PURCHASE PATTERN ANALYTICS

27 NOVEMBER 2024

TEAM MEMBERS

TEAM ID PTID-CDA-OCT-24-217

PPROJECT ID CDACL005

NIMMALA BHARATH GOUD

ARYAN KUNDHARA

ASAOLU OYINLOLAJOY

ASAOLU OYINLOLAJOY

PROBLEM STATEMENT

- In the context of a retail environment, the challenge is to leverage Market Basket Analysis (MBA) on a Transaction dataset to unveil relationships between products. By conducting exploratory data analysis, addressing data quality issues, employing data visualization techniques, and implementing the Apriori algorithm, the objective is to extract valuable insights. This includes understanding customer purchasing behavior, identifying frequently co-occurring products, and deriving actionable recommendations to enhance sales strategies and customer satisfaction.

TASKS ASSIGEND

- - Acquire and load the Transaction dataset.
- - Begin Exploratory Data Analysis (EDA):
- - Identify key statistics (mean, median, mode, etc.).
- - Visualize data distributions and patterns.
- - Detect any outliers or missing values.

Methodology

Data Collection:

- Steps:
- Establish connection to the database:
- Host: 18.136.157.135
- User: dm_team16
- Password: 2o_hihiFeTRE
- Import the dataset into a suitable analytical tool (Python, R, or SQL-based environment).
- Verify the structure and types of data fields.

Data Cleaning and Transformation

- Handle missing values:
- Impute with statistical methods (mean, median) or remove rows/columns based on relevance.
- Correct outliers or erroneous entries.
- Convert data into transactional format for Market Basket Analysis:
- Example: From tabular format to itemset lists.

Advanced Analysis with Apriori Algorithm

- Create visualizations showing trends and patterns in the data.
- Implement the Apriori algorithm to identify:
- Frequent itemsets.
- Association rules (e.g., confidence, lift)

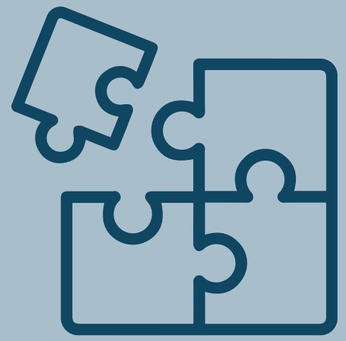
Tools and Techniques

- Libraries: Pandas, NumPy, Matplotlib, Seaborn, and MLxtend (for Apriori).
- Algorithm: Apriori for MBA.
- Visualization Platforms: Tableau or Power BI for advanced graphs.

PROJECT OBJECTIVES

Analysis Phase

- We delve into exploring our dataset to gain insights and identify patterns. We begin by examining the structure of our data using the .info() method, which provides a summary of the data types and non-null counts for each column.



Strategy Development

- It's essential to ensure that our dataset is in the right format and free of inconsistencies. In this chapter, we'll walk through the steps taken to prepare our data for analysis. This includes converting date columns, cleaning up numeric formats, checking for missing values, and generating descriptive statistics

Implementation Plan

Importing libraries
Reading our data
Data preparation
Data Filtering and Cleaning
Total Sales Analysis
Unique Item Analysis
Market Basket Analysis
Association Rule Mining
Confidence
Lift Analysis
Analysis of Association Rule Metrics

1| Importing libraries

```
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
```

READ OUR DATA

2| Reading our data

```
df = pd.read_csv("purchase patteren.csv")
df.head()
```

	BillNo	Itemname	Quantity	Present_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

```
#Get Summary
print(df.shape)
print(df.info())
```

```
(516778, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 516778 entries, 0 to 516777
Data columns (total 7 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   BillNo        516778 non-null  object
 1   Itemname      515323 non-null  object
 2   Quantity      516778 non-null  int64
 3   Present_Date  516778 non-null  object
 4   Price         516778 non-null  float64
 5   CustomerID    382811 non-null  float64
 6   Country       516778 non-null  object
dtypes: float64(2), int64(1), object(4)
memory usage: 27.6+ MB
None
```

DATA PREPARATION

3| Data preparation

Before diving into the analysis, it's essential to ensure that our dataset is in the right format and free of inconsistencies. In this chapter, we'll walk through the steps taken to prepare our data for analysis. This includes converting date columns, cleaning up numeric formats, checking for missing values, and generating descriptive statistics

```
df["Present_Date"] = pd.to_datetime(df["Present_Date"], format='%d.%m.%Y %H:%M')

df["Year/Month"] = df["Present_Date"].dt.to_period("M")
```

