**IBM- Naan Mudhalvan Project phase-02**

**Student Name :** INFANT SURYA.B

**Register Number :** 620821243045

**Branch** **:** B. TECH AI&DS

**Year**  **:** 3rd year

**Topic :** Artificial Intelligence

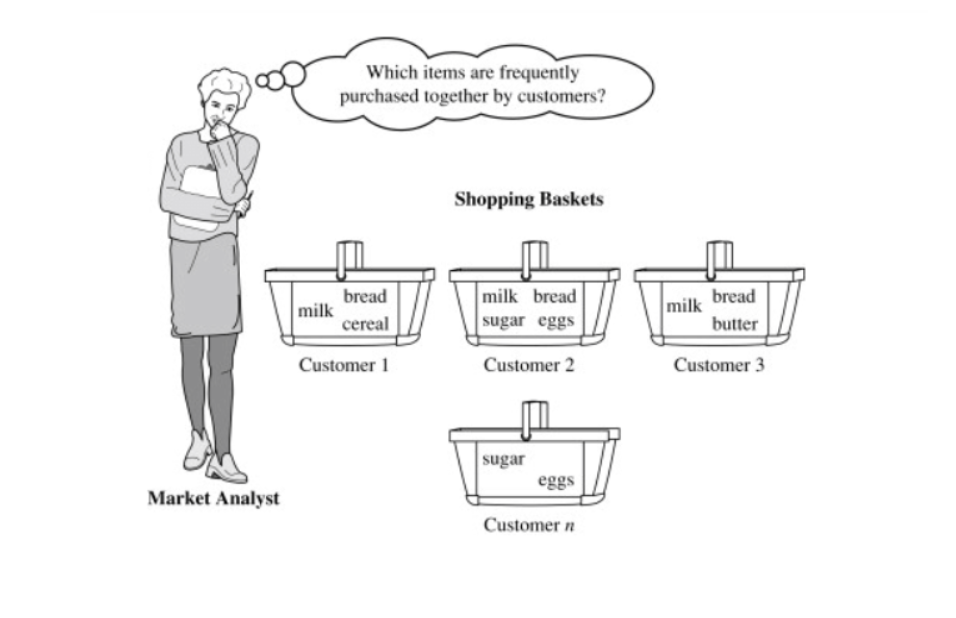
**Title :** Market Basket Insights

**College :** Gnanamani College of Technology

**MARKET BASKET ANALYSIS**

**ABSTRACT**

In this project, I will be implementing Market Basket Analysis, a data mining technique used to identify associations and patterns in customers' purchasing behaviour. By analysing transactional data, we aim to uncover relationships between products frequently bought together, which can provide valuable insights for businesses, such as cross-selling opportunities, optimizing product placement, and improving marketing strategies. And **I using a jupyter notebook, so code is one by one output processes.**



**Feature Descriptions**

Bill NO : A 6-digit unique number assigned to each Transaction.

Item name : Name of the item purchased.

Quantity : The quantity of each item being purchased

Date : Transaction Data

Price : Price of each Item

Customer Id : Unique Id of each customer

Country : The name of the country where each customer resides.

**PROGRAME:**

**import pandas as** **pd**

**import numpy as** **np**

**import matplotlib.pyplot as** **plt**

**import seaborn as** **sns**

**%matplotlib inline**

**import** **warnings**

**warnings.filterwarnings('ignore')**

**pd.set\_option('display.max\_columns', None)**

**data=pd.read\_excel('"C:\Users\infan\Downloads\Market basket\_Data.xlsx')**

**data.head()**

**OUTPUT ON HEAD ()**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bill No | Item name | Quantity | Date | Price | Customer ID | Country |  |
| 0 | 536365 | WHITE HANGING HEART T-LIGHT HOLDER | 6 | 2010-12-01 08:26:00 | 2.55 | 17850.0 | United Kingdom |
| 1 | 536365 | WHITE METAL LANTERN | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom |
| 2 | 536365 | CREAM CUPID HEARTS COAT HANGER | 8 | 2010-12-01 08:26:00 | 2.75 | 17850.0 | United Kingdom |
| 3 | 536365 | KNITTED UNION FLAG HOT WATER BOTTLE | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom |
| 4 | 536365 | RED WOOLLY HOTTIE WHITE HEART. | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom |

data.Tail()

