National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Amol Upadhyay
Student ID: X20110812

School of Computing
National College of Ireland

Supervisor: Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Amol Upadhyay
Student ID:	X20110812
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Giovani Estrada
Submission Due Date:	15/12/2022
Project Title:	Configuration Manual
Word Count:	414
Page Count:	10

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:	15th December 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

Amol Upadhyay
X20110812

1 Introduction

A complete configuration manual contains critical information on the system, hardware, and software specs. It also displays the whole flow employed to execute the study on the "An investigation of e-commerce buying behaviour across the UK and Brazil". Section 2 of the handbook discusses system specifications, such as hardware and software setups. In Section 3, you'll learn how to set up the environment, import key libraries, and do pre-processing. The fourth segment discusses model development and assessment.

2 Configuration of System

System setup discussed the hardware and software requirements for the investigation.

2.1 Specification of Hardware

OS(operating System)	Windows 10
RAM	16GB
System Processor	i6intel
Speed in Hz	3.2GHz
Disk Memory	1GB approx.
GPU	NVIDIA

2.2 Specification of Software

Programming Language	Python 3.9 version
Other Softwares	Anaconda & jupyter
Web Browsers	Google Chrome

3 Setting Environment

3.1 Launching Jupyter on Anaconda

The very first stage is to activate the Anaconda app. Anaconda delivers a wealth of relevant software to meet your requirements. Jupyter Notebook used to implement the code above, and it comes with the most recent version of Python. There are many excellent Python libraries for analysis.

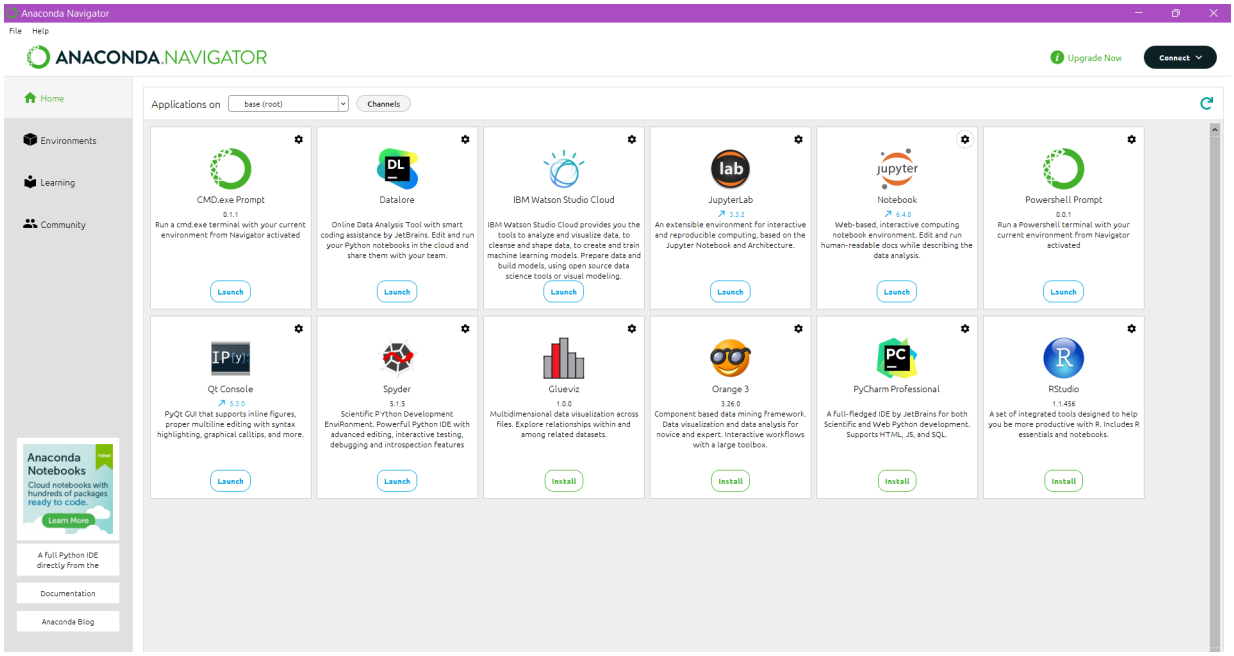

Figure 1: anaconda interface

This is the main page for Jupyter Notebook; to begin coding, create a new ipynb file.