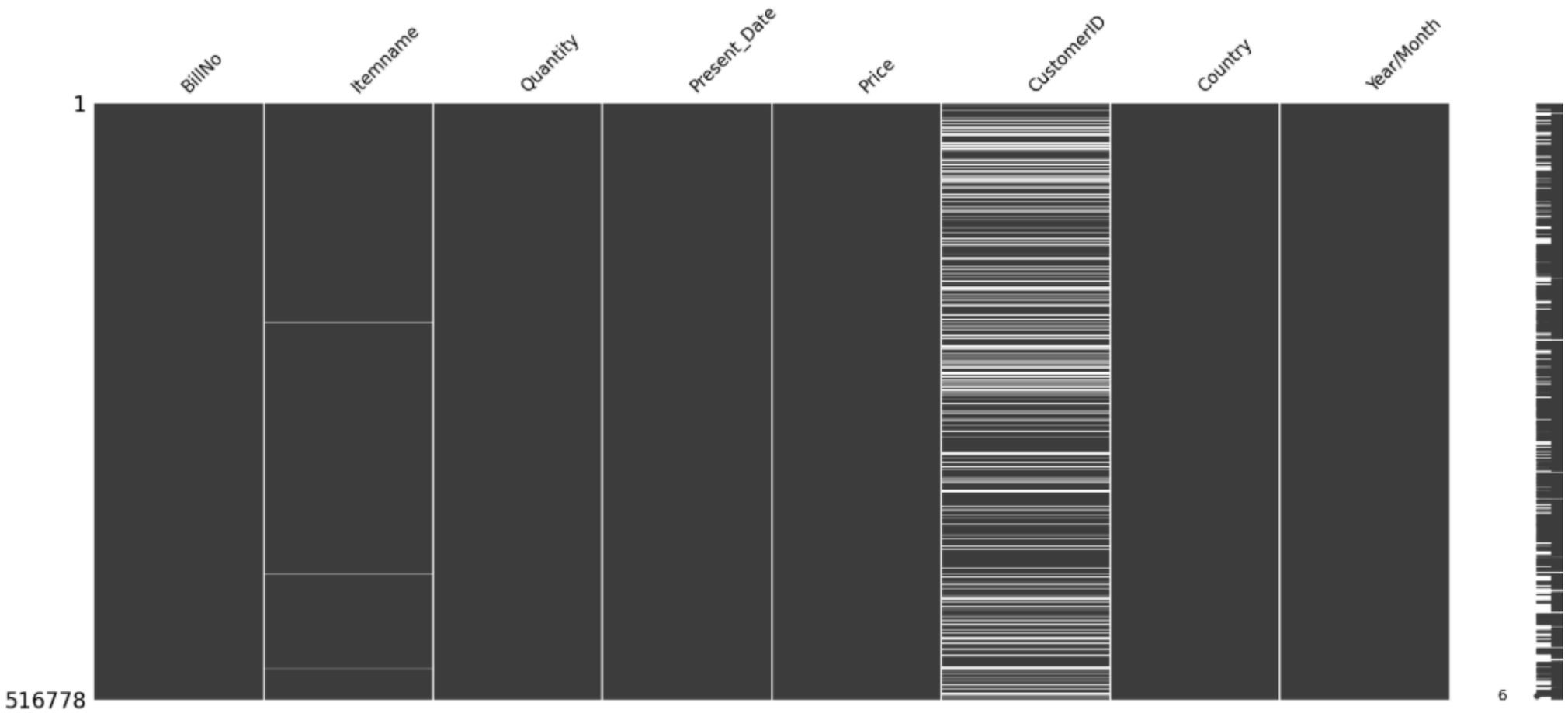
```
missing = df.isnull().sum()

print(missing)
```

```
BillNo          0
Itemname        0
Quantity        0
Present_Date    0
Price           0
CustomerID      0
Country         0
Year/Month      0
TotalPrice      0
dtype: int64
```

DATA PREPARATION

```
import missingno as msno

# Visualize missing values as a matrix
msno.matrix(df)
```

```
<Axes: >
```



KEY STATISTICS

```
#Statistics
df.describe()
```

	Quantity	Present_Date	Price	CustomerID
count	516778.00000	516778	516778.000000	382811.000000
mean	10.16247	2011-07-04 08:18:39.177790464	3.841504	15310.347702
min	-9600.00000	2010-12-01 08:26:00	-11062.060000	12346.000000
25%	1.00000	2011-03-28 09:59:00	1.250000	13928.500000
50%	3.00000	2011-07-19 14:29:00	2.080000	15249.000000
75%	10.00000	2011-10-19 12:11:00	4.130000	16820.000000
max	80995.00000	2011-12-09 12:50:00	13541.330000	18287.000000
std	161.91653	NaN	42.113493	1722.483516

TASKS ASSIGEND

- Data Cleaning and Transformation
- dataset into a suitable format for market basket analysis.
- Data Visualization and Apriori Algorithm Implementation
- Create visualizations to represent product occurrences, associations, and trends
- Implement the Apriori algorithm for market basket analysis
- Identify frequent item sets and association rules
- Visualize the results of the Apriori algorithm

4| Data Filtering and Cleaning

A crucial aspect of data analysis is ensuring the data's integrity by filtering out irrelevant or erroneous entries and handling missing values appropriately. In this chapter, we focus on filtering out non-positive values, removing rows with missing item names, filling in missing customer IDs, and calculating total prices per transaction. These steps are vital to ensure the accuracy and reliability of our subsequent analysis.

```
df = df[(df["Quantity"] > 0) & (df["Price"] > 0)]
```

```
#drop rows without item
df = df[df["Itemname"].notnull()]
```

```
# Filling missing customer IDs
df = df.fillna('#NV')
```

```
df["TotalPrice"] = df["Quantity"] * df["Price"]
```

```
df.head()
```

	BillNo	Itemname	Quantity	Present_Date	Price	CustomerID	Country	Year/Month	TotalPrice
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	2010-12	15.30
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010-12	20.34
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	2010-12	22.00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010-12	20.34
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010-12	20.34

EXPLORATORY DATA ANALYSIS

5| EDA

In this section, we delve into exploring our dataset to gain insights and identify patterns. We begin by examining the structure of our data using the .info() method, which provides a summary of the data types and non-null counts for each column.

