MARKET BASKET INSIGHTS

MEMBER: J FARHATH NASEEM (922121106014)

PHASE 4 SUBMISSION DOCUMENT: DEVELOPMENT PART 2

PROJECT: Market basket insights

Phase 4: Development Part 2

In this part I will continue building my project.

Continue building the market basket insights project

by: • Performing association analysis

• Generating insights.

Dataset Link: **https://www.kaggle.com/datasets/aslanahmedov/market-basket analysis**

About Dataset

Market Basket Analysis

Market basket analysis with Apriori algorithm

Introduction

Association Rule is most used when you are planning to build association in different objects in a set. It works when you are planning to find frequent patterns in a transaction database. It can tell you what items do customers frequently buy together and it allows retailer to identify relationships between the items.

An Example of Association Rules

Assume there are 100 customers, 10 of them bought Computer Mouth, 9 bought Mat for Mouse and 8 bought both of them.

- bought Computer Mouth => bought Mat for Mouse
- support = P(Mouth & Mat) = 8/100 = 0.08
- confidence = support/P(Mat for Mouse) = 0.08/0.09 = 0.89
- lift = confidence/P(Computer Mouth) = 0.89/0.10 = 8.9

  This just simple example. In practice, a rule needs the support of several hundred transactions, before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

Strategy

- Data Import
- Data Understanding and Exploration
- Transformation of the data – so that is ready to be consumed by the association rules algorithm • Running association rules
- Exploring the rules generated
- Filtering the generated rules
- Visualization of Rule

Dataset Description

- File name: Assignment-1_Data
- List name: retaildata
- File format: . xlsx
- Number of Row: 522065
- Number of Attributes: 7
- BillNo: 6-digit number assigned to each transaction. Nominal.
- Itemname: Product name. Nominal.
- Quantity: The quantities of each product per transaction. Numeric.
- Date: The day and time when each transaction was generated. Numeric.

	A	B	C	D	E	F	G
1	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
2	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850	United Kingdom
3	536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850	United Kingdom
4	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850	United Kingdom
5	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850	United Kingdom
6	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850	United Kingdom

- Price: Product price. Numeric.
- CustomerID: 5-digit number assigned to each customer. Nominal.
- Country: Name of the country where each customer resides. Nominal.

Libraries in R

First, we need to load required libraries. Shortly I describe all libraries.

• arules - Provides the infrastructure for representing,
manipulating and analyzing transaction data and patterns (frequent itemsets and association rules).

• arulesViz - Extends package 'arules' with various visualization.
techniques for association rules and item-sets. The package also includes several interactive visualizations for rule exploration.

• tidyverse - The tidyverse is an opinionated collection of R packages designed for data science. • readxl - Read Excel Files in R.

• plyr - Tools for Splitting, Applying and Combining Data.

• ggplot2 - A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

• knitr - Dynamic Report generation in R.

```
library(arules) #Provides the infrastructure for representing
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
1	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
2	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
3	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
4	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
5	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
6	536365	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850	United Kingdom
7	536365	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850	United Kingdom
8	536366	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
9	536366	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
10	536367	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047	United Kingdom
11	536367	POPPY'S PLAYHOUSE BEDROOM	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
12	536367	POPPY'S PLAYHOUSE KITCHEN	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
13	536367	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	2010-12-01 08:34:00	3.75	13047	United Kingdom
14	536367	IVORY KNITTED MUG COSY	6	2010-12-01 08:34:00	1.65	13047	United Kingdom
15	536367	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	2010-12-01 08:34:00	4.25	13047	United Kingdom
16	536367	BOX OF VINTAGE JIGSAW BLOCKS	3	2010-12-01 08:34:00	4.95	13047	United Kingdom
17	536367	BOX OF VINTAGE ALPHABET BLOCKS	2	2010-12-01 08:34:00	9.95	13047	United Kingdom
18	536367	HOME BUILDING BLOCK WORD	3	2010-12-01 08:34:00	5.95	13047	United Kingdom
19	536367	LOVE BUILDING BLOCK WORD	3	2010-12-01 08:34:00	5.95	13047	United Kingdom
20	536367	RECIPE BOX WITH METAL HEART	4	2010-12-01 08:34:00	7.95	13047	United Kingdom
21	536367	DOORMAT NEW ENGLAND	4	2010-12-01 08:34:00	7.95	13047	United Kingdom
22	536368	JAM MAKING SET WITH JARS	6	2010-12-01 08:34:00	4.25	13047	United Kingdom

• magrittr- Provides a mechanism for chaining commands with a new forward-pipe operator, %>%. This operator will forward a value, or the result of an expression, into the next function call/expression. There is flexible support for the type of right-hand side expressions.

• dplyr - A fast, consistent tool for working with data frame like objects, both in memory and out of memory.

• tidyverse - This package is designed to make it easy to install and load multiple 'tidyverse' packages in a single step.