OUTPUT **DATA.TAI**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bill No | Item name | Quantity | Date | Price | Customer ID | Country |  |
| 522059 | 581587 | PACK OF 20 SPACEBOY NAPKINS | 12 | 2011-12-09 12:50:00 | 0.85 | 12680.0 | France |
| 522060 | 581587 | CHILDREN'S APRON DOLLY GIRL | 6 | 2011-12-09 12:50:00 | 2.10 | 12680.0 | France |
| 522061 | 581587 | CHILDRENS CUTLERY DOLLY GIRL | 4 | 2011-12-09 12:50:00 | 4.15 | 12680.0 | France |
| 522062 | 581587 | CHILDRENS CUTLERY CIRCUS PARADE | 4 | 2011-12-09 12:50:00 | 4.15 | 12680.0 | France |
| 522063 | 581587 | BAKING SET 9 PIECE RETROSPOT | 3 | 2011-12-09 12:50:00 | 4.95 | 12680.0 | France |

data.info ()

<class 'pandas.core.frame.DataFrame'>

Range Index: 522064 entries, 0 to 522063

Data columns (total 7 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Bill No 522064 non-null object

1 Item name 520609 non-null object

2 Quantity 522064 non-null int64

3 Date 522064 non-null datetime64[ns]

4 Price 522064 non-null float64

5 Customer ID 388023 non-null float64

6 Country 522064 non-null object

dtypes: datetime64[ns](1), float64(2), int64(1), object(3)

memory usage: 27.9+ MB

data.shape

(522064, 7)

data.describe()

data[data['Quantity']<0]



from wordcloud import WordCloud, STOPWORDS

stopwords = STOPWORDS

worldcloud= WordCloud(background\_color='Black',stopwords=stopwords, height=1000, width =2000)

temp=data[data['Quantity']<0]

body =temp['Itemname'].to\_string(index=False)

worldcloud.generate(body)

plt.figure(figsize=(22,10))

plt.imshow(worldcloud)

plt.axis("off")

temp=data[data['Price']<=0]

body =temp['Itemname'].dropna().to\_string(index=False)

worldcloud.generate(body)

plt.figure(figsize=(22,10))

plt.imshow(worldcloud)

plt.axis("off")

data.duplicated().sum()

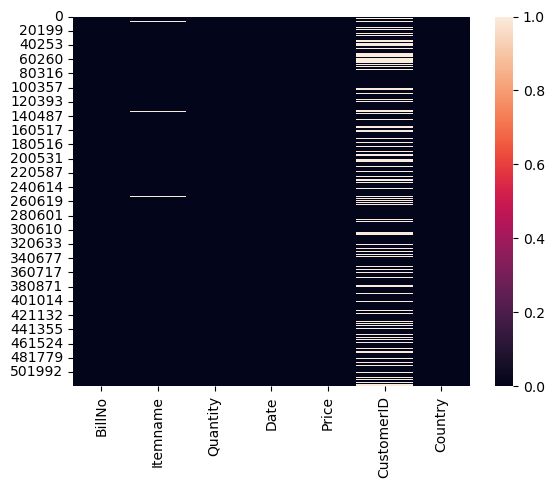
data.drop\_duplicates(inplace=True)

data['Itemname'] = data['Itemname'].str.strip()

data.isnull().sum()

data.isnull().mean()\*100

sns.heatmap(data.isnull())



data['Date']

0 2010-12-01 08:26:00

1 2010-12-01 08:26:00

2 2010-12-01 08:26:00

3 2010-12-01 08:26:00

4 2010-12-01 08:26:00

...