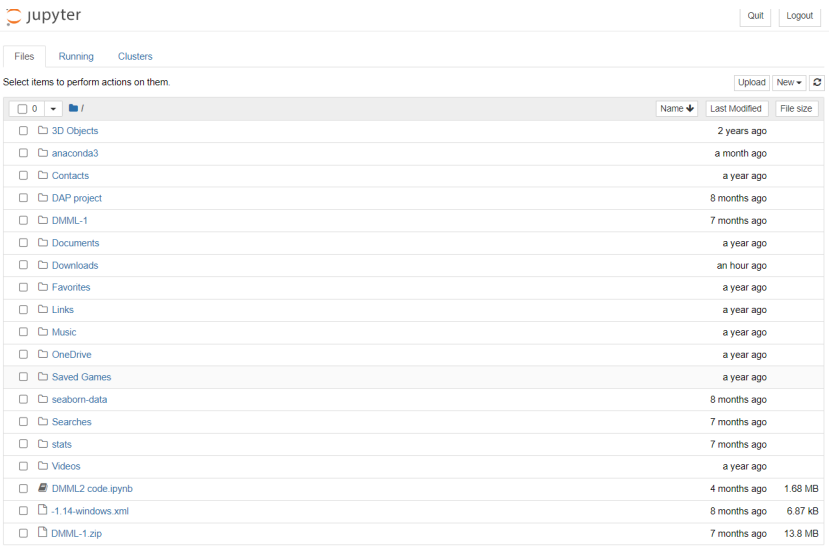

Figure 2: Jupyter

3.2 Data Preparation

Keep all the dataset files in one folder before diving into the code. All the required files are in one place in the folder MachineLearningCVE.

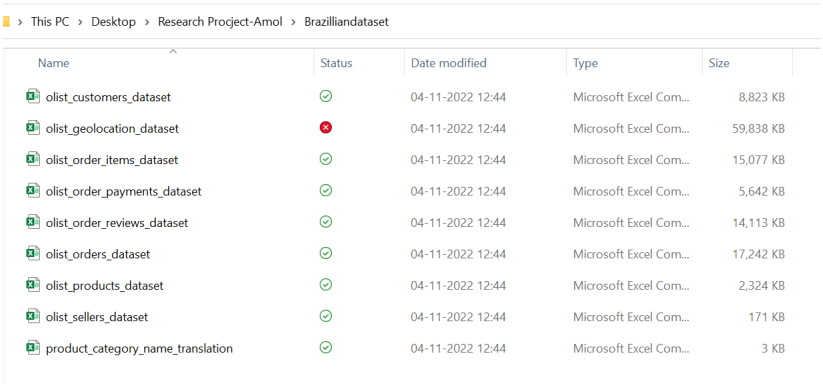

Figure 3: CSV Files-Brazil

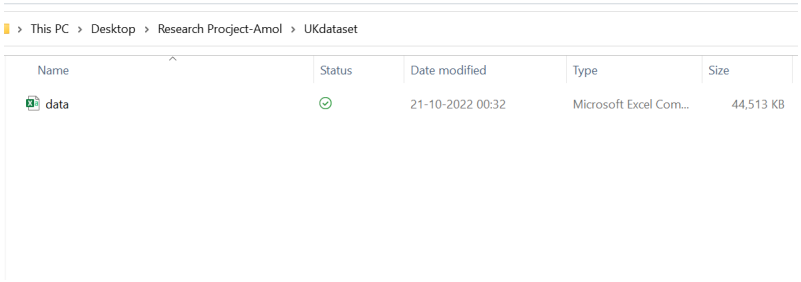

Figure 4: CSV File-UK

3.3 Importing Important Libraries

It displays all implementation-required libraries, and libraries that need installation may be installed with "pip."

```
import pandas as pd
import numpy as np
import urllib
import unidecode
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
from scipy import stats
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings("ignore")
```

