Detecting Customer Purchasing Patterns using Association Rule Mining

MSc Research Project
Data Analytics

Rhutujit Uday Paradkar
Student ID: 20187416

School of Computing
National College of Ireland

Supervisor: Dr. Paul Stynes
Dr. Anu Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rhutujit Uday Paradkar
Student ID:	20187416
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Paul Stynes
Submission Due Date:	19/09/2022
Project Title:	Detecting Customer Purchasing Patterns using Association Rule Mining
Word Count:	957
Page Count:	12

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	19th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Detecting Customer Purchasing Patterns using Association Rule Mining

Rhutujit Uday Paradkar
20187416

1 Introduction

The main aim of creation of this document is to give all the steps that are essential for reproducing the research in the future. The document acts as a technical manual for reproducing all the research related to the association rule mining , Kmeans clustering and the principal component analysis. This manual consists of the screenshots of the essential parts of the code so that the understanding of the code becomes easier.

2 Prequisites: Hardware and Software

The software and the hardware requirements that are needed for completing the research are mentioned below:

2.1 Hardware Requirements

1. **Operating System** : Windows 10 and above
2. **Processor** : Intel Core i7 7th Generation and above
3. **RAM** : 8 GigaBytes
4. **GPU** : Geforce GTX 1050 Ti
5. **GPU Memory** : 4 Gigabytes
6. **Hard Disk** : 1 Terra bytes
7. **Solid state drive** : 256 Gigabytes

3 Software Requirements

The Integrated Development environment or the IDE that was used for the research were:

1. Jupyter notebook for running the python code
2. R studio for running the R code

Anaconda software was used as a main controlling software for installing and controlling all the IDE's mentioned above.

3.1 Installing Anaconda Navigator

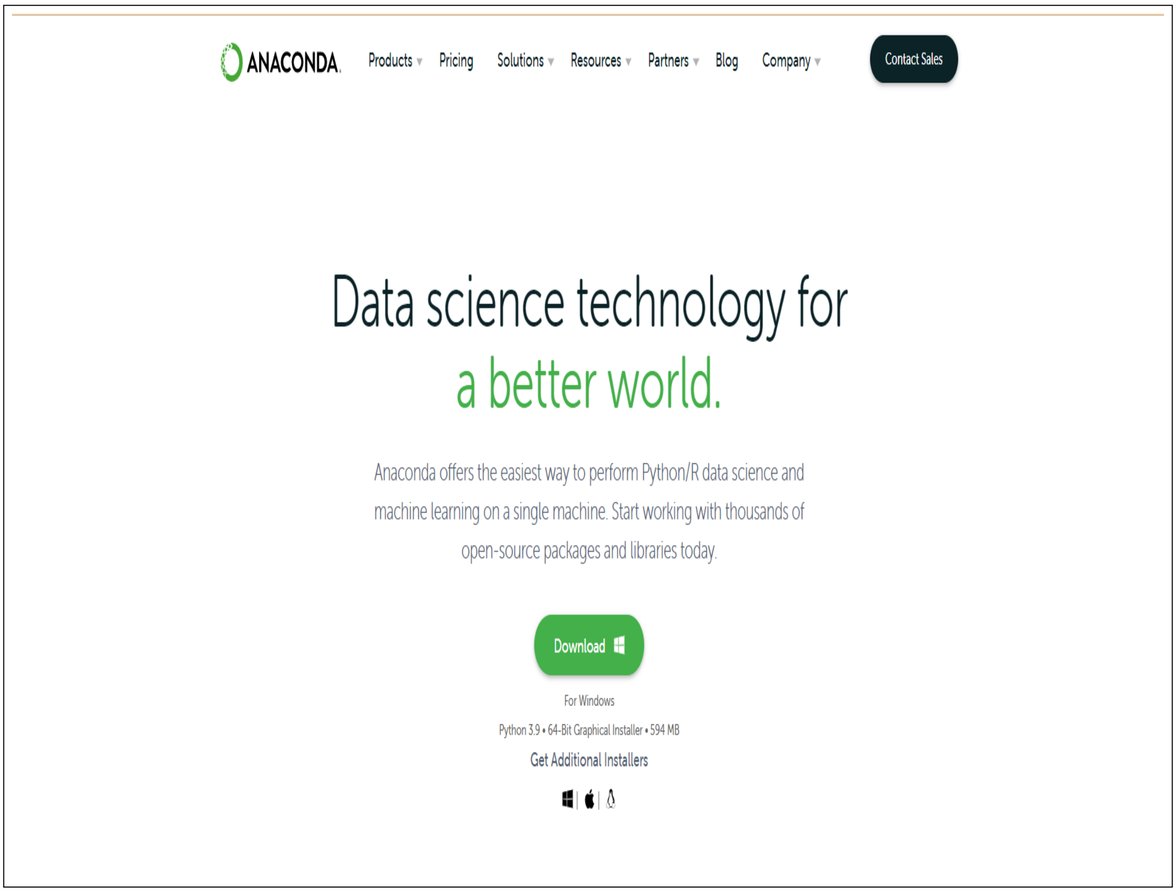

Figure 1: Website to download windows version of anaconda

The site www.anaconda.com has the links to install the anaconda for the following operating systems:

1. iOS
2. Windows
3. Linux

As the system that we are using is Windows we are using the windows version of anaconda.