522059 2011-12-09 12:50:00

522060 2011-12-09 12:50:00

522061 2011-12-09 12:50:00

522062 2011-12-09 12:50:00

522063 2011-12-09 12:50:00

Name: Date, Length: 516778, dtype: datetime64[ns]

import datetime as datetime

from datetime import datetime

data['date'] = data['Date'].dt.date

data['hour'] = data['Date'].dt.hour

data['date']= pd.to\_datetime(data['date'], infer\_datetime\_format= True)

data.drop('Date',inplace=True,axis=1)

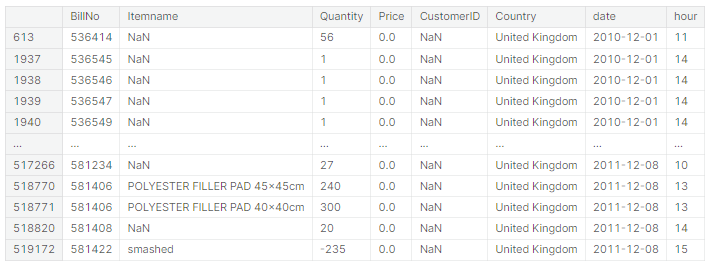
data.head(3)



data[data['Quantity']<=0]



data[data['Price']<=0]



data=data[data['Quantity']>0]

data=data[data['Price']>0]

data. Shape

plt.figure(figsize=(22,7))

plt.subplot(1,2,1)

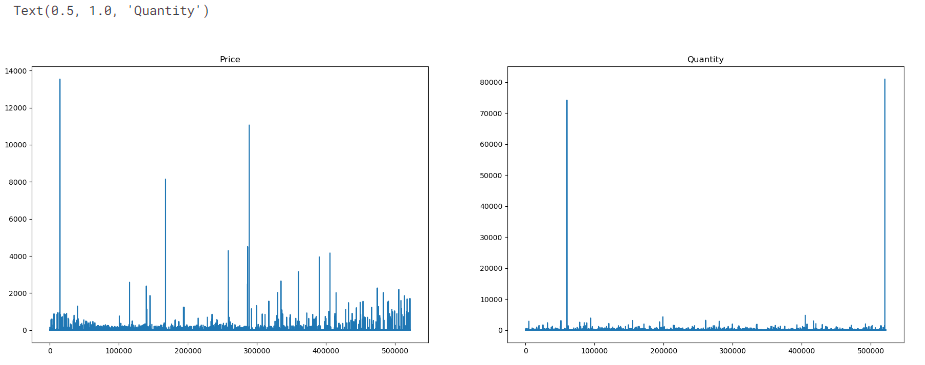
data.Price.plot()

plt.title("Price")

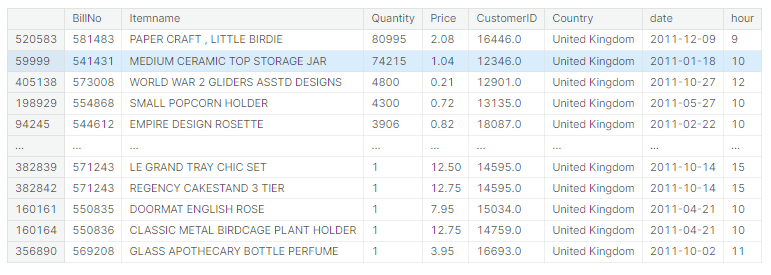
plt.subplot(1,2,2)

data.Quantity.plot()

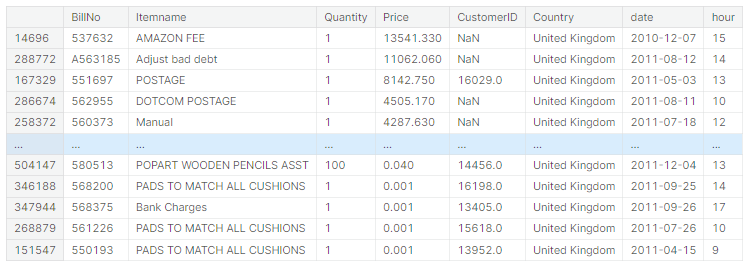
plt.title("Quantity")



data.sort\_values(by='Quantity',ascending=False)