3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 514270 entries, 0 to 516777
Data columns (total 9 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   BillNo        514270 non-null  object
 1   Itemname      514270 non-null  object
 2   Quantity      514270 non-null  int64
 3   Present_Date  514270 non-null  datetime64[ns]
 4   Price         514270 non-null  float64
 5   CustomerID    514270 non-null  object
 6   Country       514270 non-null  object
 7   Year/Month    514270 non-null  period[M]
 8   TotalPrice    514270 non-null  float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(4), period[M](1)
memory usage: 39.2+ MB
```

TOTAL SALES ANALYSIS

We group the data by month and year, calculating the total sum of sales to understand the sales trend over time. The resulting visualization depicts the total sales per month.

```
import matplotlib.ticker as ticker
# Grouping the data by month and year, and calculating the total sum of sales
monthly_sales = df.groupby('Year/Month')['TotalPrice'].sum()

plt.figure(figsize=(15,6))
monthly_sales.plot(kind='line', marker='o', color='b')
plt.title('Total Sales per Month')
plt.xlabel('Month')
plt.ylabel('Total Sales')

formatter = ticker.FormatStrFormatter('€%.2f')
plt.gca().yaxis.set_major_formatter(formatter)

plt.grid(True)
plt.show()
```

Sales Performance Analysis

Key Observations

- Downward Start: Sales start declining after December but hit a low point in February 2011.
- Fluctuations: From March to August, sales show minor fluctuations but remain relatively stable.
- Steady Growth: From September onwards, sales demonstrate consistent growth, peaking significantly in November 2011.
- Sharp Drop: Sales decline steeply from November to December

Performance Analysis

- Seasonality: The significant peak in November suggests seasonal trends, possibly influenced by holiday shopping or end-of-year activities.
- Low Months: February experienced the lowest performance, possibly due to post-holiday economic slowdowns.
- Growth Opportunities: The steady growth from September to November highlights a potential marketing or sales campaign effectiveness.

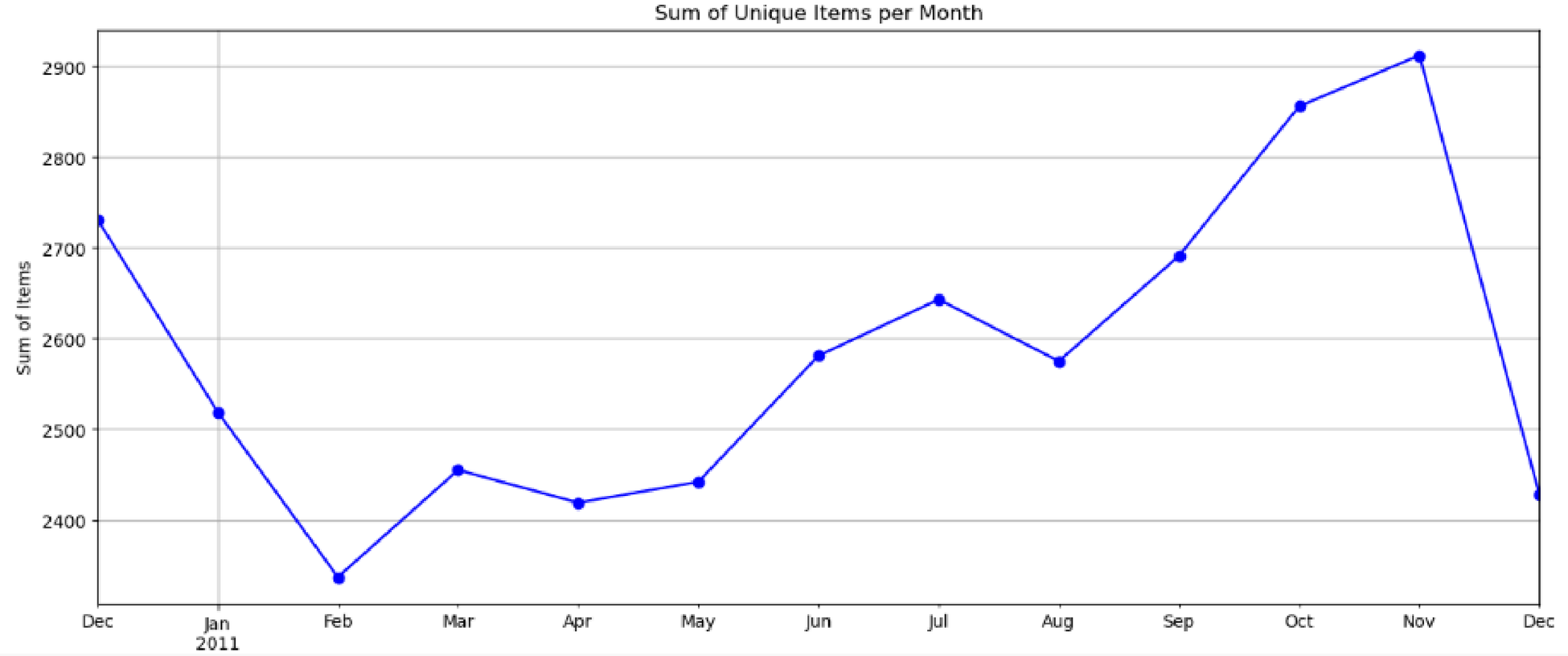
Recommendations

1. Focus on November: Identify strategies that drove the November peak and replicate them in other months.
2. Boost Early-Year Sales: Introduce promotions or campaigns in Q1 (January–March) to counter post-holiday slumps.
3. Monitor Trends: Assess market factors influencing sales stability from March to August for improvement.
4. Prepare for December: Address the sharp decline by aligning marketing strategies with year-end customer behaviors.

Unique Item Analysis

Next, we analyze the uniqueness of items sold per month by grouping the data and calculating the count of unique items. The line plot visualizes the sum of unique items per month.

