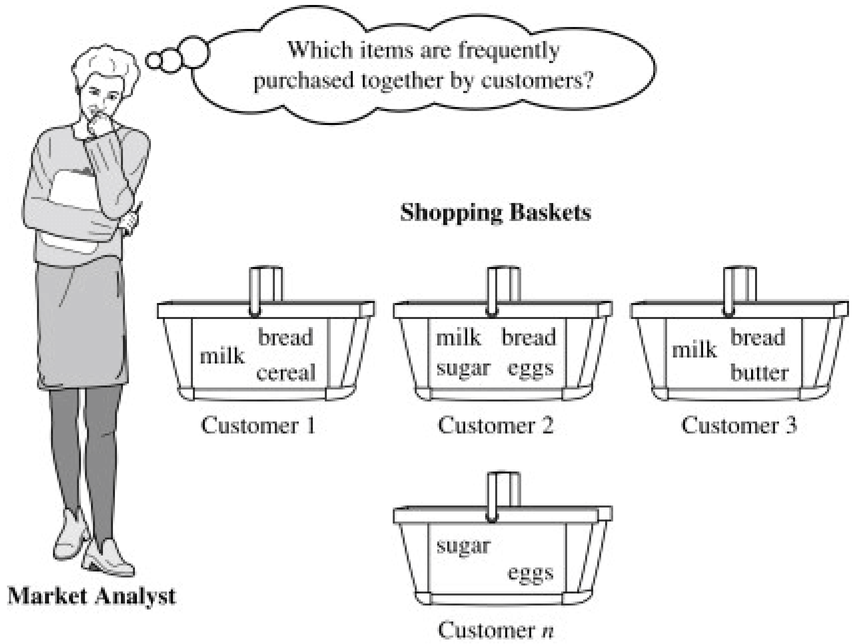
MARKET BASKET INSIGHTS

Phase 3: Development Part 1

Introduction:

Market basket analysis is a strategic data mining technique used by retailers to enhance sales by gaining a deeper understanding of customer purchasing patterns. This method entails the examination of substantial datasets, such as historical purchase records, in order to unveil inherent product groupings and identify items that tend to be bought together. By recognizing these patterns of co-occurrence, retailers can make informed decisions to optimize inventory management, devise effective marketing strategies, employ cross-selling tactics, and even refine store layout for improved customer engagement.



Association Rule Mining is primarily used when you want to identify an association between different items in a set and then find frequent patterns in a transactional database or relational database.

1. Data Collection: Make sure you have the transaction data in a suitable format. This could be a CSV, Excel, or any other structured data format.

2. Load the Dataset: Use a programming language like Python and libraries like pandas to load the dataset into your working environment.

3. Data Preprocessing: Data preprocessing is the concept of changing the raw data into a clean data set.

a. Data Cleaning:

Remove any duplicates or irrelevant columns. Handle missing values if there are any.

b. Data Transformation:

Convert the data into a suitable format for association analysis. This often means creating a binary matrix where rows represent transactions, columns represent items, and the values are binary indicators of whether an item was part of a transaction.

4. Association Analysis: It's a process of searching for hidden association or pattern in a large dataset.

a. Transaction Encoding:

Transform your dataset into a transaction format, where each row represents a transaction, and items purchased are listed.

b. Apriori Algorithm:

Use a frequent itemset mining algorithm like Apriori to identify itemsets that frequently appear together.

c. Association Rules:

From the frequent itemsets, generate association rules with support and confidence measures.

5. Filtering and Interpretation: Set a minimum support and confidence threshold to filter the rules, making them more actionable and relevant.

6. Results Visualization: Visualize the association rules and insights using tools like matplotlib or specialized association rule visualization libraries.

7. Application in Business:

a. Product Recommendations:

Use the insights to recommend-related products to customer ,increasing cross-selling and upselling opportunities.

b. Inventory Management:

Optimize stock levels and placement basedon item assosiations to reduce storage costs and stockouts.

c. Marketing Strategies:

Tailor marketing campaigns to promote items frequently purchased together, improving sales.

d. Store Layout:

Arrange products in a way that encourages customer to buy and complementary items.