Figure 5: Libraries for Brazil Market Analysis

```
import missingno as msno

import gc
import datetime as dt

%matplotlib inline
color = sns.color_palette()

from matplotlib.ticker import PercentFormatter
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
from itertools import combinations
import statsmodels.api as sm
import warnings
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from scipy import stats
from sklearn.metrics import r2_score, mean_squared_error

pd.options.mode.chained_assignment = None

plt.rcParams["axes.facecolor"] = "#A2A2A2"
plt.rcParams["axes.grid"] = 1
```

Figure 6: Libraries for UK Market Analysis

```
#Import all relevant libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

Figure 7: Libraries for Market Basket Analysis for both Uk and Brazil

3.4 Importing Dataset

The pandas package is used to load each CSV file into a python data frame.

```
In [3]: df = pd.read_csv('C:/Users/Amol/OneDrive/Desktop/Research Procject-Amol/UKdataset/data.csv', encoding='latin')
        # df = pd.read_csv('data.csv', encoding='latin')
        df.head()
```

Figure 8: Importing UK Dataset

```
import missingno as msno

import gc
import datetime as dt

%matplotlib inline
color = sns.color_palette()

from matplotlib.ticker import PercentFormatter
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
from itertools import combinations
import statsmodels.api as sm
import warnings
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from scipy import stats
from sklearn.metrics import r2_score, mean_squared_error

pd.options.mode.chained_assignment = None

plt.rcParams["axes.facecolor"] = "#A2A2A2"
plt.rcParams["axes.grid"] = 1
```

Figure 9: Importing MBA UK Dataset

Figure 10: Importing MBA Brazil Dataset

3.5 Data Pre-processing

In this part processes like eliminating missing numbers, outliers, etc are done. During this process, data is changed into a form that can be used in modeling.

3.5.1 Brazil Dataset Pre-processing

Figure 11: Data Pre-processing part 1

Figure 12: Data Pre-processing part 2

```
le = LabelEncoder()
dframe[' Label']= le.fit_transform(dframe[' Label'])
```

```
X= dframe.drop([' Label'], axis=1)
X[X < 0] = 0
print(X)
y = dframe[' Label']
del dframe
```

```
X_new = SelectKBest(chi2, k=30).fit_transform(X, y)
X_new.shape
```

```
(2062474, 30)
```

```
X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size=0.25, random_state=1234)
```

```
St_scalar = StandardScaler()
X_train = St_scalar.fit_transform(X_train)
X_test  = St_scalar.transform(X_test)
```

Figure 13: More Data Pre-processing Steps

3.5.2 UK Dataset Pre-processing

```
In [6]: df.rename(columns={"InvoiceNo":"invoice_num",
                          "StockCode":"stock_code",
                          "Description":"description",
                          "Quantity":"quantity",
                          "InvoiceDate":"invoice_date",
                          "UnitPrice":"unit_price",
                          "CustomerID":"customer_id",
                          "Country":"country"}, inplace=True)

In [7]: df['invoice_date']=pd.to_datetime(df.invoice_date, format='%m/%d/%Y %H:%M')

In [8]: df['description']=df.description.str.lower()

In [9]: df.isnull().sum()
```

Figure 14: Data Pre-processing part 1

```
In [20]: df_miss["day"] = df_miss['invoice_date'].map(lambda x: x.day)
         df_miss["month"] = df_miss['invoice_date'].map(lambda x: x.month)
         df_miss["year"] = df_miss['invoice_date'].map(lambda x: x.year)

In [21]: df_miss['daymonth']=df_miss['day'].astype(str)+'/'+df_miss['month'].astype(str)
         df_miss['daymonthyear']=df_miss['daymonth'].astype(str)+'/'+df_miss['year'].astype(str)
         df_miss['monthyear']=df_miss['month'].astype(str)+'/'+df_miss['year'].astype(str)
```

Figure 15: Data Pre-processing part 2

```
In [33]: df_new.duplicated().sum()
Out[33]: 5225

In [34]: df_new.drop_duplicates(inplace=True)

In [35]: df_new.duplicated().sum()
Out[35]: 0

In [36]: df_new['customer_id']=df_new.customer_id.astype('int64')
```

Figure 16: Data Pre-processing Part 3