3.2 Installing and Launching the Jupyter and the R notebooks from Anaconda

In this section, we will take a look at the process to install and run the Jupyter and the R notebooks. In some cases the Anaconda navigator comes preinstalled with Jupyter notebook and you just need to run it.

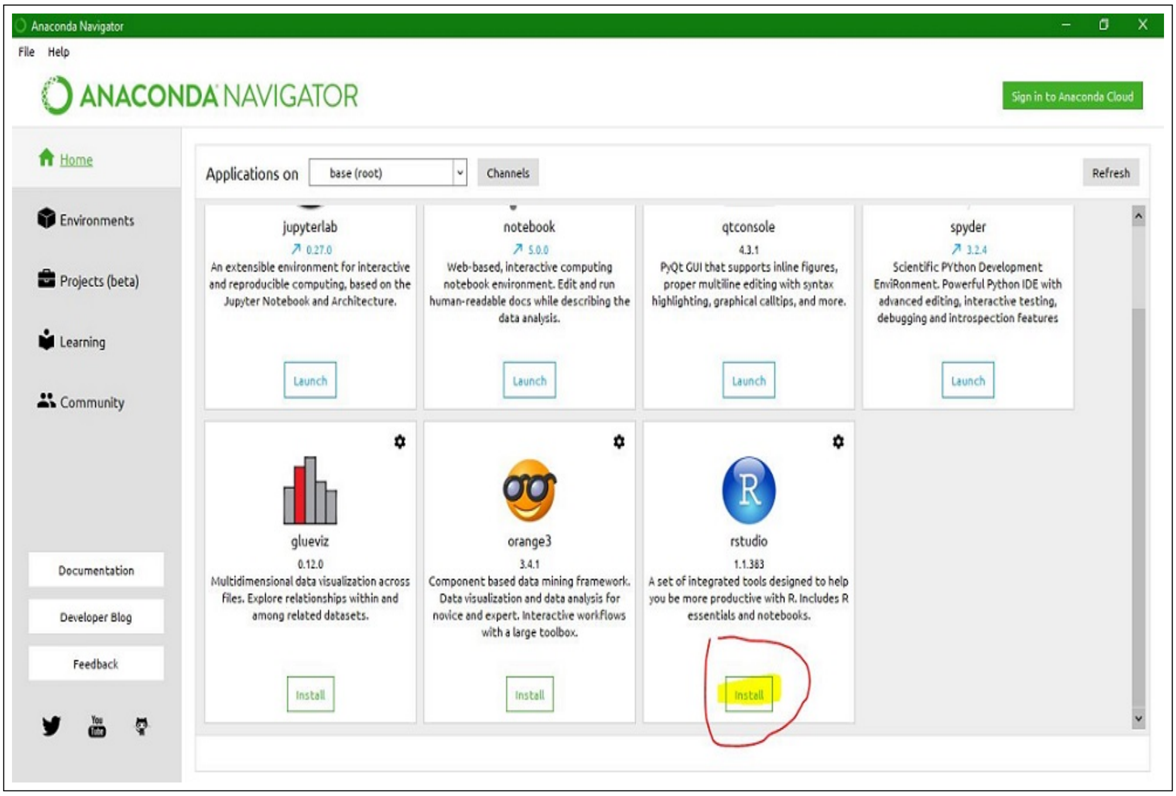

Figure 2: Installation of R studio in the anaconda distribution

Once you click on the install button, it will take some time to install the R distribution and the progress will be shown in the bottom right of the main installation window. This process is same for the Jupyter notebook as well.

