DATA ANALYSIS

DS3114

PROJECT TITLE:

Market Basket Analysis (Supermarket Dataset)

COURSE REPRESENTS :

(DR.Omaima A. Fallatah)

SUBMITTED BY:

Name	ID
Lamar Waleed Fattah	444006719
Rakha Matuq Nooh	444001287

COLLEGE OF COMPUTING UMM AL-QURA UNIVERSITY

Table of content

1. Introduction

The retailer wants to target customers with suggestions on the itemset that a customer is most likely to purchase . I was given a dataset containing data of a retailer; the transaction data provides data around all the transactions that have happened over a period of time. Retailers will use results to grow in his industry and provide for customer suggestions on itemset, we will be able to increase customer engagement and improve customer experience and identify customer behaviour. I will solve this problem by using Association Rules, a type of unsupervised learning technique that checks for the dependency of one data item on another data item.

1.1 Data Description

BillNo: 6-digit number assigned to each transaction. Nominal

Itemname: Product name. Nominal

Quantity: The quantities of each product per transaction. Numeri

Date: The day and time when each transaction was generated. Numeric

Price: Product price. Numeric

CustomerID: 5-digit number assigned to each customer. Nominal

Country: Name of the country where each customer resides. Nominal

1.2 Objectives

Detection Of Common Patterns: Identify products that are bought together frequently, which helps to understand how consumers interact with different products.

Item Frequency Analysis: Understand the most popular and frequently purchased products.

Application of the Apriori algorithm: Extracting groups of frequent items and discovering the rules associated between them.

Extracting Actionable Insights: Providing recommendations based on the analysis results to improve marketing and sales strategies.

Extracting Actionable Insights: Providing recommendations based on the analysis results to improve marketing and sales strategies

2. Data Exploration

```
[174]: import numpy as np
       import pandas as pd
       import matplotlib.pyplot as plt
       import seaborn as sns
       import plotly.express as px
       import plotly.graph_objects as go
       from plotly.subplots import make_subplots  # Importing make_subplots

       from mlxtend.frequent_patterns import apriori,association_rules


       import warnings
       warnings.filterwarnings("ignore")
```

The important libaries

The dataset has a shape of (522,064, 7) and we had problems with duplicates and missing values. To prepare it for Basket Analysis, we made these adjustments:

1. **Removed Duplicates**: We deleted repeated entries.

2. **Fixed Missing Values:** We filled or removed rows with missing data.

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

	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

This is the data before adjustments

Before (missing)

```
df.isnull().sum()
```

```
BillNo             0
Itemname        1455
Quantity           0
Date               0
Price              0
CustomerID    134041
Country            0
dtype: int64
```

After (missing)

```
df.isnull().sum()
```

```
BillNo        0
Itemname      0
Quantity      0
Date          0
Price         0
CustomerID    0
Country       0
dtype: int64
```

duplicate removal

```
df.duplicated().sum()
```

```
5210
```

```
df.drop_duplicates(inplace=True)
df.duplicated().sum()
```

```
0
```

3. Converted Data Types: Changed numbers from float to integer.

4. Formatted Dates: All dates are in the same format.

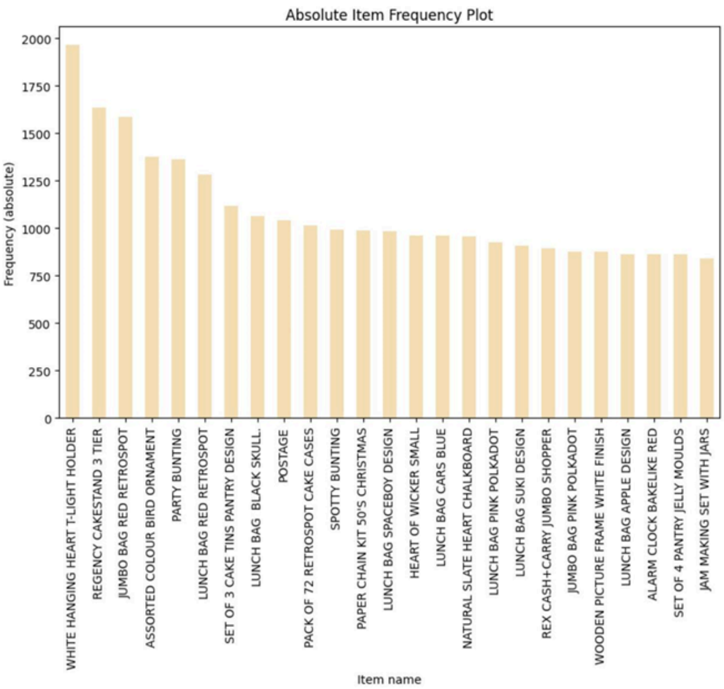
5. Cleaned Strings: Removed extra spaces and standardised text.

* These steps ensured the data is clean and ready for analysis.

```
df.head()
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom

This is the data after adjustments



We identified the most frequent items in the dataset and selected the top 25. The top three most popular items were:

1. White Hanging Heart T-Light Holder
2. Regency Cake Stand
3. Jumbo Bag Red Retrospot

These items were purchased the most frequently.

3. Data Preprocessing

We grouped the data by **bill number** and **item name**, and then summed up the **quantity** for each group.Next, we used `unstack` to rearrange the data, making the item names into columns for better readability. After that, we filled any missing values (NA) to handle items that weren't purchased in some bills.

Finally, we converted the quantities into **True** or **False** values, where **True** means the item was bought and **False** means it wasn't using **Boolean**.