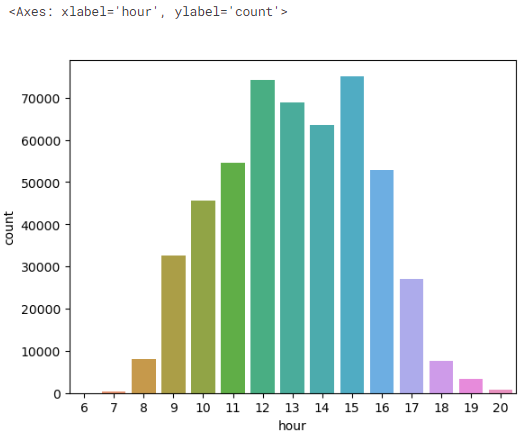
data.sort\_values(by='Price',ascending=False)



data=data[data['Price']<5000]

data=data[data['Quantity']<5000]

sns.countplot(data=data,x='hour')



# Get the top 10 item names by count

top\_10\_items = data['Itemname'].value\_counts().nlargest(10).index

# Filter the DataFrame to include only the top 10 item names

df\_top\_10 = data[data['Itemname'].isin(top\_10\_items)]

# Create a countplot for the top 10 item names

ax=sns.countplot(data=df\_top\_10, x='Itemname')

plt.xticks(rotation=90

(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),

[Text(0, 0, 'WHITE HANGING HEART T-LIGHT HOLDER'),

Text(1, 0, 'ASSORTED COLOUR BIRD ORNAMENT'),

Text(2, 0, 'LUNCH BAG RED RETROSPOT'),

Text(3, 0, 'PACK OF 72 RETROSPOT CAKE CASES'),

Text(4, 0, 'NATURAL SLATE HEART CHALKBOARD'),

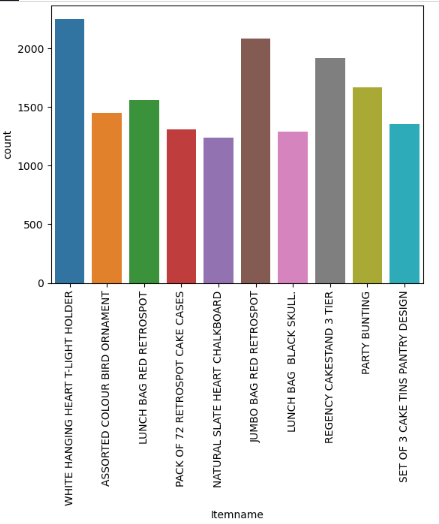
Text(5, 0, 'JUMBO BAG RED RETROSPOT'),

Text(6, 0, 'LUNCH BAG BLACK SKULL.'),

Text(7, 0, 'REGENCY CAKESTAND 3 TIER'),

Text(8, 0, 'PARTY BUNTING'),

Text(9, 0, 'SET OF 3 CAKE TINS PANTRY DESIGN')])



top\_5\_country = data['Country'].value\_counts().nlargest(10).index

df\_top\_5 = data[data['Country'].isin(top\_5\_country)]

ax=sns.countplot(data=df\_top\_5, x='Country')

plt.xticks(rotation=90)

data.Country.value\_counts().head(10)

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

mybasket= (data[data['Country'] =="Germany"]

.groupby(['BillNo', 'Itemname'])['Quantity']

.sum().unstack().reset\_index().fillna(0)

.set\_index('BillNo'))

mybasket

a=data[data['Country']=='Germany']

a['BillNo'].nunique()

def my\_encode\_units(x):

if x <= 0:

return 0

if x >= 1:

return 1

my\_basket = mybasket.applymap(my\_encode\_units)

my\_basket.drop('POSTAGE', inplace=True, axis=1) #Remove "postage" as an item

### Display sample of set

my\_basket

data[data['BillNo']==536527]

support=[0.1, 0.05, 0.01]

confidenceLevels=[0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1]

# Empty lists

rules\_sup10=[0]\*9

rules\_sup5=[0]\*9

rules\_sup1=[0]\*9

#rules for support 0.1

my\_frequent\_itemsets01 = apriori(my\_basket, min\_support=0.1, use\_colnames=True)

for i in range(len(confidenceLevels)):

rules\_sup10[i]=len(association\_rules(my\_frequent\_itemsets01, metric="confidence", min\_threshold=confidenceLevels[i]))

