in retail industry. With the advent of bid data tools and higher computing power, sophisticated algorithms can crunch huge volumes of transactional data to extract meaningful insights. Companies such as Kroger invest heavily to transform more than a hundred-year-old retail industry through analytics.[1]

This project is an attempt to apply unsupervised learning algorithms on the transactional data to formulate strategies to improve the sales of the products.

This project deals with online retail store data taken from UCI Machine Learning Repository[2]. The data pertains to a UK-based registered online retail store's transaction between 01/12/2010 and 09/12/2011. The retail store mostly sells different gift items to wholesalers around the globe.

The objective of the project is to apply statistical techniques such as clustering, association rules and collaborative filtering to come up with different business strategies that may lead to an increase in the sales of the products.

Microsoft Excel, R Studio and Tableau are the major tools used in this project.

2. Variable Dictionary[3]

The data contains 541909 observations and 8 columns. Following is the variable dictionary

1. **InvoiceNo:** Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
2. **StockCode:** Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
3. **Description:** Product (item) name. Nominal.
4. **Quantity:** The quantities of each product (item) per transaction. Numeric.
5. **InvoiceDate:** Invoice Date and time. Numeric, the day and time when each transaction was generated.
6. **UnitPrice:** Unit price. Numeric, Product price per unit in sterling.
7. **CustomerID:** Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
8. **Country:** Country name. Nominal, the name of the country where each customer resides.

[1] http://analytics-magazine.org/corporate-profile-advanced-analytics-kroger/

[2] http://archive.ics.uci.edu/ml/datasets/online+retail

[3] http://archive.ics.uci.edu/ml/datasets/online+retail

3. Scope

Based on the nature of the data, following analysis can be done:

3.1. Customer Segmentation:

Metrics such as FRM-Frequency, Recency and Monetary value will be used to segment the customers in different clusters using K-Means clustering algorithm in R. Statistical measures such as elbow curve based on *wss*- within sum of squares, silhouette distance method and gap statistic technique can be used to decide the optimum number of clusters.

Different strategies can be formulated for each customer segment to increase the quality of service and the revenue from the customers.

3.2. Market Basket Analysis:

The data will be transformed to transactions format to apply *apriori* algorithm called association rules that identifies association among the products that are sold by the online retail store. These association rules with higher lift will be used to define product combos or product recommendations that may lead to increase in average purchase basket size.

3.3. Recommender System:

A recommender system based on collaborative filtering using Cosine or Jaccard similarity matrix can be built. The retail store sells the products to 38 countries. Product sales in some countries is lower than average. A recommender system can be used to recommend products to be sold in any specific country based on collaborative filtering algorithm. *Recommenderlab*[4] package in R will be used to design product recommendations.

3.4. Tableau Dashboard:

An interactive tableau dashboard can be designed based on all the findings for the management team to assess the performance of the store sales.

[4] https://cran.r-project.org/web/packages/recommenderlab/index.html

4. Exploratory Data Analysis

Following part deals with EDA of important variable and the dataset.

4.1. Missing Values

Description and CustomerID are the variables that have missing values. These are taken care of in the following sections of the project. For instance, missing CustomerID observations are not considered during segmentation.

Table 1- Missing Values

Variable	Missing Values
InvoiceNo	0
StockCode	0
Description	1454
Quantity	0
InvoiceDate	0
UnitPrice	0
CustomerID	135080
Country	0

4.2. Quantity

Quantity variable has extreme values middle 50 percentile values lie between 1 and 10. For imputation, any observations with quantity that is higher than 10000 or lower than -10000 is removed. Please note, negative values of quantity reflect orders that have been cancelled.

Table 2 Quantity

Min.	1st Q	Median	Mean	3rd Q	Max
-80995	1	3	9.55	10	80995

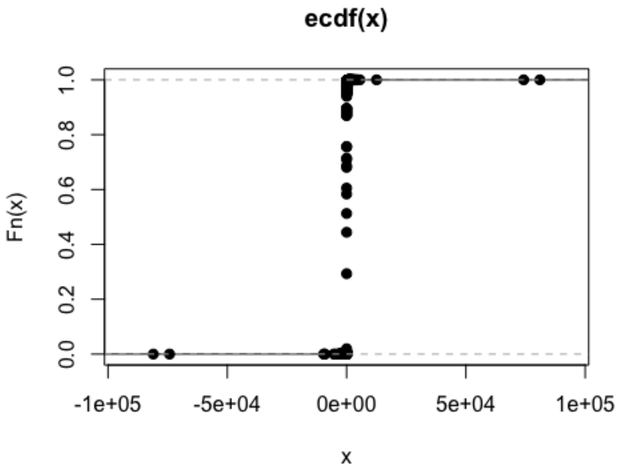


Fig 1. Empirical CDF of Quantity

4.3. Unit Price

Similar to Quantity, Unit Price has many outliers. The middle 50 percentile values lie between 1.25 and 4.13.

Table 3 Unit Price

Min.	1st Q	Median	Mean	3rd Q	Max
-11062.06	1.25	2.08	4.61	4.13	38970

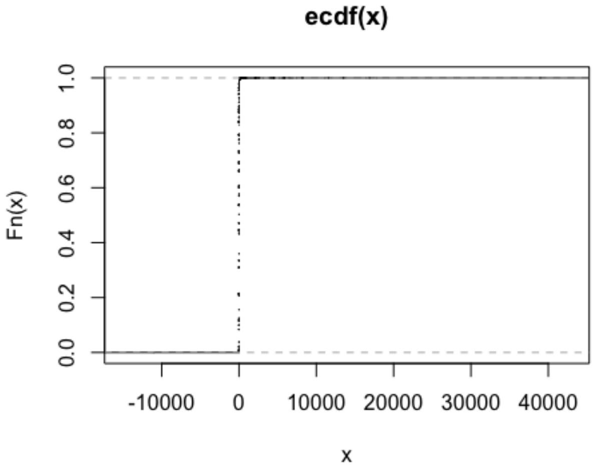
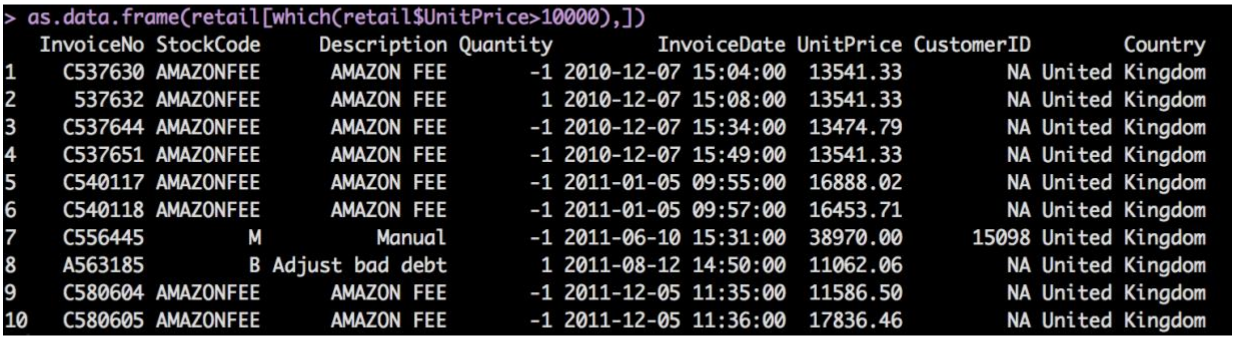