```
def to_bool(x):
    if x <= 0:
        return False
    if x >= 1:
        return True
```

Itemname	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 DAISY PEGS IN WOOD BOX	12 EGG HOUSE PAINTED WOOD	12 HANGING EGGS HAND PAINTED	12 IVORY ROSE PEG PLACE SETTINGS	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	...	ZINC STAR T-LIGHT HOLDER	ZIN SWEETHEAR SOAP DISI
BillNo													
536365	False	False	False	False	False	False	False	False	False	False	...	False	Fals
536366	False	False	False	False	False	False	False	False	False	False	...	False	Fals
536367	False	False	False	False	False	False	False	False	False	False	...	False	Fals
536368	False	False	False	False	False	False	False	False	False	False	...	False	Fals
536369	False	False	False	False	False	False	False	False	False	False	...	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	.
581583	False	False	False	False	False	False	False	False	False	False	...	False	Fals
581584	False	False	False	False	False	False	False	False	False	False	...	False	Fals
581585	False	False	False	False	False	False	False	False	False	False	...	False	Fals
581586	False	False	False	False	False	False	False	False	False	False	...	False	Fals
581587	False	False	False	False	False	False	False	False	False	False	...	False	Fals

18163 rows × 3846 columns

We use an algorithm to find products that are often bought together. It looks for combinations of products that appear in at least 2% of all transactions, which is called the **minimum support**. The result is a list of these frequent product combinations and how often they are bought together.

	support	itemsets
0	0.021527	(3 STRIPEY MICE FELTCRAFT)
1	0.039256	(6 RIBBONS RUSTIC CHARM)
2	0.024666	(60 CAKE CASES VINTAGE CHRISTMAS)
3	0.034686	(60 TEATIME FAIRY CAKE CASES)
4	0.026427	(72 SWEETHEART FAIRY CAKE CASES)
...	...	...
235	0.020922	(SPOTTY BUNTING, PARTY BUNTING)
236	0.022408	(PINK REGENCY TEACUP AND SAUCER, ROSES REGENCY...
237	0.024170	(WHITE HANGING HEART T-LIGHT HOLDER, RED HANGI...
238	0.021307	(REGENCY CAKESTAND 3 TIER, ROSES REGENCY TEACU...
239	0.025657	(WOODEN FRAME ANTIQUE WHITE, WOODEN PICTURE FR...

240 rows × 2 columns

4. Apriori algorithm Implementation

We calculate **association rules** for the frequent product combinations using a metric called **lift**.

Lift shows how much more likely two products are bought together compared to being bought individually. A lift greater than 1 indicates a strong relationship between the products.

The results are then sorted to show the rules with the highest lift, which means the strongest associations between product pairs.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
9	(GREEN REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.035952	0.028960	0.023785	0.661562	22.844013	0.022743	2.869182	0.991885
8	(PINK REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.028960	0.035952	0.023785	0.821293	22.844013	0.022743	5.394565	0.984743
10	(ROSES REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.040412	0.035952	0.027859	0.689373	19.174712	0.026406	3.103557	0.987765
11	(GREEN REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.035952	0.040412	0.027859	0.774885	19.174712	0.026406	4.262660	0.983196
60	(PINK REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.028960	0.040412	0.022408	0.773764	19.146976	0.021238	4.241541	0.976038
...	...	...	...	...	...	...	...	...	...	...
18	(JUMBO BAG RED RETROSPOT)	(JUMBO SHOPPER VINTAGE RED PAISLEY)	0.086605	0.043055	0.021582	0.249205	5.788129	0.017854	1.274577	0.905668
58	(PARTY BUNTING)	(SPOTTY BUNTING)	0.074437	0.053956	0.020922	0.281065	5.209169	0.016905	1.315897	0.873015
59	(SPOTTY BUNTING)	(PARTY BUNTING)	0.053956	0.074437	0.020922	0.387755	5.209169	0.016905	1.511753	0.854115
23	(LUNCH BAG RED RETROSPOT)	(JUMBO BAG RED RETROSPOT)	0.069702	0.086605	0.023069	0.330964	3.821547	0.017032	1.365240	0.793645
22	(JUMBO BAG RED RETROSPOT)	(LUNCH BAG RED RETROSPOT)	0.086605	0.069702	0.023069	0.266370	3.821547	0.017032	1.268075	0.808331

68 rows × 10 columns

There is a relationship within lift 22.8

4. Conclusion

In conclusion, the **association rule analysis** using the **lift metric** demonstrates that certain products have a significant relationship with each other when their lift value exceeds one. A lift value greater than one indicates that the products are more likely to be purchased together than independently. The highest lift observed was 22.8, indicating a particularly strong association between those product pairs.

5. Challenges

The data analysis process presented several challenges. Initially, we attempted to use Google Colab to analyse the dataset, but we encountered repeated crashes due to the size or complexity of the data. As a result, we switched to using Kaggle for the analysis. Kaggle's platform handled the data efficiently, offering a much smoother experience, and performed exceptionally well throughout the process. The transition to Kaggle allowed us to carry out the analysis without further disruptions, ultimately leading to successful completion of the project.