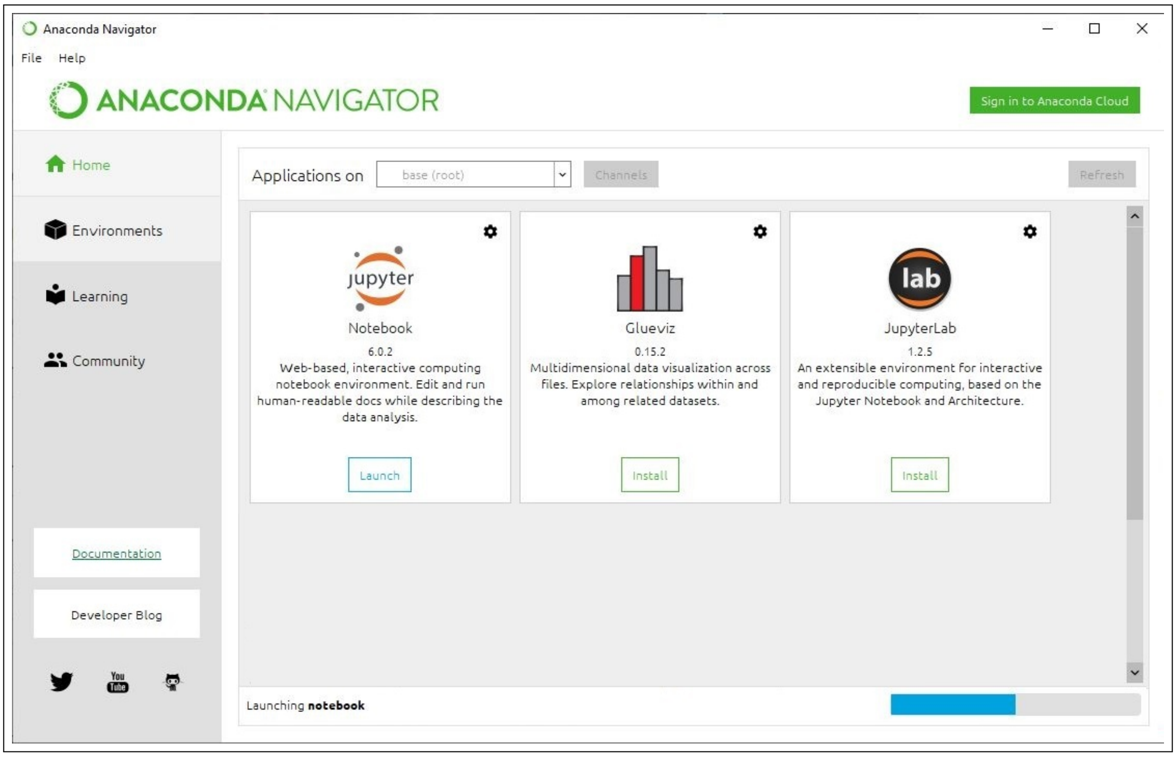

Figure 3: Installation of Jupyter in the anaconda distribution

4 Data Pre-processing

The libraries that are needed for the projects in this research paper are given below. To first implement various preprocessing techniques, it is first necessary to import the files into the code.

4.1 Experiment 1 - Market basket Analysis(Importing the libraries)

The file format of this dataset is csv. The pandas library is used for the preprocessing and the read_csv function is used to read the csv file and import the data into the jupyter notebook.

```
In [1]: !pip install apyori
        import requests
        import matplotlib.pyplot as plt

        Requirement already satisfied: apyori in c:\users\rutujit\anaconda3\lib\site-packages (1.1.2)

In [2]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

Data Preprocessing

```
In [3]: mba_dataset = pd.read_csv('Market_Basket_Analysis.csv',header=None)
```

Figure 4: Code to read the csv and import the data

For applying the apriori algorithm to the dataset, the data needs to bee in the form of the list. That is why some preprocessing is needed to convert the data to a list . The figure 5 shows the code to convert the data to a list.

```
In [2]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

Data Preprocessing

```
In [3]: mba_dataset = pd.read_csv('Market_Basket_Analysis.csv',header=None)
        transactions = []
        for i in range(0,mba_dataset.shape[0]):
            transactions.append([str(mba_dataset.values[i,j])for j in range(0,mba_dataset.shape[1])])

In [4]: print(transactions)

        [['shrimp', 'almonds', 'avocado', 'vegetables mix', 'green grapes', 'whole weat flour', 'yams', 'cottage cheese', 'energy dri
        nk', 'tomato juice', 'low fat yogurt', 'green tea', 'honey', 'salad', 'mineral water', 'salmon', 'antioxydant juice', 'frozen
        smoothie', 'spinach', 'olive oil'], ['burgers', 'meatballs', 'eggs', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan',
        'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['chutney', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan',
        'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['turkey', 'avocado', 'nan', 'nan', 'na
        n', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['mineral wate
        r', 'milk', 'energy bar', 'whole wheat rice', 'green tea', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'na
        n', 'nan', 'nan', 'nan', 'nan', 'nan'], ['low fat yogurt', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'na
        n', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['whole wheat pasta', 'french fries', 'nan', 'nan', 'na
        n', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['soup', 'light
        cream', 'shallot', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan',
        'nan', 'nan'], ['frozen vegetables', 'spaghetti', 'green tea', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan',
        'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['french fries', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'n
        an', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['eggs', 'pet food', 'nan', 'nan', 'nan',
        'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['cookies', 'nan',
        'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'na
        n'], ['turkey', 'burgers', 'mineral water', 'eggs', 'cooking oil', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'n
        an', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['spaghetti', 'champagne', 'cookies', 'nan', 'nan', 'nan', 'nan', 'nan', 'na
        n', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['mineral water', 'salmon', 'nan', 'nan',
```