```
In [49]: import re
         spec_list=[]
         for code in df_new.stock_code:
             x=re.findall(r"^\w{1}$|\D[A-Z]+\D|[A-Z]\d", code)
             if x not in spec_list:
                 if len(x) >0 :
                     spec_list.append(x)
         spec_list

Out[49]: [['POST'], ['PADS'], ['M'], ['DOT'], ['C2'], ['BANK ', 'CHARGES']]

In [50]: spec_list[5] = ['BANK CHARGES']

In [51]: spec_list2=[item for sublist in spec_list for item in sublist]
         spec_list2

Out[51]: ['POST', 'PADS', 'M', 'DOT', 'C2', 'BANK CHARGES']

In [52]: df_new[df_new['stock_code'].apply(lambda x: x in spec_list2)]
```

Figure 17: Data Pre-processing Part 4

3.5.3 MBA Data Pre-processing for Both Brazil & UK

```
#If you want to cross check the data you can replace 10 with 80.
item_freq = data['product_id'].value_counts()
data = data[data.isin(item_freq.index[item_freq >= 10]).values]
data['product_id'].value_counts()

422879e10f46682990de24d770e7f83d    137
aca2eb7d00ea1a7b8ebd4e68314663af    130
99a4788cb24856965c36a24e339b6058    125
368c6c730842d78016ad823897a372db    113
389d119b48cf3043d311335e499d9c6b    104
                                   ...
65266b2da20d04dbe00c5c2d3bb7859e     10
4c1e109ecdf58453de365d217cefa64c     10
657247f6f60543b93e3cc708a8e6329c     10
98d61056e0568ba048e5d78038790e77     10
c7fd13b5e515bffdab855d0812842edb     10
Name: product_id, Length: 329, dtype: int64

#Average products purchased per transaction
data['order_id'].value_counts().mean()

1.154075546719682

#Create a basket of all products, orders and quantity
basket_brazil = (data.groupby(['order_id','product_id'])['quantity'].sum().unstack().reset_index().fillna(0).set_index('order_id
basket_brazil.head()
```

Figure 18: Data Pre-processing for MBA Brazil

```
df.dropna(axis = 0, subset = ['CustomerID'], inplace = True)
print('Dataframe dimensions:', df.shape)
#
# gives some infos on columns types and numer of null values
tab_info=pd.DataFrame(df.dtypes).T.rename(index={0:'column type'})
tab_info=tab_info.append(pd.DataFrame(df.isnull().sum()).T.rename(index={0:'null values (nb)'}))
tab_info=tab_info.append(pd.DataFrame(df.isnull().sum()/df.shape[0]*100).T.
                         rename(index={0:'null values (%)'}))
display(tab_info)
#We deleted ~25% of null present in the CustomerID variable, as all these products are not associated with any Customers.
```

Figure 19: Data Pre-processing MBA UK part 1

```
#After deleting the Null values previously, we have now dropped all the duplicate values present in the dataset.
print('Entrées dupliquées: {}'.format(df.duplicated().sum()))
df.drop_duplicates(inplace = True)

Entrées dupliquées: 5225

#Remove rows with invoices that contain a "C"
df = df[~df["InvoiceNo"].str.contains("C")]

len(df)
```

Figure 20: Data Pre-processing MBA UK Part 2

4 Data Visualization

All of the conclusions obtained from the data are presented here.

4.1 Exploratory Data Analysis

4.1.1 EDA and RFM analysis of UK Data

```
customer_counts = df_new.customer_id.value_counts().sort_values(ascending=False).head(10)

fig, ax = plt.subplots(figsize = (10, 8))

sns.barplot(y = customer_counts.index, x = customer_counts.values, orient = "h",
            ax = ax, order = customer_counts.index, palette = "Reds_r")

plt.title("Customers that have most transactions:Top 10")
plt.ylabel("Customers")
plt.xlabel("Transaction Count")

plt.show()
```

Figure 21: EDA part 1

```
ax = df_new.groupby('invoice_num')['day'].unique().value_counts().sort_index().plot(kind='bar',color=color[0],figsize=(10,5))
ax.set_xlabel('Day',fontsize=15)
ax.set_ylabel('Number of Orders',fontsize=15)
ax.set_title('Number of orders for different Days',fontsize=15)
ax.set_xticklabels(('Mon','Tue','Wed','Thur','Fri','Sun'), rotation='horizontal', fontsize=15)
plt.show()
```

Figure 22: EDA Part 2