```
monthly_item = df.groupby("Year/Month")["Itemname"].nunique()

plt.figure(figsize=(15,6))
monthly_item.plot(kind="line",marker="o",color="b")
plt.title("Sum of Unique Items per Month")
plt.xlabel("Month")
plt.ylabel("Sum of Items")
plt.grid(True)
plt.show()
```



Unique Item Analysis

Key Observations:

1. Declining Start (December–February):
   - The sum of unique items sold drops significantly from December to February.
   - This indicates reduced variety in customer purchases, likely due to the post-holiday lull.
2. Stabilization (March–August):
   - From March to August, the count of unique items sold stabilizes, showing minor fluctuations.
   - This suggests a steady but limited demand for unique items during these months.
3. Strong Growth (September–November):
   - A sharp increase in unique item sales is observed from September to November.
   - November achieves the peak for the year, indicating heightened demand for diverse items, likely driven by seasonal factors (e.g., holiday preparations or promotions).
4. Sharp Decline in December:
   - Similar to overall sales trends, the count of unique items drops abruptly in December, potentially due to inventory sell-outs or customer focus shifting to fewer high-demand items.

Performance Insights:

- Seasonal Influences: The sharp growth in unique items from September to November aligns with the holiday shopping period.
- Limited Demand in Early Year: The decline in February suggests that Q1 experiences limited market activity for unique items.
- Stabilized Middle Months: March to August provides a stable baseline, showing a consistent but less dynamic variety in purchases.

Recommendations:

1. Capitalize on Q4 Trends:
   - Boost the availability of unique items during September to November to maximize revenue from increased demand.
   - Strategize targeted campaigns for promoting diverse products during this period.
2. Address Early-Year Slump:
   - Explore promotional strategies or new product launches in Q1 to stimulate interest in unique items.
3. Inventory Optimization:
   - Ensure adequate inventory of diverse products during peak months and avoid overstocking in slower months.
4. Analyze Customer Preferences:
   - Conduct deeper analysis to identify high-performing unique items in September–November and adapt offerings accordingly.

Market Basket Analysis

In this section, we embark on a crucial aspect of retail analytics: Market Basket Analysis. This technique allows us to uncover hidden patterns and associations among items purchased together, providing valuable insights for optimizing product placement, marketing strategies, and inventory management.

6|MBA - Preprocessing

Before diving into analysis, we need to preprocess our transactional data. We select pertinent columns, such as 'BillNo' and 'Itemname', essential for identifying item associations. To facilitate analysis, we encode the items into a binary format using one-hot encoding, creating a basket representation where each row represents a transaction (BillNo) and each column represents an item. This step lays the groundwork for further analysis.

```
# Preprocessing the data
data_processed = df[['BillNo', 'Itemname']]
data_encoded = pd.get_dummies(data_processed, columns=['Itemname'])
data_encoded.columns = data_encoded.columns.str.replace("Itemname_", "")
basket = data_encoded.groupby('BillNo').sum()

basket.head()
```

BillNo	*Boombox Ipod Classic	*USB Office Mirror Ball	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	...	ZINC STAR T-LIGHT HOLDER	ZINC SWEETHEART SOAP DISH	ZINC SWEETHEART WIRE LETTER RACK	Z H
536365	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
536366	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
536367	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
536368	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
536369	0	0	0	0	0	0	0	0	0	0	...	0	0	0	

5 rows × 4006 columns

To simplify the analysis and focus solely on item presence, we convert the occurrence values to binary format. This binary transformation facilitates the identification of item associations, regardless of the quantity purchased in each transaction.

```
basket[basket > 0] = 1
```

Applying Apriori Algorithm

Generating Frequent Itemsets Using the Apriori algorithm, we aim to extract frequent itemsets – combinations of items that frequently co-occur in transactions. These itemsets provide valuable insights into customer purchasing behaviors, revealing commonly associated products.

