**Market Basket Insights**

**Objective**

To identify the consumers, purchase patterns and insights

**Problem Statement**

The problem is to perform market basket analysis on a provided dataset to unveil hidden patterns and associations between products. The goal is to understand customer purchasing behaviour and identify potential cross-selling opportunities for a retail business. This project involves using association analysis techniques, such as Apriori algorithm, to find frequently co-occurring products and generate insights for business optimization.

**Design Thinking Process**

* Data Source:

We can perform market basket analysis on variety of datasets. The dataset will have the information about product purchased by customer.

* Data Pre-processing:

It’s about cleaning and pre-processing the data, and turning it into a format that’s compatible with associative research programs.

* Association analysis:

This includes identifying important associative rules and interpreting insights into consumer purchase behaviour.

* Insight generation:

Once an association rule is created, it is important to scrutinize it to identify the most important rules. The most important rules are the ones that are most likely to provide valuable insights into consumer purchasing behavior.

* Visualize:

We can use use an visualization library to visualize the result.

* Recommendations for business:

Based on the insights gained from the association’s regulatory analysis, recommendations can be made to the retail industry in terms of positioning, promotions and cross-border sales strategies.

**Phases of Development:**

* **Data Collection**
* **Loading and preparing the data**
* **Exploratory data analysis**
* **Performing association analysis**
* **Generating Insights**

**Collection of data:**

**We used the provided dataset from Kaggle:** <https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis>

**Loading and Preparing the DataSet:**

* **Here I’m loading the given dataset using pandas dataframe.**
* **Utilizing read excel function for reading the dataset.**

**import** numpy **as** np *# importing the necessary libraries*

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

**from** matplotlib **import** pyplot **as** plt

df**=**pd**.**read\_excel("C:/Users/avvic/Downloads/Assignment-1-Data.xlsx") #path to file

* Drop any rows where item name column is null.
* Drop any rows where item quantity sold is 0 or less.
* Fill missing customer IDs with a placeholder ID (99999)
* Create a new column, Sumprice, that tells us total sales revenue (Quantity \* Price) of the item

df**.**isnull()**.**sum()

*#Dropping rows where ItemName isn't available*

df**.**dropna(subset**=**["Itemname"],inplace**=True**)

*#Dropping rows where Quantity <=0*

df **=** df[df["Quantity"]**>**0]

df**.**isnull()**.**sum()

*#Fill missing customer IDs*

df['CustomerID']**.**fillna(99999, inplace**=True**)

*#Create SumPrice column*

df["SumPrice"]**=**df["Quantity"]**\***df["Price"]

Exploratory data Analysis:

* Let's find which countries sell the most items, and what items are the most popular in each country.

*#Find the best selling items in each country*

best\_selling\_items **=** df**.**groupby(['Country', 'Itemname'])**.**agg({'Quantity': 'sum'})**.**reset\_index()

best\_selling\_items **=** best\_selling\_items**.**groupby('Country')**.**apply(**lambda** x: x[x['Quantity'] **==** x['Quantity']**.**max()])**.**reset\_index(drop**=True**)

best\_selling\_items**.**sort\_values("Quantity",ascending**=False**)

*#Find the total sales by country.*

total\_sales\_country **=** df**.**groupby(['Country'])**.**agg({'SumPrice': 'sum'})**.**reset\_index()

total\_sales\_country **=** total\_sales\_country**.**sort\_values('SumPrice', ascending**=False**)**.**reset\_index(drop**=True**)

total\_sales\_country

*#Visualizing Total sales by country.*

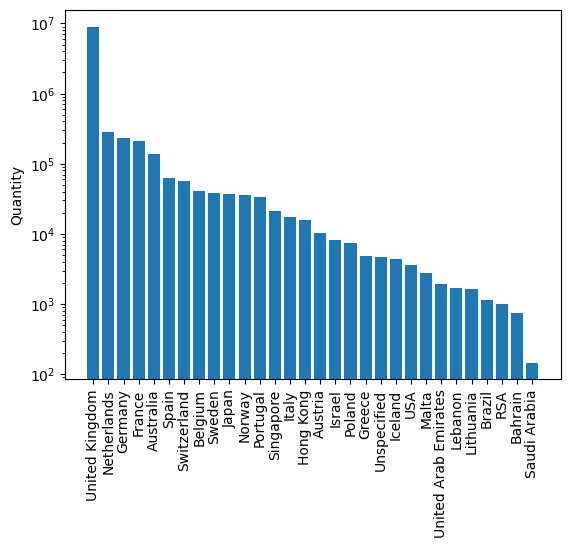
plt**.**bar(total\_sales\_country["Country"],total\_sales\_country["SumPrice"])

plt**.**yscale('log')

plt**.**ylabel('Quantity')

plt**.**xticks(rotation**=**90)

plt**.**show()



* UK has the most amount of sales and the most popular item sold in UK is 'PAPER CRAFT, LITTLE BIRDIE'.
* This outsells the most popular items in other countries by a large magnitude.
* Let's dig in by only looking at UK's grocery store data.