Fig 2. Empirical CDF of Unit Price

Upon investigation, it has been found that all the values where the unit price was exceptionally higher are the cancelled orders. These orders are removed from the data.



```
> as.data.frame(retail[which(retail$UnitPrice>10000),])
   InvoiceNo StockCode     Description Quantity         InvoiceDate UnitPrice CustomerID        Country
1    C537630 AMAZONFEE      AMAZON FEE       -1 2010-12-07 15:04:00  13541.33         NA United Kingdom
2     537632 AMAZONFEE      AMAZON FEE        1 2010-12-07 15:08:00  13541.33         NA United Kingdom
3    C537644 AMAZONFEE      AMAZON FEE       -1 2010-12-07 15:34:00  13474.79         NA United Kingdom
4    C537651 AMAZONFEE      AMAZON FEE       -1 2010-12-07 15:49:00  13541.33         NA United Kingdom
5    C540117 AMAZONFEE      AMAZON FEE       -1 2011-01-05 09:55:00  16888.02         NA United Kingdom
6    C540118 AMAZONFEE      AMAZON FEE       -1 2011-01-05 09:57:00  16453.71         NA United Kingdom
7    C556445         M          Manual       -1 2011-06-10 15:31:00  38970.00      15098 United Kingdom
8    A563185         B Adjust bad debt        1 2011-08-12 14:50:00  11062.06         NA United Kingdom
9    C580604 AMAZONFEE      AMAZON FEE       -1 2011-12-05 11:35:00  11586.50         NA United Kingdom
10   C580605 AMAZONFEE      AMAZON FEE       -1 2011-12-05 11:36:00  17836.46         NA United Kingdom
```

Fig 3. Exceptionally high Unit Prices

4.4. Orders and revenue trend over time

For the purpose of analysing the trend of orders and revenue over time, the orders were aggregated based on the day of the order. The trend was plotted on the graph to analyse.

Number of orders per day were higher at the end of 2010 however, it saw a dip between April 2011 and July 2011. The orders per day started growing after August 2011 and the trend continued until the end of the period.

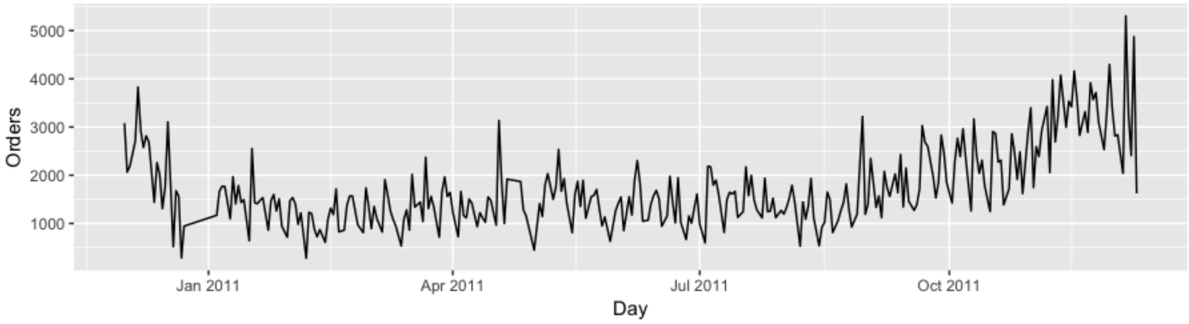


Fig 4. Number of orders per day

Revenue over time was almost flat during the period under consideration. However, after October 2011, the revenue per day witnessed a steep rise that continued until the end of the period.

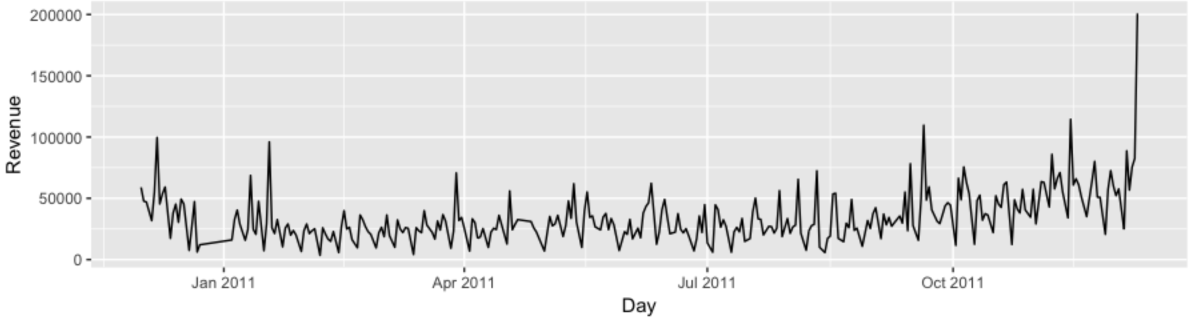


Fig 5. Revenue per day

5. Customer Segmentation

Customer segmentation is a way of clustering the customers in different groups based on their buying behaviour, demographics, lines of business, location etc. enabling the organization to share relevant communication to each customer segment. Segmentation when coupled with demographic data, also helps organizations define user personas that can be used to explore new geographies, businesses or products to introduce in the market.

RFM[5] segmentation is a widely used method that is based on purchase pattern of the customers. RFM stands for Recency, Frequency and Monetary.

Recency- How much time has passed since the customer made the last purchase
Frequency- What is the frequency of purchase by the customer in a given period
Monetary- How much money does a customer spend on average per purchase

Based on the above metrics, the data is aggregated for every customer. This aggregated data is used for segmentation. Also note that the observations with missing CustomerID values are not considered for segmentation.