Figure 5: Converting the data to a list

4.2 Experiment 2 - Importing the dataset and the libraries

```r
#Load the data files into the R environment
BigTempRb <- read.csv("20181129_PMRB_0001.txt")
view(BigTempRb)
Products <- read.csv("tcg_products_20181219.csv")

##############################################Data Pre-processing####################################
#Converting the JSON data into the structured format.
json <- BigTempRb$RequestBasketJsonString[1:300000] #Only taking 300,000 records due to system limitations
nested_dataframe <- jsonlite::stream_in(textConnection(gsub("\\n", "", json)))
nested_dataframe <- cbind(nested_dataframe, dt=BigTempRb$ResponseFinancialTimestamp[1:300000], basket_items=BigTempRb$RequestNumbe
df <- nested_dataframe %>% unnest(items)
df = df[-c(2)]    # Remove the unwanted column Curr
names(Products)[1] <- "b" # Rename Product table EAN to b
df <- inner_join(df, Products, by.="b")
df1 <- df

#Removing the items having negative quantity. Also, the baskets with single items are ignored.
df <- filter(df, b != '5000128785617')
```

Figure 6: Importing data for the Glantus dataset

1. In this experiment the data is imported by using the inbuilt read_csv function of R
2. The nested json is converted to simplified version by using the jsonlite library of the R

```r
library(arules)
library(arulesViz)
library(cluster)
library(cluster.datasets)
library(clValid)
library(clustree)
library(corrplot)
library(cowplot)
library(dendextend)
library(dplyr)
library(factoextra)
library(FactoMineR)
library(ggfortify)
library(GGally)
library(ggiraphExtra)
library(jsonlite)
library(knitr)
library(kableExtra)
library(lubridate)
library(magrittr)
library(NbClust)
library(RColorBrewer)
library(tidyverse)
```

Figure 7: Importing the necessary libraries in python for the instacart dataset

The figure 8 shows the necessary libraries that are need to perform the research on the glantus dataset. For installing the necessary libraries please follow the following command

1. install.packages(package name)

4.3 Experiment 3 - Importing the data and installing required packages

The datasets for the 3rd experiment are from the instacart dataset which is taken from kaggle. The dataset consists of multiple files combined together in a zip. Therefore, for importing the dataset the zipfile package from python is used which unzips the files so that we can get the individual csv files.

```
In [2]:
import pandas as pd
import numpy as np

from sklearn.cluster import KMeans

#For dimensionality reduction.
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn import decomposition

%matplotlib inline
import matplotlib.pyplot as plt
from pylab import rcParams
import seaborn as sb
rcParams['figure.figsize'] = 12, 4
sb.set_style('whitegrid')

np.random.seed(42)
```

Figure 8: Importing the necessary libraries in the R studio

```python
import zipfile # Unzips the files
from subprocess import check_output

#Prior Dataset
with zipfile.ZipFile("instacart-market-basket-analysis/"+prior+".zip","r") as z:
    z.extractall(".")
prior = pd.read_csv("order_products__prior.csv")

#Order_Train Dataset.
with zipfile.ZipFile("instacart-market-basket-analysis/"+order_train+".zip","r") as z:
    z.extractall(".")
order_train = pd.read_csv("order_products__train.csv")

#Orders Dataset.
with zipfile.ZipFile("instacart-market-basket-analysis/"+orders+".zip","r") as z:
    z.extractall(".")
orders = pd.read_csv("orders.csv")

#Products
with zipfile.ZipFile("instacart-market-basket-analysis/"+products+".zip","r") as z:
    z.extractall(".")
products = pd.read_csv("products.csv")

#Aisles
with zipfile.ZipFile("instacart-market-basket-analysis/"+aisles+".zip","r") as z:
    z.extractall(".")
aisles = pd.read_csv("aisles.csv")

#Departments
with zipfile.ZipFile("instacart-market-basket-analysis/"+departments+".zip","r") as z:
    z.extractall(".")
departments = pd.read_csv("departments.csv")
```

Figure 9: Code to unzip the files and get the individual csv files

The code in the figure 9 unzips all the files at the given path by passing the '.' parameter which means all files at the givenpath. The 'r' parameter of the ZipFile reads the raw string as it is.

```
In [5]: combined_df_list = [products,orders, departments, aisles, prior, order_train]

In [6]: #Check the size of the datasets.
        for i in combined_df_list:
            print (i.shape)

        del combined_df_list

(49688, 4)
(3421083, 7)
(21, 2)
(134, 2)
(32434489, 4)
(1384617, 4)
```

Figure 10: Code to see the features of the dataset

The code in the figure 10 is used to get an idea about the number of records in each dataset.