```
frequent_itemsets = apriori(basket,min_support=0.02,use_colnames=True)
frequent_itemsets.head()
```

	support	itemsets
0	0.023364	(3 STRIPEY MICE FELTCRAFT)
1	0.023671	(4 TRADITIONAL SPINNING TOPS)
2	0.048160	(6 RIBBONS RUSTIC CHARM)
3	0.021319	(60 CAKE CASES DOLLY GIRL DESIGN)
4	0.030675	(60 CAKE CASES VINTAGE CHRISTMAS)

Association Rule Mining

In this phase, we employ association rule mining techniques to extract meaningful patterns and relationships between items in our dataset. By analyzing these rules, we can uncover actionable insights that drive business decisions, such as product placement strategies and targeted marketing campaigns.

CONFIDENCE

Confidence represents the likelihood that the consequent item (item bought after) is purchased given the antecedent item(s) (item bought before).
A confidence of 0.6 implies that the consequent item is purchased in 60% of transactions where the antecedent item(s) are also present, indicating a strong positive relationship.

```
RULES_CONFIDENZ = ASSOCIATION_RULES(FREQUENT_ITEMSETS, METRIC="CONFIDENCE", MIN_THRESHOLD=0.8, NUM_ITEMSETS= 10000000)
PRINT(RULES_CONFIDENZ.SHAPE)
RULES_C = RULES_CONFIDENZ.ROUND(3)
RULES_C.HEAD()
```

```
(16, 14)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
0	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.049	0.053	0.032	0.655	12.467	1.0	0.030	2.743	0.968	0.465	0.635	0.635
1	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.053	0.049	0.032	0.616	12.467	1.0	0.030	2.478	0.971	0.465	0.596	0.635
2	(PINK REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.038	0.050	0.031	0.822	16.520	1.0	0.029	5.332	0.976	0.547	0.812	0.721
3	(GREEN REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.050	0.038	0.031	0.621	16.520	1.0	0.029	2.538	0.989	0.547	0.606	0.721

LIFT ANALYSIS

- Lift measures how much more likely the consequent item(s) are purchased when the antecedent item(s) are present compared to when they are not.
- A lift value of 1.0 indicates that the items in the consequent are bought together as often as would be expected by chance.
- rules_lift = association_rules(frequent_itemsets, metric="lift", min_threshold= 2.5, num_itemsets=100000000)
- print(rules_lift.shape)
- rules_l = rules_lift.round(3)
- rules_l.head()

```
(174, 14)
```

:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
0	(60 TEATIME FAIRY CAKE CASES)	(PACK OF 72 RETROSPOT CAKE CASES)	0.041	0.065	0.022	0.545	8.337	1.0	0.020	2.055	0.917	0.265	0.513	0.443
1	(PACK OF 72 RETROSPOT CAKE CASES)	(60 TEATIME FAIRY CAKE CASES)	0.065	0.041	0.022	0.340	8.337	1.0	0.020	1.454	0.942	0.265	0.312	0.443
2	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE PINK)	0.049	0.039	0.021	0.425	10.953	1.0	0.019	1.672	0.956	0.313	0.402	0.483
3	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE GREEN)	0.039	0.049	0.021	0.542	10.953	1.0	0.019	2.073	0.945	0.313	0.518	0.483

SUPPORT ANALYSIS

- Support quantifies the frequency with which a rule occurs in the dataset.
- A support of 0.05 means that the rule occurs in at least 5% of transactions, signifying its significance in the dataset.
- rules_support = association_rules(frequent_itemsets, metric="support", min_threshold=0.03,num_itemsets=100000000)
- print(rules_support.shape)
- rules_s = rules_support.round(3)
- rules_s.head()

```
(16, 14)
```