#rules for support 0.05

my\_frequent\_itemsets005 = apriori(my\_basket, min\_support=0.05, use\_colnames=True)

for i in range(len(confidenceLevels)):

rules\_sup5[i]=len(association\_rules(my\_frequent\_itemsets005, metric="confidence", min\_threshold=confidenceLevels[i]))

#rules for support 0.01

my\_frequent\_itemsets001 = apriori(my\_basket, min\_support=0.01, use\_colnames=True)

for i in range(len(confidenceLevels)):

rules\_sup1[i]=len(association\_rules(my\_frequent\_itemsets001, metric="confidence", min\_threshold=confidenceLevels[i]))

# Plot the data

plt.figure(figsize=(22,5))

plt.subplot(1,3,1)

plt.plot(confidenceLevels,rules\_sup10, marker='o', label='Support=0.1')

plt.title("No.of Rules for support=0.1")

plt.xlabel("Confidence Level")

plt.ylabel("No.of Rules")

plt.subplot(1,3,2)

plt.plot(confidenceLevels,rules\_sup5, marker='o', label='Support=0.1')

plt.title("No.of Rules for support=0.05")

plt.xlabel("Confidence Level")

plt.ylabel("No.of Rules")

plt.subplot(1,3,3)

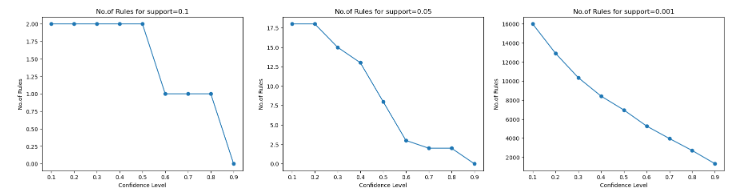
plt.plot(confidenceLevels,rules\_sup1, marker='o', label='Support=0.1')

plt.title("No.of Rules for support=0.001")

plt.xlabel("Confidence Level")

plt.ylabel("No.of Rules")

plt.savefig("comparision")



my\_frequent\_itemsets = apriori(my\_basket, min\_support=0.07, use\_colnames=True)

myrules=association\_rules(my\_frequent\_itemsets, metric="lift", min\_threshold=1)

myrules

mybasket= (data[data['Country'] =="France"]

.groupby(['BillNo', 'Itemname'])['Quantity']

.sum().unstack().reset\_index().fillna(0)

.set\_index('BillNo'))

mybasket

def my\_encode\_units(x):

if x <= 0:

return 0

if x >= 1:

return 1

my\_basket = mybasket.applymap(my\_encode\_units)

my\_basket.drop('POSTAGE', inplace=True, axis=1) #Remove "postage" as an item

### Display sample of set

my\_basket

support=[0.1, 0.05, 0.01]

confidenceLevels=[0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1]

# Empty lists

rules\_sup10=[0]\*9

rules\_sup5=[0]\*9

rules\_sup1=[0]\*9

#rules for support 0.1

my\_frequent\_itemsets01 = apriori(my\_basket, min\_support=0.1, use\_colnames=True)

for i in range(len(confidenceLevels)):

rules\_sup10[i]=len(association\_rules(my\_frequent\_itemsets01, metric="confidence", min\_threshold=confidenceLevels[i]))

#rules for support 0.05

my\_frequent\_itemsets005 = apriori(my\_basket, min\_support=0.05, use\_colnames=True)

for i in range(len(confidenceLevels)):

rules\_sup5[i]=len(association\_rules(my\_frequent\_itemsets005, metric="confidence", min\_threshold=confidenceLevels[i]))

#rules for support 0.01

my\_frequent\_itemsets001 = apriori(my\_basket, min\_support=0.01, use\_colnames=True)

for i in range(len(confidenceLevels)):

rules\_sup1[i]=len(association\_rules(my\_frequent\_itemsets001, metric="confidence", min\_threshold=confidenceLevels[i]))