```
group_country_orders = df_new.groupby('country')['invoice_num'].count().sort_values()
# del group_country_orders['United Kingdom']

# plot number of unique customers in each country (with UK)
plt.subplots(figsize=(15,8))
group_country_orders.plot(kind='barh', fontsize=12, color=color[0])
plt.xlabel('Number of Orders', fontsize=12)
plt.ylabel('Country', fontsize=12)
plt.title('Number of Orders for different Countries', fontsize=12)
plt.show()
```

Figure 23: EDA Part 3

4.1.2 EDA and RFM analysis of Brazil Data

```
other_state_geolocation = geo_df.groupby(['zip_code'])['geolocation_state'].nunique().reset_index(name='count')
other_state_geolocation[other_state_geolocation['count']>= 2].shape
max_state = geo_df.groupby(['zip_code','geolocation_state']).size().reset_index(name='count').drop_duplicates(subset = 'zip_code'
```

```
geo_updated = geo_df.groupby(['zip_code','geolocation_city','geolocation_state'])[['geolocation_lat','geolocation_lng']].median()
geo_updated = geo_updated.merge(max_state,on=['zip_code','geolocation_state'],how='inner')
```

```
geo_updated = customers_df.merge(geo_updated,left_on='zip_code',right_on='zip_code',how='inner')
```

```
def plot_brasil_map(data):
    brazil = plt.imread(urllib.request.urlopen('https://i.pinimg.com/originals/3a/0c/e1/3a0ce18b3c842748c255bc0aa445ad41.jpg'),'
    ax = data.plot(kind="scatter", x="geolocation_lng", y="geolocation_lat", figsize=(10,10), alpha=0.3,s=0.3,c='blue')
    plt.axis('off')
    plt.imshow(brazil, extent=[-73.98283055, -33.8,-33.75116944,5.4])
    plt.show()
```

Figure 24: EDA part 1

```
sns.countplot(data["review_score"])
data["review_score"].value_counts() / data["review_score"].count() * 100
```

Figure 25: EDA Part 2

```
top_prod_categ = data.groupby('product_category_name_english')['order_id'].nunique().sort_values(ascending=False).head(10)
top_prod_categ



plt.figure(figsize=(15,7))


sns.barplot(y=top_prod_categ.index, x=top_prod_categ, palette="Set2")
plt.xlabel('Number of orders')
plt.title('Top 10 Product categories in terms of number of order ')
```

Figure 26: EDA Part 3

5 Segmentation Technique and MBA techniques on UK and Brazil Data

5.1 Segmentation Technique on Brazil and UK Data

```
# Train the model on 4 clusters
kmean_model = KMeans(n_clusters=4, random_state=5)
kmean_y = kmean_model.fit_predict(RFM_Table_scaled)
# Add labels to df
rfm_df['Cluster'] = kmean_model.labels_
```

```
# Function to visualize clusters
def rfm_values(df):
    df_new = df.groupby(['Cluster']).agg({
            'Recency': 'mean',
            'Frequency': 'mean',
            'Monetary': ['mean', 'count']
        }).round(0)

    return df_new
```

```
rfm_values(rfm_df)
```

Figure 27: RFM Technique on Brazil Data

```
In [354]: # Train the model on 6 clusters
          kmean_model = KMeans(n_clusters=6, random_state=5)
          kmean_y = kmean_model.fit_predict(RFM_Table_scaled)
          # Add labels to df
          rfm_df['Cluster'] = kmean_model.labels_

In [355]: # Function to visualize clusters
          def rfm_values(df):
              rfm_df = df.groupby(['Cluster']).agg({
                      'Recency': 'mean',
                      'Frequency': 'mean',
                      'Monetary': ['mean', 'count']
                  }).round(0)

              return rfm_df

In [356]: rfm_values(rfm_df)
```

Figure 28: RFM Technique on UK Data

5.2 Market Basket Analysis on UK and Brazil data