For calculation of **Recency**, it has been assumed that the segmentation was done in early 2012 and an arbitrary date of **2nd Jan 2012** is assumed to calculate recency. The recency metric will measure how many days have passed since the last order was made until 2nd Jan.

Deciding optimum number of clusters is one of the major questions that rise during customer segmentation. For the purpose of deciding optimum number of clusters, following 3 methods are used.

5.1. Silhouette Distance Method

Silhouette distance method suggests the optimal number of clusters to be 3.

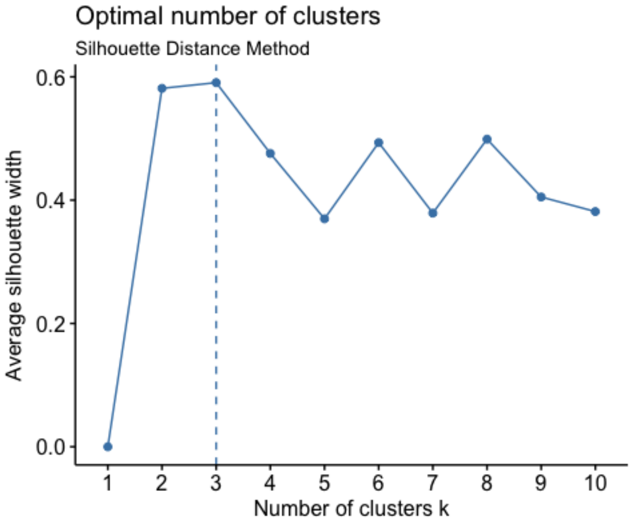


Fig 6. Silhouette Distance Method

[5] https://www.optimove.com/learning-center/rfm-segmentation

5.2. Elbow Curve Method

Elbow curve method measures twss- Total within sum of squares of the clusters for different values of K. From the below graph, it can be seen that, this method also suggests 3 as the optimal number of clusters. There is a steep flat curve at K=6, however, 6 clusters would be too many.

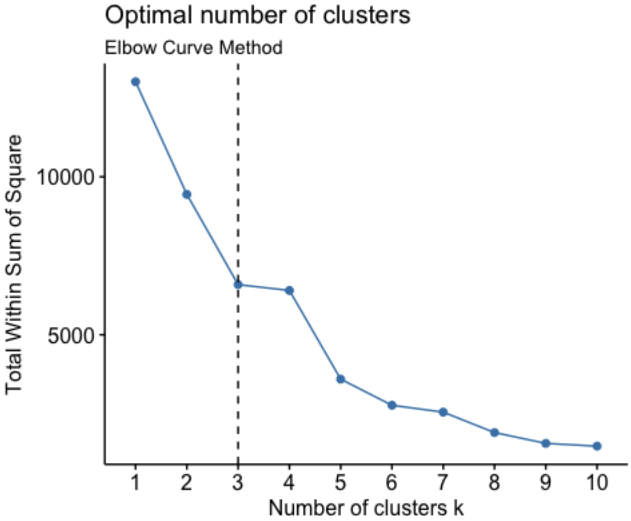


Fig 7. Elbow Curve Method

5.3. Gap Statistic Method[6]

Gap statistic is another technique proposed by Rob Tibshirani, Tervor Hastie and Guenther Walther. This method standardises the graph of $\log(W_k)$ by comparing it with its expectation under appropriate null reference of the data. W_k, in this case stands for within-cluster dispersion. Gap statistic method suggests 2 as optimal number of clusters.

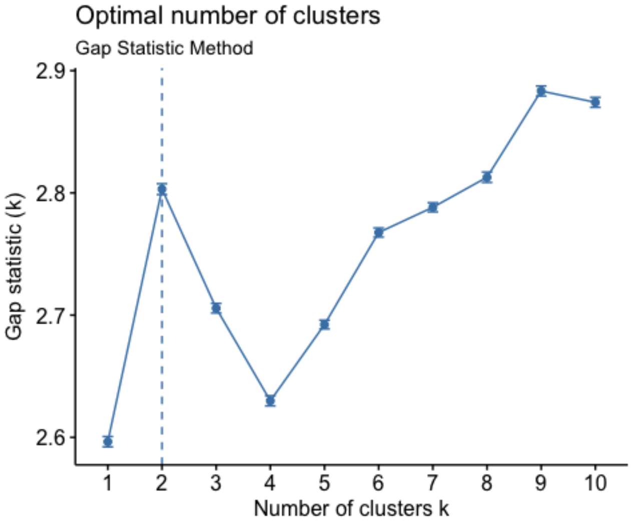


Fig 8. Gap Statistic Method

[6] https://statweb.stanford.edu/~gwalther/gap

5.4. Final Segmentation

Based on above analysis, we finally create three customer segments using K-Means algorithm and analyse their features. In order to visualize the cluster, the dimensions are reduced to 2, they together explain about 74% of the variance.

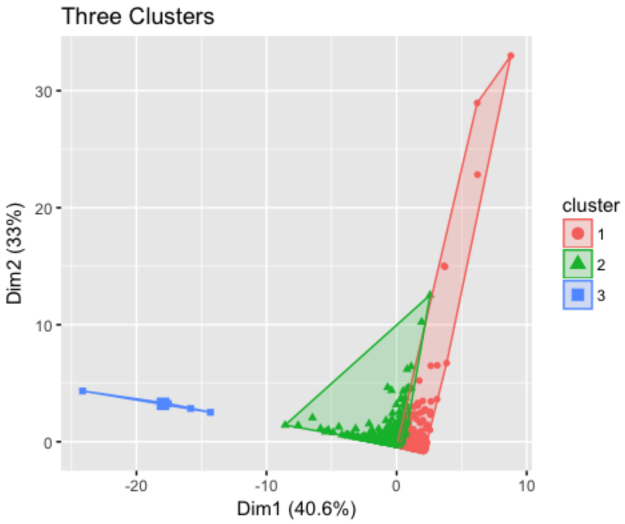


Fig 9. Final Clusters

Now, we analyse the properties of the three clusters that have been defined.

Table 4 Segment Averages

segment	customers	freq	rec	money
1	1073	27.43	270.77	50.35709
2	3260	105.91	64.79	29.48278
3	4	5807.75	25.11	12.6613

Following strategies can be used to tackle the segments that have been identified.