*#Isolate the UK data and let's sort the most popular items in UK by quantity sold.*

only\_uk **=** df[df["Country"]**==**"United Kingdom"]

only\_uk**.**groupby("Itemname")["Quantity"]**.**sum()**.**sort\_values(ascending**=False**)

*#Let's find out what items, across the globe, bring in the most revenue.*

total\_sales\_item **=** df**.**groupby(['Itemname'])**.**agg({'Price': 'mean', 'Quantity': 'sum', 'SumPrice': 'sum'})**.**reset\_index()

*# Create a new column with the count of rows for each group*

total\_sales\_item['Count'] **=** df**.**groupby(['Itemname'])**.**size()**.**values

*# Sort the dataframe by 'SumPrice' column in descending order*

total\_sales\_item **=** total\_sales\_item**.**sort\_values("SumPrice", ascending**=False**)

total\_sales\_item

* We find out that the most sold item globally, 'PAPER CRAFT, LITTLE BIRDIE' was sold in just one transaction. Perhaps this was a large corporate order. If we were to ever do a marketing or promotional push in the future, that required us to analyse our most popular products, this would be an anomaly that we would need to adjust for.

Performing Association Analysis using Apriori Algorithm and Association Rule Mining:

* Convert the Dataset into transactional format (Each row is one bill number with every item sold in that bill in a list)
* Create a one-hot matrix of the products (Product sold = 1, Not sold = 0)
* Merge the transactional matrix and the one hot matrix
* Import the mlxtend library and perform association mining and generate association rules

*#Convert the dataset into transactional format*

transactions **=** df**.**groupby(['BillNo'])['Itemname']**.**apply(list)

transactions

*#Create a one-hot matrix of the products*

one\_hot **=** pd**.**get\_dummies(df['Itemname'])

one\_hot

*#Add the BillNo column back to the one-hot encoded matrix*

one\_hot['BillNo']**=**df['BillNo']

one\_hot

*#Now, we group the One-Hot Matrix by BillNo and sum the values*

one\_hot **=** one\_hot**.**groupby('BillNo')**.**sum()

one\_hot

*#Now, we merge the one-hot encoded matrix, with the transactional data*

transaction\_matrix **=** pd**.**merge(transactions, one\_hot, on**=**'BillNo')

transaction\_matrix

*#Now we have to convert the product columns to 0s and 1s. We are converting sum values to binary as number doesn't matter*

transaction\_matrix[one\_hot**.**columns[:**-**1]] **=** (transaction\_matrix[one\_hot**.**columns[:**-**1]] **>=** 1)**.**astype(int)

transaction\_matrix

transaction\_matrix**.**iloc[:, 1:] **=** transaction\_matrix**.**iloc[:, 1:]**.**astype(bool)

*#Perform frequent itemset mining*

frequent\_itemsets **=** apriori(transaction\_matrix**.**iloc[:, 1:], min\_support**=**0.01, use\_colnames**=True**)

frequent\_itemsets

*# generate association rules*

rules **=** association\_rules(frequent\_itemsets, metric**=**"lift", min\_threshold**=**1)

rules

*#Let's see the top 10 rules by lift*

rules**.**sort\_values('lift', ascending**=False**)**.**head(10)

**Visualizing lift and confidence using scatter plot.**

**import** mpld3

*# create scatter plot with x and y as lift and confidence values*

fig, ax **=** plt**.**subplots()

scatter **=** ax**.**scatter(rules['lift'], rules['confidence'], alpha**=**0.5)

*# Define tooltips*

tooltips **=** []

**for** i **in** range(len(rules)):

rule **=** rules**.**iloc[i]

tooltip **=** f"Rule: {rule['antecedents']} -> {rule['consequents']}\nSupport: {rule['support']:.3f}\nConfidence: {rule['confidence']:.3f}\nLift: {rule['lift']:.3f}"

tooltips**.**append(tooltip)

*# Add tooltips to scatter plot using mpld3*

mpld3**.**plugins**.**connect(fig, mpld3**.**plugins**.**PointHTMLTooltip(scatter, tooltips))

*# Set axis labels and title*

ax**.**set\_xlabel("Lift")

ax**.**set\_ylabel("Confidence")

ax**.**set\_title("Association Rules Scatter Plot")

*# Show the plot*

mpld3**.**display()

rules[(rules['lift'] **>** 40) **&** (rules['lift'] **<** 50)]

**Insights Produced(Recommendation for business):**

* UK has the most amount of sales and the most popular item sold in UK is 'PAPER CRAFT, LITTLE BIRDIE'. By digging into uk’s data We find out that the most sold item globally, 'PAPER CRAFT, LITTLE BIRDIE' was sold in just one transaction. Perhaps this was a large corporate order. If we were to ever do a marketing or promotional push in the future, that required us to analyse our most popular products, this would be an anomaly that we would need to adjust for.
* **The most purchased item is PAPER CRAFT, LITTLE BIRDIE.**
* **The most frequently purchased item is WHITE HANGING HEART T-LIGHT HOLDER.**
* **The best combination items are PINK REGENCY TEACUP AND SAUCER and GREEN REGENCY TEACUP AND SAUCER.**

**Libraries used**

* **Numpy**
* **Pandas**
* **Pyplot from matplotlib**
* **mpld3**