```
#Load excel in R dataframe i named it itemslist
itemslist <- read_excel('/Users/asik/Desktop/Assignment-1_Data.xlsx')
```

Data Pre-processing

Next, we need to upload Assignment-1_Data. xlsx to R to read the dataset.Now we can see our data in R.

```
#complete.cases(data) removing rows with missing values in any column of data frame
itemslist <- itemslist[complete.cases(itemslist), ]
```

After we will clear our

data frame, will remove missing values.

To apply Association Rule mining, we need to convert dataframe into transaction data to make all items that a

```r
#ddply(dataframe, variables_to_split_dataframe, function)
transaxtionData <- ddply(itemslist,c("BillNo","Date"),
                         function(df1)paste(df1$Itemname,
                                            collapse = ","))
```

ine all products from one BillNo and Date and combine all products from that BillNo and D

```r
transaxtionData$BillNo <- NULL
transaxtionData$Date <- NULL
#will gave the name to column "item"
colnames(transaxtionData) <- c("items")
```

We don't need BillNo and Date, we will make it as Null.

items			
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	KNITTED UNION FLAG HOT WATER BOTTLE
HAND WARMER UNION JACK	HAND WARMER RED POLKA DOT		
ASSORTED COLOUR BIRD ORNAMENT	POPPY'S PLAYHOUSE BEDROOM	POPPY'S PLAYHOUSE KITCHEN	FELTCRAFT PRINCESS CHARLOTTE DOLL
JAM MAKING SET WITH JARS	RED COAT RACK PARIS FASHION	YELLOW COAT RACK PARIS FASHION	BLUE COAT RACK PARIS FASHION
BATH BUILDING BLOCK WORD			
ALARM CLOCK BAKELIKE PINK	ALARM CLOCK BAKELIKE RED	ALARM CLOCK BAKELIKE GREEN	PANDA AND BUNNIES STICKER SHEET
PAPER CHAIN KIT 50'S CHRISTMAS			
HAND WARMER RED POLKA DOT	HAND WARMER UNION JACK		
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	EDWARDIAN PARASOL RED
VICTORIAN SEWING BOX LARGE			
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	EDWARDIAN PARASOL RED
HOT WATER BOTTLE TEA AND SYMPATHY	RED HANGING HEART T-LIGHT HOLDER		
HAND WARMER RED POLKA DOT	HAND WARMER UNION JACK		
JUMBO BAG PINK POLKADOT	JUMBO BAG BAROQUE BLACK WHITE	JUMBO BAG CHARLIE AND LOLA TOYS	STRAWBERRY CHARLOTTE BAG
JAM MAKING SET PRINTED			
RETROSPOT TEA SET CERAMIC 11 PC	GREY PINK TOOL SET	JUMBO SHOPPER VINTAGE RED PAISLEY	AIRLINE LOUNGE

Next, you have to store this transaction data into .csv

```r
#quote: If TRUE it will surround character or factor column with double quotes.
#If FALSE nothing will be quoted
#row.names: either a logical value indicating whether the row names of x are to be
#written along with x, or a character vector of row names to be written.
write.csv(transaxtionData, "assigment1_itemslist.csv", quote = FALSE, row.names = FALSE)
```

This how should look transaction data before we will go to next step.

At this step we already have our transaction dataset, and it shows the matrix of items which bought together. We can't see here any rules and how often it was purchase together. Now let's check how many transactions

```
transactions as itemMatrix in sparse format with
 [illegible] rows (elements/itemsets/transactions) and
 [illegible] columns (items) and a density of [illegible]

most frequent items:
WHITE HANGING HEART T-LIGHT HOLDER        REGENCY CAKESTAND 3 TIER        JUMBO BAG RED RETROSPOT
                              [illegible]                     [illegible]                [illegible]
                     PARTY BUNTING    ASSORTED COLOUR BIRD ORNAMENT                       (Other)
                              [illegible]                     [illegible]                [illegible]

element (itemset/transaction) length distribution:
sizes
[illegible]

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    [illegible]   [illegible]   [illegible]   [illegible]  [illegible]

includes extended item information - examples:
                labels
1             [illegible] HANGER
2     10 COLOUR SPACEBOY PEN
3 12 COLOURED PARTY BALLOONS
```

we have and what they are. We will have to have to load this transaction data into an object of the transaction class. This is done by using the R function read.transactions of the arules package. Our format of Data frame is basket.