5.4.1. **Segment 1:** These are the wholesalers that purchased products around **27** times on average during the given period and have average order size of about **$50,** highest among the group. These customers should be motivated to buy products more frequently by running campaigns, discount coupons, loyalty program options etc. Increasing the purchase frequency of these customers can lead to considerable rise in the revenue for the retail store.

5.4.2. **Segment 2:** These are the customers that have second highest average basket value of about **$30** and second highest purchase frequency. Efforts should be taken to increase the basket size of orders from these customers. For instance, introducing product combos, bulk purchase discounts, product recommendations can lead to increase in average basket value. This segment generates the most revenue for the retail store.

5.4.3. **Segment 3:** These are four customers that have the highest purchase frequency of **5807**, however, their average basket value is the least of about **$12.** Efforts should be taken to increase the basket value of these customers.

6. Market Basket Analysis[7]

Market basket analysis is a statistical technique that is used to find association among the products that are sold to come up with baskets of similar products. Following are the major concepts on which market basket analysis is based on.

6.1. Support

Support of a product or combination of products is defined as the ratio of number of transactions where the product or combination of products was purchased to the total number of transactions.

$$Support = \frac{Transactions\ where\ product\ was\ purchased}{Total\ transactions}$$

6.2. Confidence

Confidence of rule A->B, where A and B are two different products is defined as the ratio of transactions where both products A and B were purchased to the number of transactions where only product A without product B was purchased.

$$Confidence = \frac{Support(A,B)}{Support\ B}$$

6.3. Lift

Lift for any rule A->B is the ratio of support of A and B to the product of support of A and support of B.

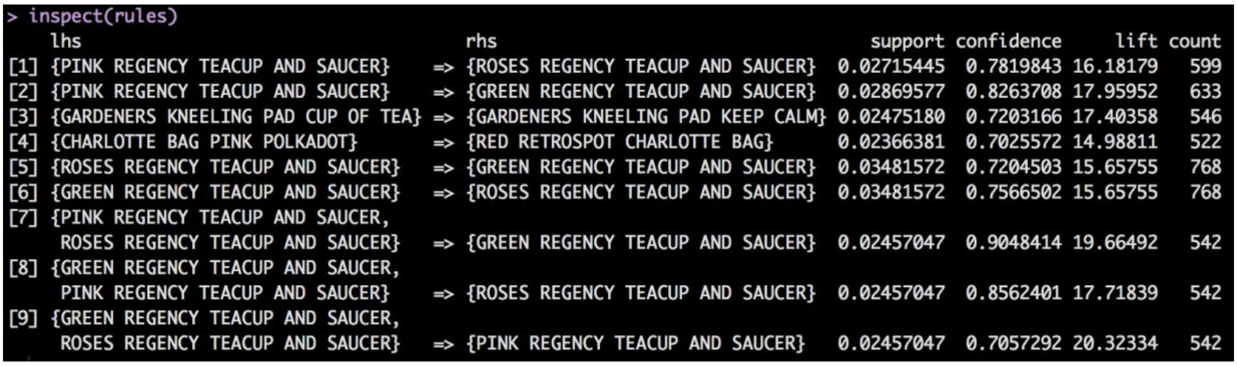
$$Lift = \frac{Support(A,B)}{Support\ (A) * Support(B)}$$

For the purpose of this analysis, we have used apriori method from arules[8] package in R. Deciding minimum support and confidence values is a strategic decision made based on the number of rules the organization wants to create and the total number of distinct products that are sold. For the purpose of this project, we have used minimum support as **0.02** and minimum confidence of **0.7.** Meaning that, for any rule A->B we define, we will be **70%** confident that any customer buying product A will also buy product B.

[7] http://www.albionresearch.com/data_mining/market_basket.php
[8] https://cran.r-project.org/web/packages/arules/index.html

6.4. Final Rules

Based on minimum support of **0.02** and minimum confidence of **0.7,** we have following results. There were **9** rules that were identified by the algorithm.



```
> inspect(rules)
    lhs                                   rhs                                  support confidence     lift count
[1] {PINK REGENCY TEACUP AND SAUCER}    => {ROSES REGENCY TEACUP AND SAUCER}  0.02715445  0.7819843 16.18179   599
[2] {PINK REGENCY TEACUP AND SAUCER}    => {GREEN REGENCY TEACUP AND SAUCER}  0.02869577  0.8263708 17.95952   633
[3] {GARDENERS KNEELING PAD CUP OF TEA} => {GARDENERS KNEELING PAD KEEP CALM} 0.02475180  0.7203166 17.40358   546
[4] {CHARLOTTE BAG PINK POLKADOT}       => {RED RETROSPOT CHARLOTTE BAG}      0.02366381  0.7025572 14.98811   522
[5] {ROSES REGENCY TEACUP AND SAUCER}   => {GREEN REGENCY TEACUP AND SAUCER}  0.03481572  0.7204503 15.65755   768
[6] {GREEN REGENCY TEACUP AND SAUCER}   => {ROSES REGENCY TEACUP AND SAUCER}  0.03481572  0.7566502 15.65755   768
[7] {PINK REGENCY TEACUP AND SAUCER,
     ROSES REGENCY TEACUP AND SAUCER}   => {GREEN REGENCY TEACUP AND SAUCER}  0.02457047  0.9048414 19.66492   542
[8] {GREEN REGENCY TEACUP AND SAUCER,
     PINK REGENCY TEACUP AND SAUCER}    => {ROSES REGENCY TEACUP AND SAUCER}  0.02457047  0.8562401 17.71839   542
[9] {GREEN REGENCY TEACUP AND SAUCER,
     ROSES REGENCY TEACUP AND SAUCER}   => {PINK REGENCY TEACUP AND SAUCER}   0.02457047  0.7057292 20.32334   542
```

Fig 10. Association Rules

Below is the visualization of the rules that have been identified. For the purpose of readability, the complete description of the product is shown in the below graph. In the below graph, size and the colour of the circles represent support and lift respectively.