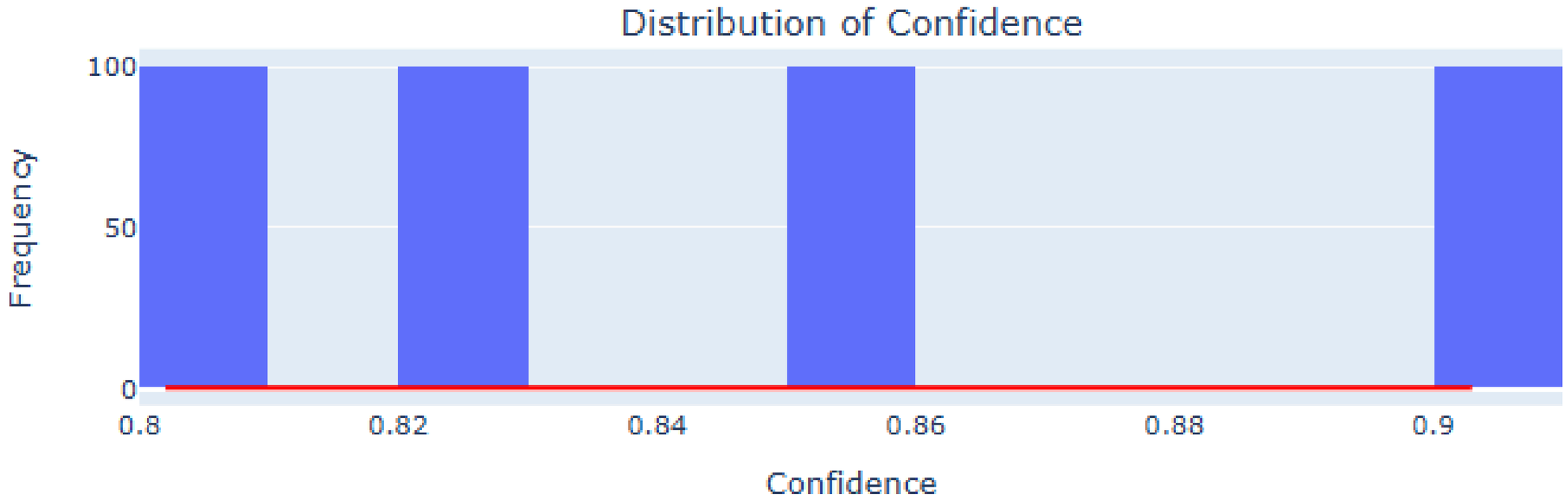
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
0	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.049	0.053	0.032	0.655	12.467	1.0	0.030	2.743	0.968	0.465	0.635	0.635
1	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.053	0.049	0.032	0.616	12.467	1.0	0.030	2.478	0.971	0.465	0.596	0.635
2	(PINK REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.038	0.050	0.031	0.822	16.520	1.0	0.029	5.332	0.976	0.547	0.812	0.721
3	(GREEN REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.050	0.038	0.031	0.621	16.520	1.0	0.029	2.538	0.989	0.547	0.606	0.721

Analysis of Association Rule Metrics

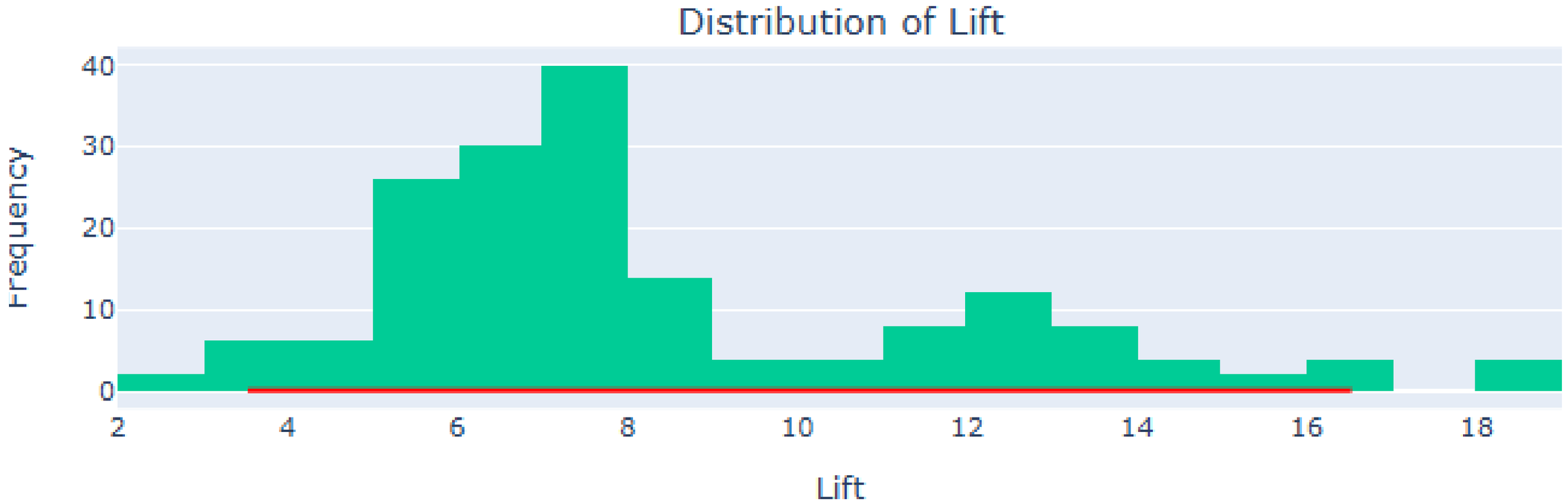
- In this section, we visually inspect the distribution of key metrics associated with our association rules, namely Confidence, Lift, and Support. These histograms provide insights into the frequency and distribution of these metrics across the extracted association rules.

```
# Create subplots in Plotly
fig = make_subplots(rows=3, cols=1, subplot_titles=('Distribution of Confidence',
                                                    'Distribution of Lift',
                                                    'Distribution of Support'))

# Confidence Histogram
fig.add_trace(
    go.Histogram(x=rules_c['confidence'], nbinsx=20, histnorm='density', name='Confidence'),
    row=1, col=1
)
fig.add_trace(
    go.Scatter(x=rules_c['confidence'], y=pd.Series(rules_c['confidence']).value_counts(normalize=True).sort_index(),
               mode='lines', line=dict(color='red'), name='KDE'),
    row=1, col=1
)

# Lift Histogram
fig.add_trace(
    go.Histogram(x=rules_l['lift'], nbinsx=20, histnorm='density', name='Lift'),
    row=2, col=1
)
fig.add_trace(
    go.Scatter(x=rules_l['lift'], y=pd.Series(rules_l['lift']).value_counts(normalize=True).sort_index(),
               mode='lines', line=dict(color='red'), name='KDE'),
    row=2, col=1
)
```