```r
transactions <- read.transactions('/Users/asik/Desktop/assigment1_itemslist.csv',
                                  format = 'basket', sep=',')
```

```r
summary(transactions)
```

Let's have a view our transaction object by summary(transaction)

We can see 18193 transactions (rows) and 7698 items (columns). 7698 is the product descriptions and 18193 transactions are collections of these items.

The summary gives us some useful information:

- Density tells the percentage of non-zero cells in a sparse matrix. In other words, total number of items that are purchased divided by a possible number of items in that matrix. You can calculate how many items were purchased by using density: 18193x7698x0.002291294=337445
- Summary will show us most frequent items.
- Element (itemset/transaction) length distribution: It will gave us how many transactions are there for 1-itemset, 2-itemset and so on. The first row is telling you a number of items and the second row is telling you the number of transactions.

For example, there is only 1546 transaction for one item, 860 transactions for 2 items, and there are 419 items in one transaction which is the longest.

```r
itemFrequencyPlot(transactions,topN=20,type='absolute',
                  col=brewer.pal(8,'Pastel2'), main="Absolute Item Frequency Plot")
```

Let's check item frequency plot, we will generate an itemFrequencyPlot to create an item Frequency Bar Plot to view the distribution of objects based on itemMatrix (e.g., >transactions or items in >itemsets and >rules) which is our case.

```r
if (!require("RColorBrewer")) {install.packages("RColorBrewer")
  library(RColorBrewer)
```

In itemFrequencyPlot(transaction,topN=20,type="absolute") first argument - our transaction object to be plotted that is tr. topN is allows us to plot top N highest frequency items. type can be as type="absolute" or type="relative". If we will chouse absolute it will plot numeric frequencies of each item independently. If relative it will plot how many times these items have appeared as compared to others. As well I made it in colure for better visualization.

Generating Rules

Next, we will generate rules using the Apriori algorithm. The function apriori() is from package arules. The algorithm employs level-wise search for frequent itemsets. Algorithm will generate frequent itemsets and association rules. We pass supp=0.001 and conf=0.8 to return all the rules that have a support of at least 0.1% and confidence of at least 80%. We sort the rules by decreasing confidence and will check summary of the rules.

The apriori will take (transaction) as the transaction object on which mining is to be applied. parameter will allow you to set min_sup and min_confidence. The default values for parameter are minimum support of 0.1, the minimum confidence of 0.8, maximum of 10 items (maxlen).

Summary of rules give us clear information as:

- Number of rules: 97267
- The distribution of rules by length: a length of 6 items has the most 33296 and length of 2 items has lowest number of rules 111
- The summary of quality measures: ranges of support, confidence, and lift.
- The information on data mining: total data mined, and the minimum parameters we set earlier

Now, 97267 it a lot of rules. We will identify only top 10.

Using the above output, you can make analysis such as:

- 100% of the customers who bought 'ART LIGHTS ' also bought 'FUNK MONKEY'. • 100% of the customers who bought 'BILLBOARD FONTS DESIGN ' also bought 'WRAP'. We can limit the size and number of rules generated. we can set parameter in Apriori. If we want stronger rules, we must to increase the value of conf. and for more extended rules give higher value to maxlen.

Visualizing Association Rules

We have thousands of rules generated based on data, we will need a couple of ways to present our findings. We will use ItemFrequencyPlot to visualize association rules.

Scatter-Plot:

A straight-forward visualization of association rules is to use a scatter plot using plot() of the arulesViz package. It uses Support and Confidence on the axes. In addition, third measure Liftis used by default to color (grey levels) of the points.

Interactive Scatter-Plot:

We can have a look for each rule (interactively) and view all quality measures (support, confidence and lift).

Graph - Based Visualization and Group Method:

Graph plots are a great way to visualize rules but tend to become congested as the number of rules increases. So, it is better to visualize a smaller number of rules with graph-based visualizations. We can see as well group method for top 10 items.

Conclusion

Based on the results of these calculations can be used as a recommendation for retail owners to arrange the arrangement of product catalogs and take strategic steps to improve product marketing.. By utilizing the association rules which are discovered as a result of the analyses, the retailer can apply effective marketing and sales promotion strategies, he will be able increase customer engagement and improve customer experience and identify customer behavior.

PROGRAM

```
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori,
association_rules from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import sys
```

```python
if not sys.warnoptions:
 import warnings
 warnings.simplefilter("ignore")
```

In [2]:

```python
df = pd.read_csv('../input/market-basket-analysis/Assignment-1_Data.csv',
sep=';')
df.head()
```

Out[2]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850.0	United Kingdom

In [3]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 522064 entries, 0 to 522063
Data columns (total 7 columns):
```

```
 # Column Non-Null Count Dtype
--- ------ -------------- -----
 0 BillNo 522064 non-null object
 1 Itemname 520609 non-null object
 2 Quantity 522064 non-null int64
 3 Date 522064 non-null object
 4 Price 522064 non-null object
 5 CustomerID 388023 non-null float64
 6 Country 522064 non-null object
dtypes: float64(1), int64(1), object(5)
memory usage: 27.9+ MB
```

In [4]:

```
if df.isna().sum().sum() > 0:
 df = df.dropna()


df['Price'] = df['Price'].str.replace(',',
'.').astype('float64') df['CustomerID'] =
df['CustomerID'].astype('int')
df['Date'] = pd.to_datetime(df['Date'])
df['Itemname'] = df['Itemname'].str.strip()
df['Total_Price'] = df.Quantity * df.Price

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 388023 entries, 0 to 522063
Data columns (total 8 columns):
 # Column Non-Null Count Dtype
--- ------ -------------- -----
 0 BillNo 388023 non-null object
 1 Itemname 388023 non-null object
 2 Quantity 388023 non-null int64
 3 Date 388023 non-null datetime64[ns]
 4 Price 388023 non-null float64
 5 CustomerID 388023 non-null int64
 6 Country 388023 non-null object
 7 Total_Price 388023 non-null float64
dtypes: datetime64[ns](1), float64(2), int64(2),
object(3) memory usage: 26.6+ M
```

In [7]:

linkcode

```
country = input(" Write the country of the customer: ")
ID = int(input(" Write the customer's ID number: "))

def hot_encode(x):
 if(x<= 0):
 return 0
 if(x>= 1):
 return 1


def apriori_model(country = country, ID = ID):
 data = df[df['Country'] == country]
 today_date = max(data["Date"])
 #RFM
 rfm = data.groupby('CustomerID').agg({'Date': lambda Date: (today_date  -
Date.max()).days,
 'CustomerID': lambda CustomerID:  CustomerID.count(),
 'Total_Price': lambda Total_Price:  Total_Price.sum()})
 rfm.columns = ["recency", "frequency", "monetary"]
 scaler = StandardScaler().fit(rfm)
 rfm_scale = scaler.transform(rfm)
 #Kmeans
 kmeans = KMeans(n_clusters = 4, n_init=25, max_iter=300)
k_means = kmeans.fit(rfm_scale)
 segment = k_means.labels_
 rfm['segment'] = segment
 rfm = rfm.reset_index().rename(columns={'index': 'CustomerID'})
new_df = data.merge(rfm, right_on = 'CustomerID', left_on =
'CustomerID')

 #Apriori

 number_of_cluster = list(rfm[rfm['CustomerID'] == ID]['segment'])[0]

 apriori_df = new_df[new_df['segment'] == number_of_cluster ]
basket = (apriori_df.groupby(['BillNo', 'Itemname'])['Quantity']
.sum().unstack().reset_index().fillna(0)
 .set_index('BillNo'))
 # Encoding the datasets
 basket_encoded = basket.applymap(hot_encode)
 basket = basket_encoded
```

```python
 frq_items = apriori(basket, min_support = 0.03, use_colnames = True)
rules = association_rules(frq_items, metric ="lift", min_threshold =
0.8)
 rules = rules.sort_values(['confidence', 'lift'], ascending =[False,
False])
 return rules


rules = apriori_model(country=country, ID=ID)
rules.head()
```

Out[7]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
61	(CHILDS BREAKFAST SET DOLLY GIRL)	(CHILDS BREAKFAST SET SPACEBOY)	0.035971	0.043165	0.035971	1.0	23.166667	0.034419	inf
41	(CARD DOLLY GIRL)	(SPACEBOY BIRTHDAY CARD)	0.043165	0.057554	0.043165	1.0	17.375000	0.040681	inf
280	(POSTAGE, CARD DOLLY GIRL)	(SPACEBOY BIRTHDAY CARD)	0.043165	0.057554	0.043165	1.0	17.375000	0.040681	inf
283	(CARD DOLLY GIRL)	(SPACEBOY BIRTHDAY CARD, POSTAGE)	0.043165	0.057554	0.043165	1.0	17.375000	0.040681	inf
256	(ALARM CLOCK BAKELIKE PINK, ALARM CLOCK BAKELI...	(ALARM CLOCK BAKELIKE RED)	0.035971	0.064748	0.035971	1.0	15.444444	0.033642	inf