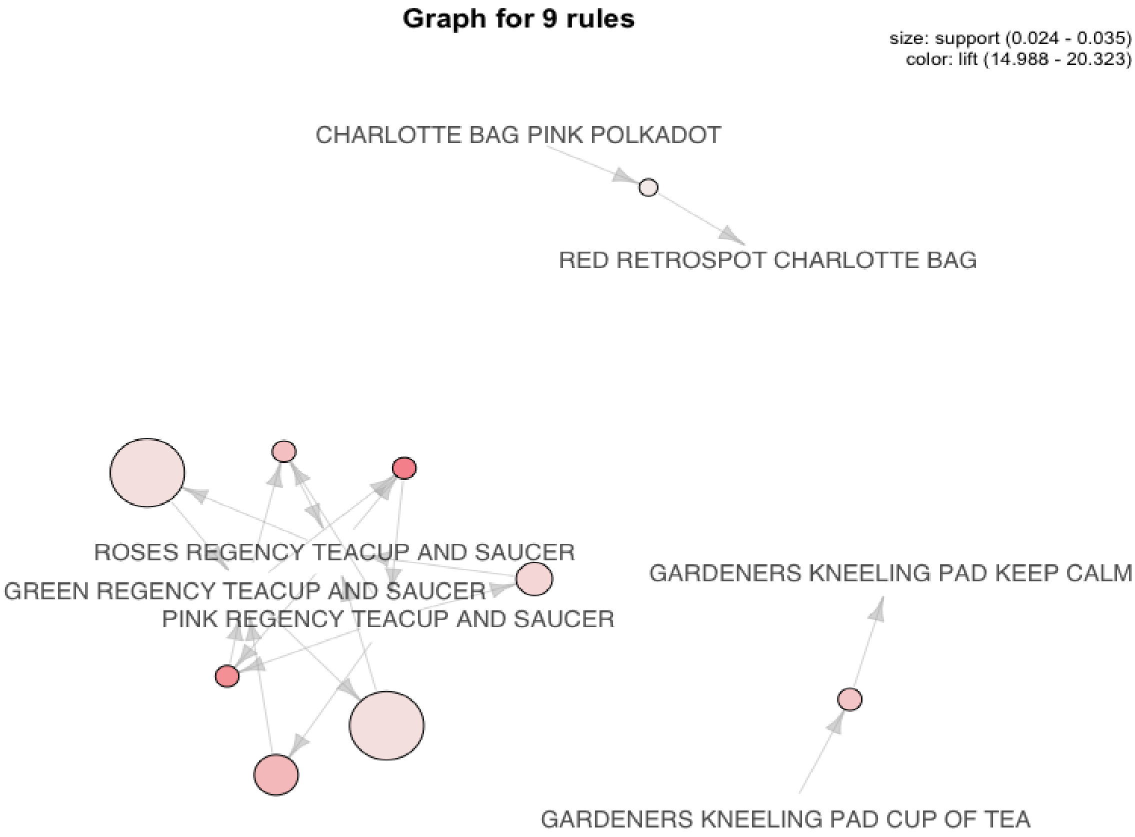

Graph for 9 rules
size: support (0.024 - 0.035)
color: lift (14.988 - 20.323)
CHARLOTTE BAG PINK POLKADOT
RED RETROSPOT CHARLOTTE BAG
ROSES REGENCY TEACUP AND SAUCER
GREEN REGENCY TEACUP AND SAUCER
PINK REGENCY TEACUP AND SAUCER
GARDENERS KNEELING PAD KEEP CALM
GARDENERS KNEELING PAD CUP OF TEA

Fig 11. Visualization of rules

Support and confidence of the 9 rules that have been identified are plotted in the below graph. The intensity of the red colour represents the lift of the rule. Higher the lift darker the colour.

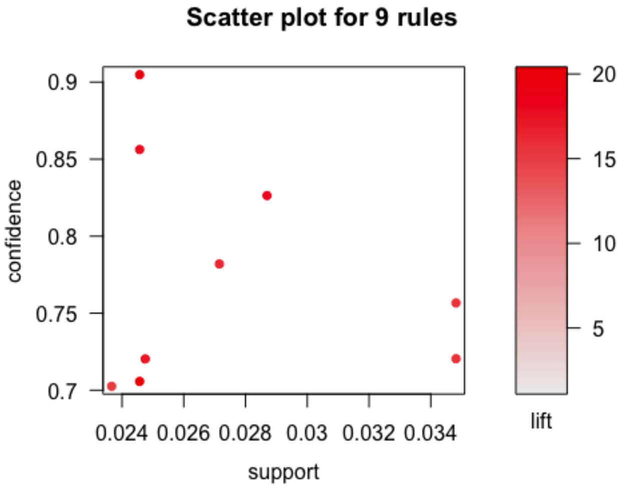


Fig 11. Scatter plot of rules

7. Recommender System

The retail store has customers from 38 different countries. United Kingdom, Germany and France are the major countries that purchase products from the retail store. There are however other countries where lesser number of products are sold annually. Product recommendations for any country can be given based on collaborative filtering[9], which is a branch of information filtering.

For the purpose of this analysis, the data is aggregated at country level and the record of whether a particular product is purchased or not is kept. There are about 4069 products that are sold on the website, we have generated a 38X4069 matrix where the count of each product sold is recorded. Some countries have higher affinity towards certain products than others. In order to measure this affinity, every record in the cell is divided by the total number of products that the country purchased. Thus, we get a matrix of countries and their relative affinity for every product.

Once, the matrix is generated, it is converted to realRatingMatrix[10] so that methods in recommenderlab package can be used. Then, we try to find out user similarity based on cosine distance as the measure.

[9] https://en.wikipedia.org/wiki/Collaborative_filtering
[10] https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Below is the user similarity graph that has been generated. Darker colours represent higher cosine similarity between the pair of users.

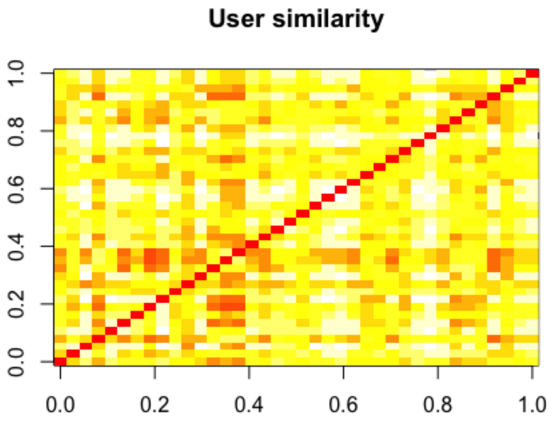


Fig 12. User Similarity

For the purpose of comparison, we try following methods of recommendations-

Random: - Random product recommendations
Popularity: - Popularity based product recommendation
User Based Jaccard Distance: - User based collaborative filtering using Jaccard distance measure.
User Based Cosine Distance: - User based collaborative filtering using cosine distance
User Based Pearson Correlation: - User based collaborative filtering using Pearson's correlation coefficient.

evaluate method in the recommenderLab package lets us compare the performance of all the listed methods in one go.