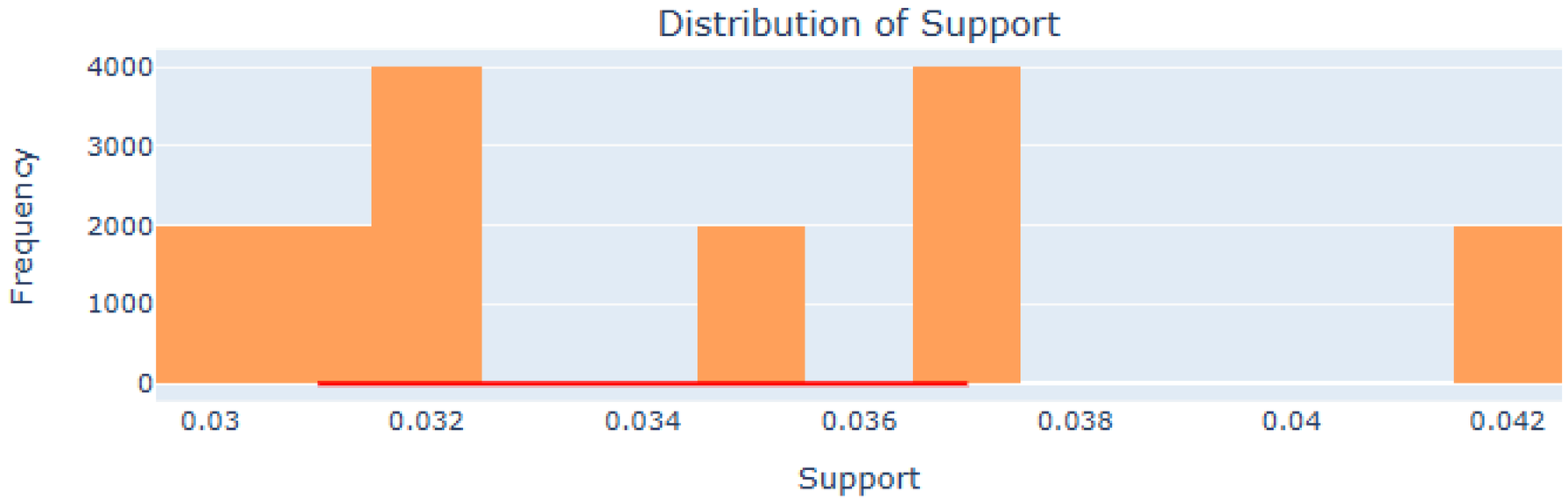
```
# Support Histogram
fig.add_trace(
    go.Histogram(x=rules_s['support'], nbinsx=20, histnorm='density', name='Support'),
    row=3, col=1
)
fig.add_trace(
    go.Scatter(x=rules_s['support'], y=pd.Series(rules_s['support']).value_counts(normalize=True).sort_index(),
               mode='lines', line=dict(color='red'), name='KDE'),
    row=3, col=1
)

# Update layout
fig.update_layout(height=900, width=800, title_text="Distribution of Confidence, Lift, and Support",
                  showlegend=False)

# Update axes titles
fig.update_xaxes(title_text="Confidence", row=1, col=1)
fig.update_yaxes(title_text="Frequency", row=1, col=1)
fig.update_xaxes(title_text="Lift", row=2, col=1)
fig.update_yaxes(title_text="Frequency", row=2, col=1)
fig.update_xaxes(title_text="Support", row=3, col=1)
fig.update_yaxes(title_text="Frequency", row=3, col=1)

fig.show()
```



- •Strong Association between Regency Teaware: There's a high confidence level (0.826 to 0.905) between different combinations of Regency teacups and saucers. This indicates that customers who purchase one item from this category are highly likely to purchase another.
- •Regency Teaware and Roses: There's a strong association between Regency teaware and roses. This suggests that customers who purchase Regency teaware are likely to also purchase roses.
- •Jumbo Storage Bag and Red Retrospot: There's a high confidence level (0.802) between the Jumbo Storage Bag and the Red Retrospot. This indicates a strong association between these two items.
- •Actionable Insights:
- •Cross-Selling Opportunities: Use these insights to cross-sell related products to customers. For example, if a customer purchases a green Regency teacup, recommend a pink Regency saucer.
- •Product Bundling: Consider bundling complementary products, such as the green and red Bakelite alarm clocks or the different Regency teaware items.
- •Targeted Marketing: Use these insights to create targeted marketing campaigns for specific customer segments. For example, you could target customers who have purchased Regency teaware with promotions for roses.
- •Inventory Management: Ensure adequate stock levels for frequently purchased items to avoid stockouts.