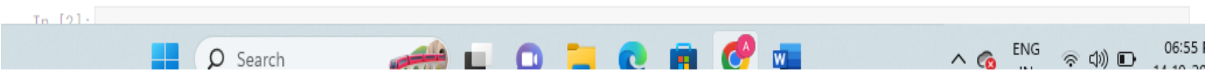
Program:

```
import pandas as pd

from mlxtend.frequent_patterns import apriori, association_rules

pd.set_option('display.max_columns', None)

pd.set_option('display.max_rows', None)

pd.set_option('display.width', 500)

df = pd.read_excel("/kaggle/input/market-basket-analysis/Assignment-
1_Data.xlsx")

def outlier_thresholds(dataframe, variable):

    quartile1 = dataframe[variable].quantile(0.01)

    quartile3 = dataframe[variable].quantile(0.99)

    interquantile_range = quartile3 - quartile1

    up_limit = quartile3 + 1.5 * interquantile_range

    low_limit = quartile1 - 1.5 * interquantile_range

    return low_limit, up_limit

def replace_with_thresholds(dataframe, variable):

low_limit, up_limit = outlier_thresholds(dataframe, variable)

    dataframe.loc[(dataframe[variable] < low_limit), variable] = low_limit

    dataframe.loc[(dataframe[variable] > up_limit), variable] = up_limit

def retail_data_prep(dataframe):

    dataframe = dataframe[dataframe["Quantity"] > 0]

    dataframe = dataframe[dataframe["Price"] > 0]

    replace_with_thresholds(dataframe, "Quantity")

    replace_with_thresholds(dataframe, "Price")

    return dataframe
```

```
df = retail_data_prep(df)

df.describe().T

df_fr = df[df['Country'] == "France"]

df_fr.groupby(['BillNo', 'Itemname']).agg({"Quantity":
"sum"}).unstack().fillna(0).iloc[0:5, 0:5]

fr_inv_pro_df=df_fr.groupby(['BillNo', 'Itemname']). \
            agg({"Quantity": "sum"}). \
            unstack(). \
            fillna(0). \
            applymap(lambda x: 1 if x > 0 else 0)

frequent_itemsets = apriori(fr_inv_pro_df.astype("bool"),
                       min_support=0.01,
                       use_colnames=True)

frequent_itemsets.sort_values("support", ascending=False).head()
```

Output:

	count	mean	min	25%	50%	75%	max	std
Quantity	519551.0	9.39742	1.0	1.0	3.0	10.0	248.5	21.281261
Date	519551	2011-07-04 16:03:31.051080704	2010-12-01 08:26:00	2011-03-28 10:52:00	2011-07-20 11:55:00	2011-10-19 15:08:00	2011-12-09 12:50:00	NaN
Price	519551.0	3.32647	0.001	1.25	2.08	4.13	41.94	3.87738
CustomerID	387985.0	15317.042994	12346.0	13950.0	15265.0	16837.0	18287.0	1721.813298



	Quantity				
Itemname	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND
BillNo					
536370	0.0	0.0	0.0	0.0	0.0
536852	0.0	0.0	0.0	0.0	0.0
536974	0.0	0.0	0.0	0.0	0.0
537065	0.0	0.0	0.0	0.0	0.0
537463	0.0	0.0	0.0	0.0	0.0

	support	itemsets
330	0.765306	((Quantity, POSTAGE))
332	0.188776	((Quantity, RABBIT NIGHT LIGHT))
371	0.181122	((Quantity, RED TOADSTOOL LED NIGHT LIGHT))
320	0.170918	((Quantity, PLASTERS IN TIN WOODLAND ANIMALS))
315	0.168367	((Quantity, PLASTERS IN TIN CIRCUS PARADE))

This code imports the pandas library and loads your transaction dataset from a CSV file named 'data.csv' into a DataFrame named 'data'.

In this section, it prepares the data for association analysis by creating a list of transactions. It iterates over the 'data' DataFrame, grouping transactions and creating a list of unique items within each transaction. These lists are stored in the 'transactions' list.

This code imports the necessary functions from the mlxtend library, which is commonly used for association analysis. It then converts the 'transactions' list into a one-hot encoded DataFrame ('oht'). One-hot encoding converts the list of transactions into a binary matrix where each row corresponds to a transaction, each column represents an item, and the values are binary indicators (0 or 1) of whether an item is present in the transaction.

Here, the Apriori algorithm is applied to the one-hot encoded data to find frequent itemsets. 'min_support' is set to 0.1, which means that only itemsets appearing in at least 10% of the transactions are considered frequent. 'use_colnames=True' ensures that the item names are used in the results.

After finding frequent itemsets, this code generates association rules using the association_rules function. It uses 'confidence' as the metric to evaluate rule strength and sets a minimum confidence threshold of 0.7 to filter the rules. You can adjust the threshold according to your requirements.

The code provided helps you load your transaction data, transform it into a suitable format for association analysis, find frequent item sets using the Apriori algorithm, and generate association rules based on your specified criteria. You can then further filter and interpret these rules to gain insights into market basket patterns.

Conclusion:

Market basket insights can lead to increased sales, improved customer satisfaction, and better inventory management. By understanding customer behaviour and preferences, businesses can tailor their strategies to meet the needs of their target audience and stay competitive in their market. Market basket insights, also known as market basket analysis or association analysis, are essential for businesses to understand the purchasing behaviour of their customers. This analysis helps identify patterns and relationships between products or services that customers tend to buy together. It is particularly valuable for retail, e-commerce, and other industries where cross-selling, upselling, and personalized recommendations are crucial. Here's a general outline of how to develop market basket insights: The code provided helps you load your transaction data, transform it into a suitable format for association analysis, find frequent itemsets using the Apriori algorithm, and generate association rules based on your specified criteria. You can then further filter and interpret these rules to gain insights into market basket patterns.