Below are the results of the comparison.

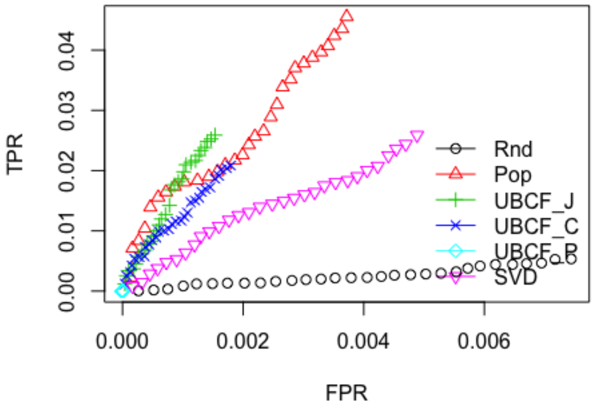


Fig 13. Evaluation, type= TopN

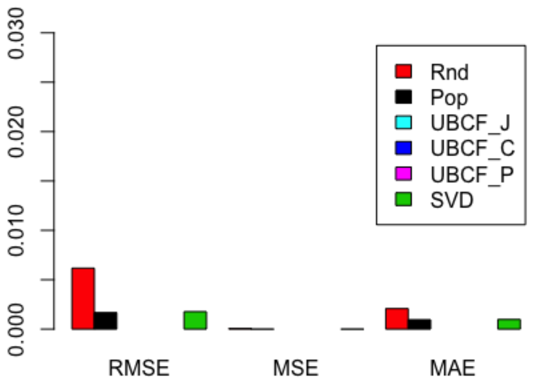


Fig 14. Evaluation, type= ratings

As we can see from above that popularity-based method performs the best. We will go ahead and use this method to make product recommendations.

For the purpose of test, we will recommend products for EIRE country.

Table 5 Product Recommendations

StockCode	**Predicted Rating**
37448	0.001085535
23255	0.001102793
22659	0.001135
23190	0.001310328
20682	0.001419168
22067	0.001721582
48138	0.00216056
22314	0.002281174
20967	0.004246654
POST	0.019689618

We know, POST is not a product, but rest are. Similarly, we can use this method for recommending products for any other country.

8. Tableau Dashboard

Based on all the analysis done so far in project, tableau dashboard is designed for the management to get the report of the business at a glance. The dashboard that has been designed is interactive, the results change in real time as the mouse curser is hover over the geographical map.

Interactive version of the dashboard can be found on Tableau Public at below link-
https://public.tableau.com/profile/swapnil.patil#!/vizhome/RetailViz_0/Dashboard1

Following major metrics are tracked in the dashboard.

Revenue: - Total revenue generated in the selected country for the given time period.
Customers: - Total number of customers who purchased products from the selected country
Products: - Total unique products that were purchased in the selected country
Revenue Per Customer: - Average revenue generated per customer from the given country
Revenue Per Segment: - Average revenue per segment for the given country
Revenue Per Month: - Time series graph of revenue generated per month for the country

Please note that the revenue per segment may not add up to total revenue generated as there are many records with missing CustomerID, who's segments could not be identified. It can be noted that **segment 2** generates the most revenue overall.

Below is the screengrab of default dashboard.

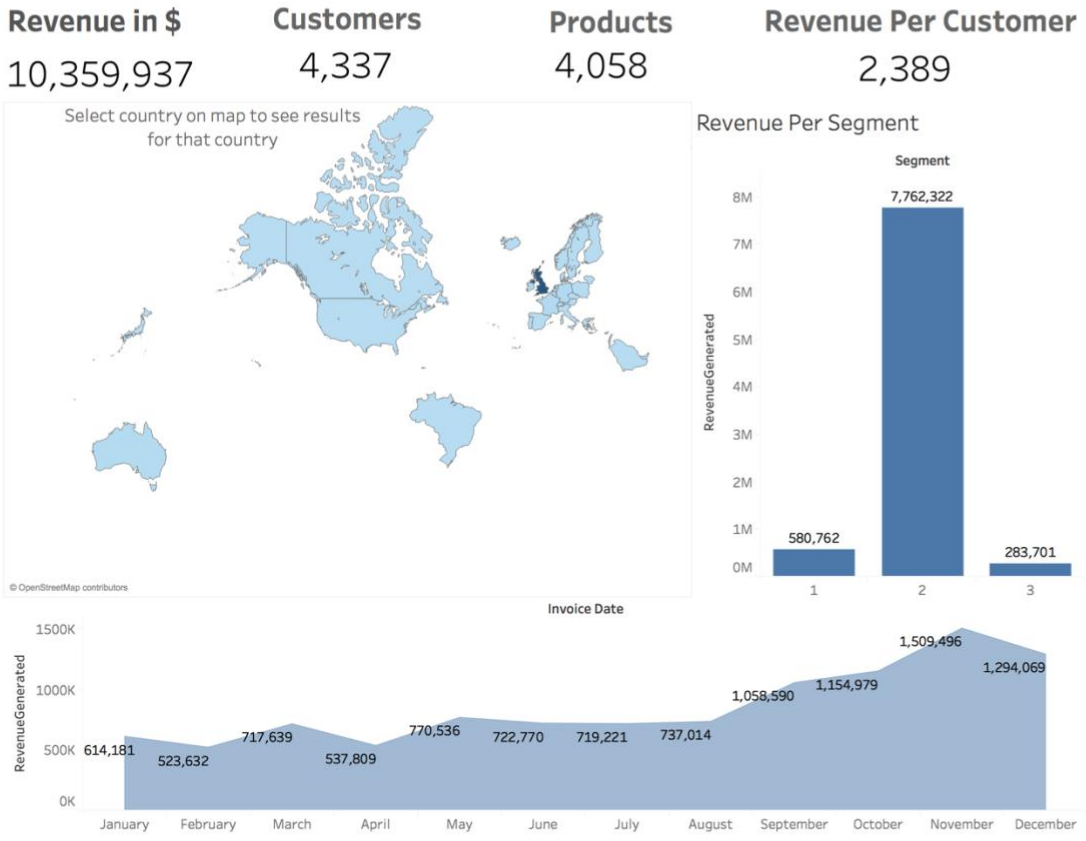


Fig 15. Default dashboard

Below is the screengrab of the dashboard when country United Kingdom was selected. Note that the metrics have been updated with numbers relating the sale in United Kingdom. United Kingdom accounts for almost **90%** of the total revenue of the company.