- •Strong Association between Alarm Clocks: There's a strong association between green and red Bakelite alarm clocks. This suggests a potential for bundling these items or cross-selling to customers who purchase one.
- •Regency Teaware Affinity: Green and pink Regency teacups and saucers are often purchased together, indicating a strong customer preference for these color combinations.
- •Product Recommendations: Leverage these insights to recommend complementary products to customers based on their purchase history. For instance, if a customer buys a green Bakelite alarm clock, recommend the red one as well.
- •Targeted Marketing Campaigns: Create targeted marketing campaigns for specific customer segments based on their purchasing behavior. For example, you could target customers who have purchased green Regency teacups with promotions for pink saucers.
- •Inventory Management: Optimize inventory levels to ensure sufficient stock of frequently purchased items.
-



- Popular Products: The higher the support value, the more frequently the itemset is purchased. For example, the itemset of "Green Regency Teacup and Saucer" and "Pink Regency Teacup and Saucer" has a relatively high support value, indicating that these items are frequently purchased together.
- inventory Management: Understanding the support of different itemsets can help optimize inventory levels. Items with high support values should be stocked in sufficient quantities to avoid stockouts.
- Product Bundling: Items with high support values can be bundled together to create attractive offers for customers. For example, you could bundle the green and pink Regency teacups and saucers.

Data Analysis

- Quantity Overview:
- Mean: The average quantity per transaction is approximately 10 items, but the data shows extreme outliers:
  - Minimum: -9600, which is likely an error (negative quantities are unusual unless representing returns or errors).
  - Maximum: 80995, which is an extremely high value and may need validation.
- Price Analysis:
- Mean: The average price is 3.84, but the data includes negative and abnormally high values:
  - Minimum: -11062.06, which may indicate incorrect pricing or refund data.
  - Maximum: 13541.33, which might represent luxury products or incorrect entries.
- The standard deviation (42.11) indicates a wide variation in product prices.
- CustomerID Data:
- **Around 134,000 CustomerID values are missing (516778 - 382811),** which might complicate customer-level insights.
- Date Range:
- The transactions span from 2010-12-01 to 2011-12-09, covering about one year

INSIGHTS

1. Seasonal Trends
- Peak in November: Indicates strong seasonal demand, likely tied to holidays or end-of-year activities.
- Low Performance in Q1: February records the lowest sales, often due to post-holiday slowdowns.
- Growth Opportunities: Steady growth from September to November signals effective campaigns or seasonal product relevance.

2. Unique Item Patterns
- High Variety in Q4: The demand for diverse items peaks from September to November, showing customer interest in unique offerings.
- Stable Middle Period: March to August reflects consistent but less dynamic demand.

3. Association Rule Analysis
- Frequent Itemsets: Items like Regency Teacups and Roses, and Green/Red Bakelite Alarm Clocks, show strong associations.
- Cross-Selling Potential: High confidence between certain product pairs, e.g., Jumbo Storage Bags and Red Retrospot.

4. Inventory Challenges
- Outliers in Data: Extreme values in quantity (-9600 and 80995) and price (-11062.06 and 13541.33) suggest data errors or unique events, emphasizing the need for validation.
- Stocking Needs: Items with high support values (frequent purchase) require robust inventory management to prevent stockouts.

Recommendations

1. Strategic Campaigns
- Q1 Promotions: Implement early-year discounts or introduce new products to counteract the post-holiday slump.
- Boost November Sales: Replicate successful strategies from November during other potential peak months.
- Targeted Marketing: Focus campaigns on customer segments purchasing high-support items.

2. Product Bundling
- Bundle highly associated items (e.g., Green and Pink Regency Teacups) to encourage larger transactions.
- Introduce holiday-themed bundles in Q4 to capitalize on heightened demand.

3. Inventory Management
- Maintain sufficient stock for high-support items like Green Regency Teacup & Saucer to avoid stockouts.
- Use trends from Market Basket Analysis to align inventory with expected demand.

4. Data Validation
- Address data anomalies (e.g., extreme negative quantities and prices) to improve analysis reliability.
- Enrich the dataset with CustomerID details to enhance customer segmentation and personalization.

5. Visualization and Monitoring
- to monitor sales trends, product performance, and inventory levels in real time.
- Regularly review metrics such as lift, confidence, and support to refine marketing and inventory strategies.

6. Seasonal Product Insights
- Increase the availability of diverse products during September to November to capture heightened demand.
- Analyze low-performing months to identify any overlooked opportunities or necessary adjustments.

CONCLUSION

- The Market Basket Analysis (MBA) provided valuable insights into customer purchasing patterns and product relationships within the transaction dataset. The findings reveal significant seasonal trends, including peaks in November and slumps in February, emphasizing the importance of timing in sales strategies. The association rule analysis identified strong relationships between specific products, presenting opportunities for cross-selling, product bundling, and targeted marketing.
- Data anomalies, such as extreme outliers in quantity and price, highlight the necessity for robust data validation and cleaning processes to ensure accurate analysis. Effective inventory management, informed by high-support itemsets, is critical to meeting customer demand and avoiding stockouts.
- By leveraging these insights, the business can implement strategic campaigns, optimize inventory, and enhance customer satisfaction, ultimately driving revenue growth and building stronger customer relationships. To maximize impact, it is essential to continually monitor performance, adapt strategies based on trends, and refine operations to align with evolving market dynamics.


Thank you