```
#Create Rules
rules = association_rules(frequent_itemsets, metric = 'support', min_threshold = 0.0001)
rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	convictio
0	(e0cf79767c5b016251fe139915c59a26)	(0a4093a4af429dc0a9334300e5c13ae5)	0.005467	0.001491	0.000166	0.030303	20.323232	0.000158	1.02971
1	(0a4093a4af429dc0a9334300e5c13ae5)	(e0cf79767c5b016251fe139915c59a26)	0.001491	0.005467	0.000166	0.111111	20.323232	0.000158	1.11884
2	(0aabfb375647d9738ad0f7b4ea3653b1)	(6c3effec7c8ddba466d4f03f982c7aa3)	0.006958	0.005302	0.000331	0.047619	8.982143	0.000294	1.04443
3	(6c3effec7c8ddba466d4f03f982c7aa3)	(0aabfb375647d9738ad0f7b4ea3653b1)	0.005302	0.006958	0.000331	0.062500	8.982143	0.000294	1.05924
4	(0bcc3eeca39e1064258aa1e932269894)	(368c6c730842d78016ad823897a372db)	0.003479	0.014414	0.000166	0.047619	3.303777	0.000116	1.03486
...	...	...	...	...	...	...	...	...	...
125	(53759a2ecddad2bb87a079a1f1519f73, 389d119b48c...	(422879e10f46682990de24d770e7f83d, 0bcc3eeca39...	0.000663	0.000166	0.000166	0.250000	1509.000000	0.000166	1.33311
126	(422879e10f46682990de24d770e7f83d)	(53759a2ecddad2bb87a079a1f1519f73, 0bcc3eeca39...	0.017396	0.000166	0.000166	0.009524	57.485714	0.000163	1.00944
127	(0bcc3eeca39e1064258aa1e932269894)	(422879e10f46682990de24d770e7f83d, 53759a2ecdd...	0.003479	0.000166	0.000166	0.047619	287.428571	0.000165	1.04982
128	(53759a2ecddad2bb87a079a1f1519f73)	(422879e10f46682990de24d770e7f83d, 0bcc3eeca39...	0.012757	0.000166	0.000166	0.012987	78.389610	0.000164	1.01299
129	(389d119b48cf3043d311335e499d9c6b)	(422879e10f46682990de24d770e7f83d, 0bcc3eeca39...	0.013917	0.000166	0.000166	0.011905	71.857143	0.000163	1.01188

130 rows × 9 columns

```
#Products having 50% confidence likely to be purchased together
rules[rules['confidence'] >= 0.50]
```

Figure 29: MBA for Brazil

```
# Applying apriori algorithm
#Generate the frequent itemsets
frequent_itemsets = apriori(basket_filtered, min_support=0.03, use_colnames=True).sort_values("support",ascending=False)
frequent_itemsets.head(10)
```

```
C:\Users\Amol\anaconda3\lib\site-packages\mlxtend\frequent_patterns\fpcommon.py:111: DeprecationWarning: DataFrames with non-bo
ol types result in worse computationalperformance and their support might be discontinued in the future.Please use a DataFrame
with bool type
  warnings.warn(
```

	support	itemsets
99	0.121358	(WHITE HANGING HEART T-LIGHT HOLDER)
44	0.093197	(JUMBO BAG RED RETROSPOT)
80	0.090466	(REGENCY CAKESTAND 3 TIER)
6	0.084417	(ASSORTED COLOUR BIRD ORNAMENT)
71	0.082986	(PARTY BUNTING)
58	0.072841	(LUNCH BAG RED RETROSPOT)
86	0.064971	(SET OF 3 CAKE TINS PANTRY DESIGN)
52	0.064646	(LUNCH BAG BLACK SKULL)
69	0.061004	(PAPER CHAIN KIT 50'S CHRISTMAS)
64	0.060939	(NATURAL SLATE HEART CHALKBOARD)

```
#Apply association rules
assoc_rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1).sort_values("lift",ascending=False).reset_inde
assoc_rules
```

Figure 30: MBA for UK