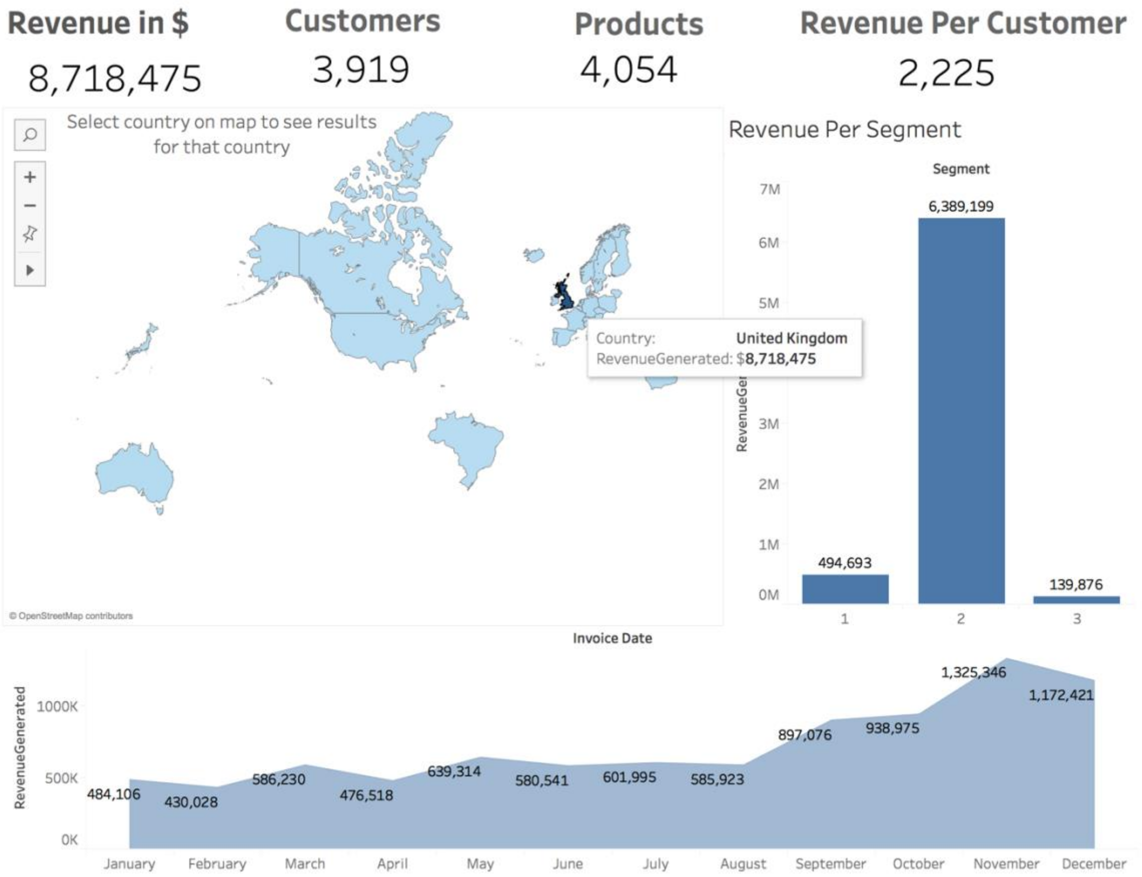


Fig 16. Dashboard when UK is selected

References-

1. http://analytics-magazine.org/corporate-profile-advanced-analytics-kroger/
2. http://archive.ics.uci.edu/ml/datasets/online+retail
3. https://cran.r-project.org/web/packages/recommenderlab/index.html
4. https://www.optimove.com/learning-center/rfm-segmentation
5. https://statweb.stanford.edu/~gwalther/gap
6. http://www.albionresearch.com/data_mining/market_basket.php
7. https://cran.r-project.org/web/packages/arules/index.html
8. https://en.wikipedia.org/wiki/Collaborative_filtering
9. https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Appendix-

```
library(readxl)
library(tidyverse)
library(factoextra)
library(NbClust)
library(sqldf)
library(Matrix)
library(arules)
library(arulesViz)
library(recommenderlab)
setwd("~/Study/MS-Bana/Capstone")

retail<-read_xlsx('Online Retail.xlsx')
head(retail)

### EDA Part ###
str(retail)
colSums(is.na(retail))
dim(retail)
summary(retail)

### Quantity
summary(retail$Quantity)
hist(retail$Quantity)
plot.ecdf(retail$Quantity)

# Anything that has quantity greater than 10k and less than -10k should be removed
retail<-retail%>%
  filter(Quantity<10000,Quantity>-10000)
table(retail$Description)

## Unit Price
summary(retail$UnitPrice)
hist(retail$UnitPrice)
plot.ecdf(retail$UnitPrice)
#View(retail$UnitPrice)

retail[which(retail$UnitPrice==max(retail$UnitPrice)),]
as.data.frame(retail[which(retail$UnitPrice>10000),])

## Removing cancelled orders
retail<-retail[!startsWith(retail$InvoiceNo,'C'),]

### Date
ts<-retail%>%
  group_by(as.Date(InvoiceDate))%>%
  summarise(orders=n())
colnames(ts)<-c('Day','Orders')

ggplot(data = ts,aes(x = Day,y = Orders))+geom_line()

ts2<-retail%>%
  group_by(as.Date(InvoiceDate))%>%
  summarise(revenue=sum(Quantity*UnitPrice))

colnames(ts2)<-c('Day','Revenue')

ggplot(data = ts2,aes(x = Day,y = Revenue))+geom_line()
```

```
## Segmentation ##

## It has been observed that there are missing values in the customerID column.
For the purpose of segmentation
# We will remove those observations. But we will consider those observations for
MarketBasket Analysis.

### Test 556444

retail<-retail[-which(retail$InvoiceNo==556444),]
### End test
aggregated<-retail%>%
  filter(!is.na(CustomerID))%>%
  group_by(CustomerID)%>%

summarise(frequency=n(),latest=max(InvoiceDate),monetory=mean(UnitPrice*Quantity))

aggregated<-as.data.frame(aggregated)
head(aggregated)

## we need to express recency in the number of days since the last purchase has
been made.
#The latest date is 9th Dec 2011.
max(aggregated$latest)
# We can assume that this analysis was done in early 2012 and proceed accordingly.
today<-as.POSIXct("2012-01-02 00:00:00 UTC")
aggregated<-aggregated%>%
  mutate(recency=today-latest)
aggregated$latest<-NULL
aggregated$recency<-as.numeric(aggregated$recency)
head(aggregated)
summary(aggregated)

## there are observations with negative monetory value. These could be because of
some errors.
# we can remove those observations

#aggregated<-aggregated%>%
 # filter(monetory>=0)

## for the purpose of analysis, we need to scale the variables
test<-scale(aggregated[,-1])

## We will use following methods to decide optimum number of clusters

# 1. Silhouette method
fviz_nbclust(test, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette Distance Method")

# 2. Elbow method
fviz_nbclust(test, kmeans, method = "wss") +
  geom_vline(xintercept = 3, linetype = 2)+
  labs(subtitle = "Elbow Curve Method")


# 3. Gap statistic
set.seed(22334455)
fviz_nbclust(test, kmeans, nstart = 25,  method = "gap_stat", nboot =100)+
  labs(subtitle = "Gap Statistic Method")
```

```
## 2 out of the above three method suggest going for 4 clusters. We will create 4
clusters based on data
test<-aggregated
test$frequency<-scale(test$frequency)
test$recency<-scale(test$recency)
test$monetory<-scale(test$monetory)
head(test)

km<-kmeans(test[,-1],centers = 3,iter.max = 30)
test$segment<-km$cluster

aggregated<-sqldf('select aggregated.*,test.segment from aggregated inner join
test