4.4 Exploratory data analysis

```
In [21]: plt.figure(figsize=(18,8))
         color = plt.cm.plasma(np.linspace(0, 1, 40))
         dataset[0].value_counts().head(40).plot.bar(color = color)
         plt.title('Most Popular Items', fontsize = 20)
         plt.xticks(rotation = 90 )
         plt.grid()
         plt.show()
```

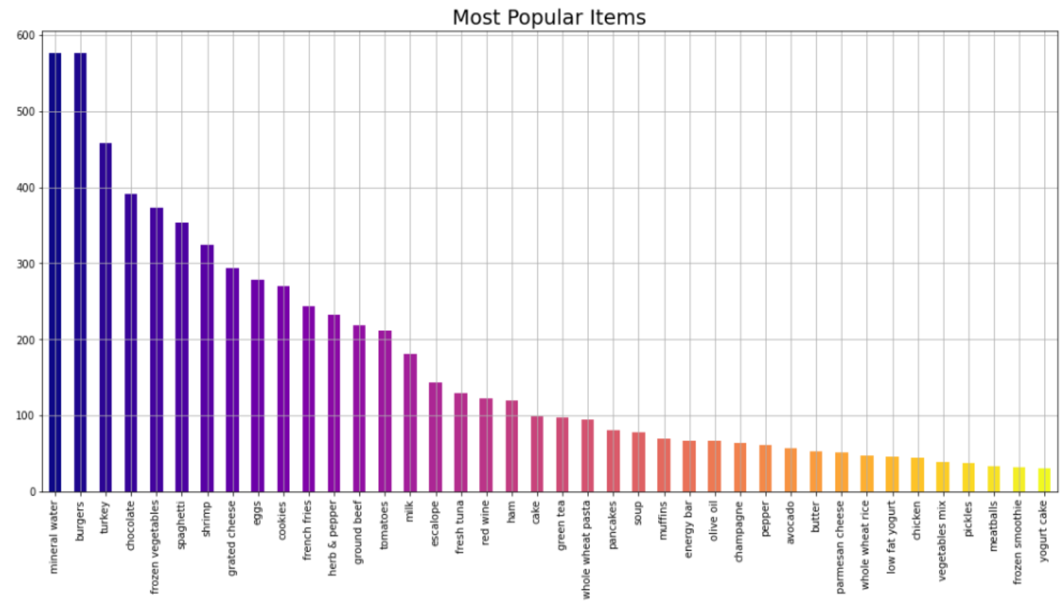

Figure 11: Popular products in the dataset

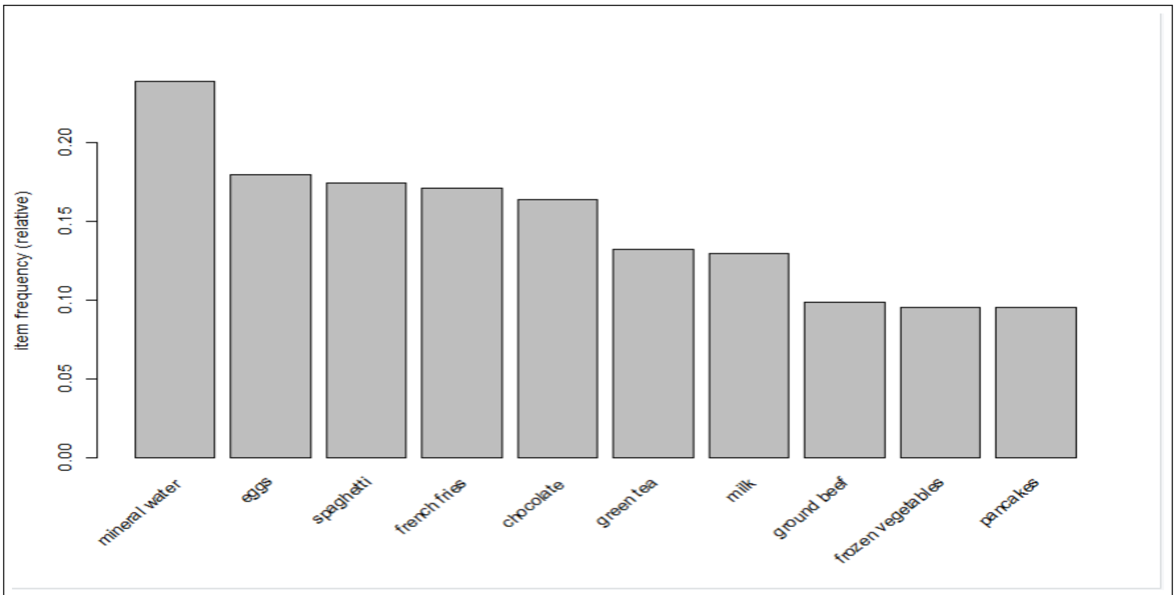

Figure 12: Popular products in the dataset

The figures 12and 11 show the most popular items in the datasets.

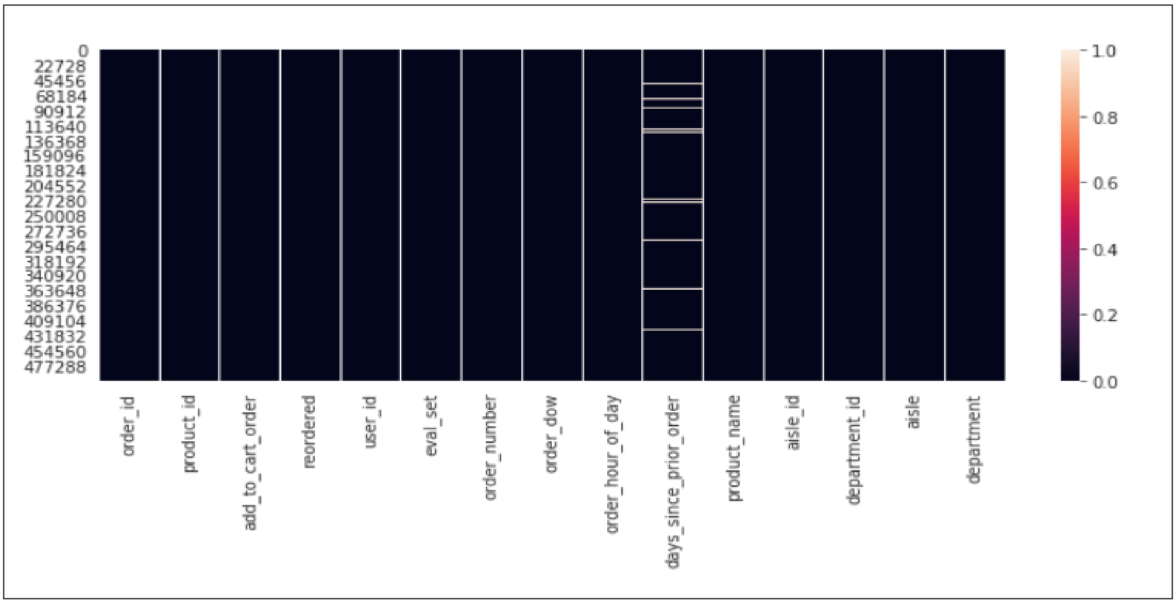

Figure 13: Heatmap for the instacart data

The seaborn library is used to generate the heatmap shown in the fig.13.

```
pd.DataFrame(combined_df.groupby('user_id')['product_id'].count()).sort_values('product_id', ascending=False).head(2)

#User_id = 142131
```

	product_id
user_id	
142131	176
169550	161

Figure 14: Most popular user

The groupby clause is used here to get the user id of the customer who spends the most in the store.

```
In [21]: pd.DataFrame(combined_df['aisle'].value_counts()).head(5)
```

Out[21]:

	aisle
fresh fruits	56326
fresh vegetables	52073
packaged vegetables fruits	27347
yogurt	22485
packaged cheese	14960

Figure 15: The aisle with the most products

The above code is used to get the aisle with the most number of products. The pandas library was used here.

```
In [20]: pd.DataFrame(combined_df['product_name'].value_counts()).head(5)
Out[20]:
```

	product_name
Banana	7365
Bag of Organic Bananas	5920
Organic Strawberries	4023
Organic Baby Spinach	3797
Organic Hass Avocado	3317

Figure 16: Most selling products

The code in the figure 16 is used to get the most frequent items in the dataset.