# Plot the data

plt.figure(figsize=(22,5))

plt.subplot(1,3,1)

plt.plot(confidenceLevels,rules\_sup10, marker='o', label='Support=0.1')

plt.title("No.of Rules for support=0.1")

plt.xlabel("Confidence Level")

plt.ylabel("No.of Rules")

plt.subplot(1,3,2)

plt.plot(confidenceLevels,rules\_sup5, marker='o', label='Support=0.1')

plt.title("No.of Rules for support=0.05")

plt.xlabel("Confidence Level")

plt.ylabel("No.of Rules")

plt.subplot(1,3,3)

plt.plot(confidenceLevels,rules\_sup1, marker='o', label='Support=0.1')

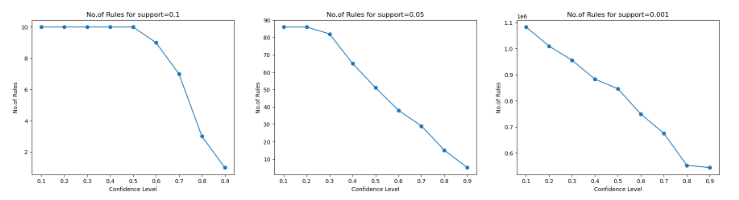
plt.title("No.of Rules for support=0.001")

plt.xlabel("Confidence Level")

plt.ylabel("No.of Rules")

plt.savefig("comparision")

**VISUALIZATION OUTPUT**

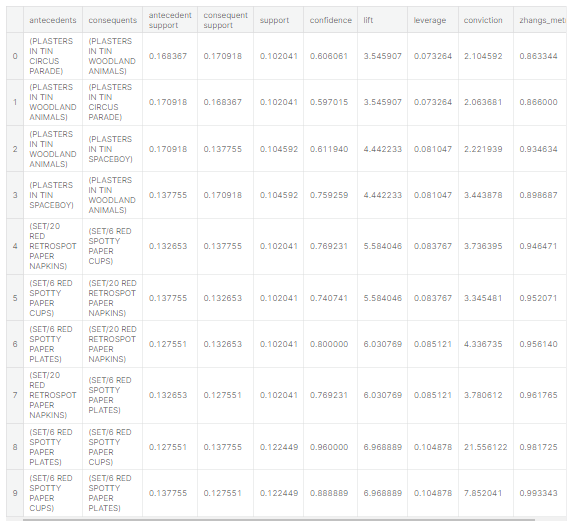


my\_frequent\_itemsets = apriori(my\_basket, min\_support=0.10, use\_colnames=True)

myrules=association\_rules(my\_frequent\_itemsets, metric="lift", min\_threshold=1)

myrules

**TABLE OUTPUT:**



**CONCLUSION:**

Market Basket Analysis is a powerful tool that can be used to gain valuable insights into customer behaviour. By identifying frequent item sets and generating association rules, businesses can better understand what items their customers are likely to purchase together. This information can then be used to improve product placement, create targeted marketing campaigns, and make recommendations for complementary items.

There are a number of different algorithms that can be used to perform Market Basket Analysis, including the Apriori algorithm, the FP-growth algorithm, and the ECLAT algorithm. The Apriori algorithm is one of the most popular algorithms for Market Basket Analysis, and it can be implemented in Python using the apriori function from the mlxtend.frequent\_patterns library.

Once you haven identified frequent item sets, you can then generate association rules. Association rules are statements that express the likelihood of one item being purchased with the purchase of another item. The confidence of an association rule is the probability that if a customer purchases the antecedent (the item on the left side of the rule), they will also purchase the consequent (the item on the right side of the rule). The support of an association rule is the proportion of transactions in the dataset that contain both the antecedent and the consequent.

There are a number of different metrics that can be used to evaluate association rules, such as confidence, support, and lift. Lift is a measure of how much more likely an item is to be purchased when its antecedent is also purchased. A high lift value indicates that the two items are strongly associated with each other.

Market Basket Analysis can be a valuable tool for businesses of all sizes. By understanding what items their customers are likely to purchase together, businesses can better target their marketing campaigns and improve their product placement. This can lead to increased sales and improved customer satisfaction.