      on aggregated.CustomerID=test.CustomerID')

## Finding out features of the clusters
fviz_cluster(km, geom = "point", data = aggregated[,-c(1,5)]) + ggtitle("Three
Clusters")

as.data.frame(aggregated%>%
  group_by(segment)%>%

summarise(customers=n(),freq=mean(frequency),rec=mean(recency),money=mean(monetory
)))



## Writing back in the original data frame
retail<-sqldf('select retail.*,test.segment from retail left join test
      on retail.CustomerID=test.CustomerID')

write.csv(retail,"VizTableau.csv")




#### Market Basket Analysis
head(retail)

## Any order that has been cancelled has Invoice Number starting from 'C'
# We will not consider those orders
test<-retail%>%group_by(InvoiceNo,StockCode)%>%summarise(Value=1)
test<-test[!startsWith(test$InvoiceNo,'C'),]
head(test)
test<-test%>%spread(StockCode,Value,fill = 0)
test<-as.data.frame(test)
head(test)
str(test)
rowSums(test[,-1])
colSums(is.na(test))

### Association Rules
test<-retail%>%group_by(InvoiceNo,Description)%>%summarise(Value=1)
test<-test[!startsWith(test$InvoiceNo,'C'),]
head(test)
test<-test%>%spread(Description,Value,fill = 0)
test<-as.data.frame(test)
```

```
head(test)
str(test)
rowSums(test[,-1])
colSums(is.na(test))

### Association Rules
Mat<-as.matrix(test[-1,-1])
dim(Mat)
class(Mat[2,3])
#buckets <- eclat (Mat[,-1], parameter = list(supp = 0.0015, minlen = 2))
#inspect(buckets)


### 9 rules 0.02 conf=0.7
s<-as(Mat,"transactions")
rules <- apriori(s, parameter = list(supp = 0.02,conf = 0.7))
plot(rules)
plot(rules, method="graph")
inspect(rules)


###### 3 Recommendations, Collaborative filtering ##
test<-retail%>%group_by(Country,StockCode)%>%summarise(Value=n())
head(test)
test<-test%>%spread(StockCode,Value,fill = 0)
test<-as.data.frame(test)
head(test)
str(test)
rowSumVector<-rowSums(test[,-1])
colSums(is.na(test))

## Now we divide the numbers in the data by rowsums.

for(r in 1:nrow(test))
{
  for(c in 2:ncol(test))
  {
    test[r,c]<-test[r,c]/rowSumVector[r]
  }
}

for(r in 1:nrow(test))
{
  for(c in 1:ncol(test))
  {
    if(test[r,c]==0)
      {
      test[r,c]<-NA
    }
  }
}

head(test)
df<-test
## Converting to Matrix
test<-sapply(data.frame(test),as.numeric)
test[1:5,]
#test<-as.matrix(test)
#colnames(test)<-colnames(df)
```

```
#rownames(test)<-rownames(df)

testRating<- as(test, "realRatingMatrix")
colnames(testRating)<-colnames(df)
rownames(testRating)<-rownames(df)
image(testRating, main = "Raw Ratings")
## Finding out similar users
similarity_users <- similarity(testRating, method =  "cosine", which = "users")
image(as.matrix(similarity_users), main = "User similarity")
set.seed(222)
scheme <- evaluationScheme(testRating, method="split", train = 0.9,
                           k=5, given=9, goodRating=0.00025)

algorithms <- list("Rnd" = list(name="RANDOM", param=NULL),
                   "Pop" = list(name="POPULAR"),
                   "UBCF_J" = list(name="UBCF",param = list(method = "jaccard")),
                   "UBCF_C" = list(name="UBCF",param = list(method = "cosine")),
                   "UBCF_P" = list(name="UBCF",param = list(method = "pearson")),
                   #"IBCF_J" = list(name="IBCF", pparam = list(method = 
"jaccard"))
                   #"IBCF_C" = list(name="IBCF", pparam = list(method = 
"cosine")),
                  # "IBCF_P" = list(name="IBCF", param = list(method = "pearson"))
                   "SVD" = list(name="SVD")
                   )

### Comparing the top recommendations ####
results1 <- evaluate(scheme, algorithms, type = "topNList",n=1:30)
#results1
plot(results1,legend="bottomright")
#plot(results1, "prec/rec", legend="bottomright")

results2 <- evaluate(scheme, algorithms, type = "ratings")
plot(results2,ylim = c(0,0.03),xlim = c(0,10),col=factor(names(results2)))


## we are going with popularity
recoModelPop<-Recommender(testRating,method = "POPULAR")
popRecom <- predict(recoModelPop, testRating[11,],type="topNList")
popRecomMat<-as(popRecom,"matrix")
popRecoMat1<-t(popRecomMat)
colnames(popRecoMat1)<-df[,1][11]

popRecodf<-as.data.frame(popRecoMat1)
head(popRecodf)
View(popRecodf)
df[,1]
```