5 Implementation

The implementation section of the manual describes the different modelling techniques that are performed on the datasets in the research paper. The screenshots in the section show the code that was used to implement the modelling techniques.

```
In [5]: from apyori import apriori
        rules = apriori(transactions= transactions,  min_support = .003, min_confidence = 0.25, min_lift = 2, min_length = 2, max_length
```

Visualising the results

Displaying the first results coming directly from the output of the apriori function

```
In [6]: results = list(rules)

In [7]: results

Out[7]: [RelationRecord(items=frozenset({'burgers', 'almonds'}), support=0.005199306759098787, ordered_statistics=[OrderedStatistic(ite
        ms_base=frozenset({'almonds'}), items_add=frozenset({'burgers'}), confidence=0.25490196078431376, lift=2.923577382023146)]),
         RelationRecord(items=frozenset({'bacon', 'spaghetti'}), support=0.003199573390214638, ordered_statistics=[OrderedStatistic(ite
        ms_base=frozenset({'bacon'}), items_add=frozenset({'spaghetti'}), confidence=0.36923076923076925, lift=2.1206738131699847)]),
         RelationRecord(items=frozenset({'milk', 'black tea'}), support=0.004266097853619517, ordered_statistics=[OrderedStatistic(item
        s_base=frozenset({'black tea'}), items_add=frozenset({'milk'}), confidence=0.29906542056074764, lift=2.3079112341833006)]),
         RelationRecord(items=frozenset({'spaghetti', 'blueberries'}), support=0.003466204506058577, ordered_statistics=[OrderedStatis
        tic(items_base=frozenset({'blueberries'}), items_add=frozenset({'spaghetti'}), confidence=0.37681159420289856, lift=2.164214217
        546663)]),
         RelationRecord(items=frozenset({'body spray', 'french fries'}), support=0.004266097853619517, ordered_statistics=[OrderedStati
        stic(items_base=frozenset({'body spray'}), items_add=frozenset({'french fries'}), confidence=0.37209302325581395, lift=2.177121
        50346479)]),
         RelationRecord(items=frozenset({'cereals', 'milk'}), support=0.007065724570057326, ordered_statistics=[OrderedStatistic(items_
        base=frozenset({'cereals'}), items_add=frozenset({'milk'}), confidence=0.2746113989637306, lift=2.119197637476279)]),
         RelationRecord(items=frozenset({'light cream', 'chicken'}), support=0.004532728969470737, ordered_statistics=[OrderedStatistic
        (items_base=frozenset({'light cream'}), items_add=frozenset({'chicken'}), confidence=0.29059829059829057, lift=4.8439506172839
        5)]),
         RelationRecord(items=frozenset({'chocolate', 'tomato sauce'}), support=0.005065991201173177, ordered_statistics=[OrderedStatis
        tic(items_base=frozenset({'tomato sauce'}), items_add=frozenset({'chocolate'}), confidence=0.3584905660377358, lift=2.187988393
        6932925)]),
         RelationRecord(items=frozenset({'eggs', 'cider'}), support=0.004266097853619517, ordered_statistics=[OrderedStatistic(items_ba
        se=frozenset({'cider'}), items_add=frozenset({'eggs'}), confidence=0.4050632911392405, lift=2.2539909101153137)]),
         RelationRecord(items=frozenset({'milk', 'cider'}), support=0.003332888948140248, ordered_statistics=[OrderedStatistic(items_ba
        se=frozenset({'cider'}), items_add=frozenset({'milk'}), confidence=0.31645569620253167, lift=2.4421133510444344)]),
         RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), support=0.005732568990801226, ordered_statistics=[Ordere
```

Figure 17: Applying apriori on the dataset

The code in the figure 17 is used to generate the apriori rules and the package used for that is 'apyori'.

```
writing  ... [1328 set(s)] done [0.01s].
Creating S4 object  ... done [0.00s].
> eclat_rules <- ruleInduction(rules, marketdata, confidence = 0.2,lift=2)
> inspect(sort(eclat_rules, by = 'lift')[1:10])
     lhs                                       rhs            support     confidence lift     itemset
[1]  {mineral water, whole wheat pasta}       => {olive oil}    0.003866151 0.4027778 6.115863 296
[2]  {frozen vegetables, milk, mineral water} => {soup}         0.003066258 0.2771084 5.484407 615
[3]  {fromage blanc}                          => {honey}        0.003332889 0.2450980 5.164271  43
[4]  {spaghetti, tomato sauce}                => {ground beef}  0.003066258 0.4893617 4.980600  58
[5]  {light cream}                            => {chicken}      0.004532729 0.2905983 4.843951  89
[6]  {pasta}                                  => {escalope}     0.005865885 0.3728814 4.700812  25
[7]  {french fries, herb & pepper}            => {ground beef}  0.003199573 0.4615385 4.697422 540
[8]  {cereals, spaghetti}                     => {ground beef}  0.003066258 0.4600000 4.681764 247
[9]  {frozen vegetables, mineral water, soup} => {milk}         0.003066258 0.6052632 4.670863 615
[10] {french fries, ground beef}              => {herb & pepper} 0.003199573 0.2307692 4.665768 540
>
```

Figure 18: Applying eclat on the mba dataset

The 'arules' package is used for applying the eclat algorithm on the mba dataset. The inspect function is used to sort the generated rules and then display it. The rules are sorted by the decreasing value of the lift.

5.1 Principal component analysis on the instacart dataset

```
In [28]:
ks = range(1,10)
inertias = []
for k in ks:
    model = KMeans(n_clusters=k)
    model.fit(X)                    #
    inertias.append(model.inertia_) #

plt.plot(ks, inertias, '-o', color='black')
plt.xlabel('number of clusters, k')
plt.ylabel('inertia')
plt.xticks(ks)
plt.show()
```

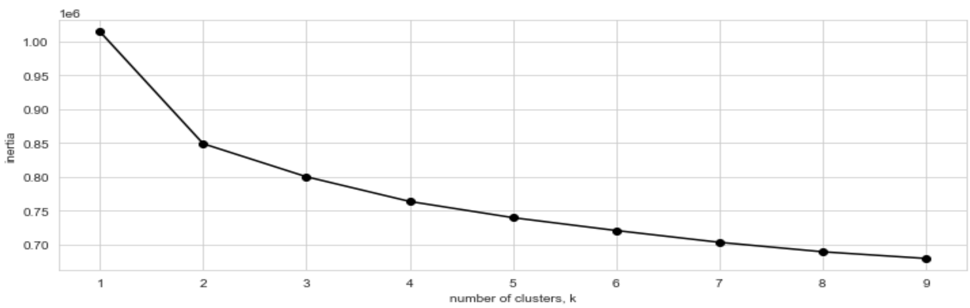

Figure 19: Elbow method for the PCA

The figure shows the line plot and the code that is used to find the optimal number of clusters for the data.

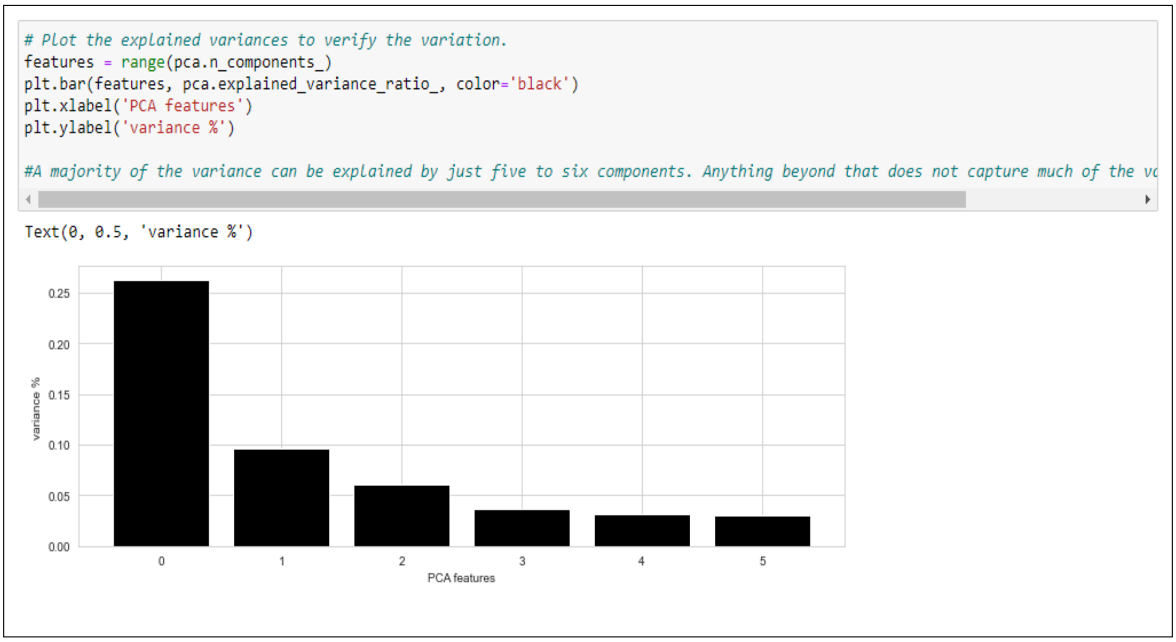

Figure 20: Bar graph variance

This code is used to plot a bar graph that shows the variance that is explained by the number of selected features by performing the principal component analysis.

6 Conclusion

In the conclusion, the manual can be used to replicate the research that is done in the research. The steps and the code are explained in the manual by actual screenshots from the code.