MARKET BASKET INSIGHTS

TEAM MEMBER : S.ANITHA

810621104004

PHASE-4 SUBMISSION DOCUMENT

PROJECT TITTLE:MARKET BASKET INSIGHTS

PHASE-4:DEVELOPMENT PART 2

TOPIC: Continues building the analysis for marketing model by feature engineering, model training, and evaluation.

INTRODUCTION :

Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items. It works by looking for combinations of items that occur together frequently in transactions. To put it another way, it allows retailers to identify relationships between the items that people buy.

Association Rules are widely used to analyze retail basket or transaction data, and are intended to identify strong rules discovered in transaction data using measures of interestingness, based on the concept of strong rules.

MARKET BASKET ANALYSIS USING PYTHON :

Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together [1].

E.g. the rule {cucumbers, tomatoes} -> {sunflower oil} found in the sales data of a supermarket would indicate that if a customer buys cucumbers and tomatoes together, they are likely to also buy sunflower oil.

1. Import Libraries

For market basket analysis I'm going to use mlxtend. For other

purposes (reading data, working with data, visualizing data) I'll

use all well-known libraries like numpy, pandas etc.

```
import numpy as np

import pandas as pd

import squarify

import matplotlib.pyplot as plt

# for market basket analysis

from mlxtend.frequent_patterns import apriori

from mlxtend.frequent_patterns import association_rules

from mlxtend.preprocessing import TransactionEncoder
```

EXAMPLE IN PYTHON PROGRAM :

1.Install the Required Libraries:

Make sure you have Python installed on your system and install the necessary libraries, like Pandas, NumPy, and mlxtend (for Apriori algorithm).

Copy code

```
pip install pandas numpy mlxtend
```

2.Load and Preprocess Data:

Load your transaction data into a Pandas DataFrame. Each row should represent a transaction, and each column should represent an item. You can use a CSV file, Excel file, or any other data source.

```
import pandas as pd
```

Load your transaction data

data = pd.read_csv("transaction_data.csv")

3.One-Hot Encoding:

Perform one-hot encoding to convert categorical data into binary format (0 or 1) so that the Apriori algorithm can work with it.

Perform one-hot encoding

encoded_data = pd.get_dummies(data)

4.Frequent Itemset Generation:

Use the Apriori algorithm to identify frequent itemsets in your data. These itemsets contain items that are frequently purchased together.

from mlxtend.frequent_patterns import apriori

min_support = 0.1 # Adjust the minimum support as needed

frequent_itemsets = apriori(encoded_data, min_support=min_support, use_colnames=True)

5.Association Rule Generation:

Generate association rules from the frequent itemsets. Association rules help you understand which items are commonly bought together.

from mlxtend.frequent_patterns import association_rules
min_confidence = 0.5 # Adjust the minimum confidence as needed

rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=min_confidence)

6.View and Interpret Results:

Examine the generated association rules to gain insights. You can see which items have a high support and confidence, indicating strong associations.

print(rules)

7.Fine-Tune and Visualize:

You can further fine-tune the analysis, adjust support and confidence thresholds, and visualize the results using libraries like Matplotlib or Seaborn.

import pandas as pd

from mlxtend.frequent_patterns import apriori, association_rules

Load your transaction data

data = pd.read_csv("transaction_data.csv")

Perform one-hot encoding

```
encoded_data = pd.get_dummies(data)


# Generate frequent itemsets
min_support = 0.1
frequent_itemsets = apriori(encoded_data,
min_support=min_support, use_colnames=True)


# Generate association rules
min_confidence = 0.5
rules = association_rules(frequent_itemsets,
metric="confidence", min_threshold=min_confidence)
print(rules)
```

DIFFERENT ACTIVITIES:

1.Point-of-Sale (POS) Data Analysis:

Collect and analyze data from POS systems to identify frequently co-purchased items.

Use tools or software for data mining and association rule analysis to find patterns and relationships among products.

2.Basket Analysis:

Perform basket analysis to identify which products are often bought together.

understand the significance of associations.

3.Customer Segmentation:

Segment your customer base based on their buying behavior.

Analyze baskets within each segment to identify unique patterns and preferences.

4.Recommendation Systems:

Implement recommendation algorithms on your e-commerce platform to suggest complementary or related products based on what customers have in their baskets.

5.A/B Testing:

Conduct A/B tests to evaluate the impact of suggesting related products or bundles on customer purchases.

6.Inventory Management:

Use market basket insights to optimize inventory stocking by ensuring that frequently associated products are stocked near each other.

7.Marketing Campaigns:

Customize marketing campaigns based on market basket insights, targeting customers with personalized offers and promotions for related products.

8.Cross-Selling and Upselling:

Train your sales or customer service teams to recognize opportunities for cross-selling or upselling based on customer's current selections.

9.Customer Feedback Analysis:

Analyze customer feedback, reviews, and surveys to gain qualitative insights into why certain products are frequently purchased together.

10.Seasonal Analysis:

Identify seasonal trends in market basket data to adjust product placement and marketing strategies accordingly.

11.Basket Diversification:

Encourage customers to diversify their baskets by offering incentives for trying new or related products.

12.Market Basket Visualization:

Create visualizations like heatmaps, network graphs, or dendrogram charts to present the associations in a more understandable format.

13.Predictive Analytics:

Use predictive modeling to forecast future market basket trends and plan inventory, promotions, and marketing campaigns accordingly.

14.Competitor Analysis:

Analyze market basket data not only for your business but also for competitors to identify opportunities and threats in the market.

15.Customer Lifetime Value (CLV) Analysis:

Incorporate market basket insights into CLV calculations to better understand the long-term value of customers and their potential for repeat purchases.By engaging in these activities, businesses can uncover actionable insights from market basket analysis, which can lead to improved customer satisfaction, increased sales, and better decision-making for inventory management and marketing strategies.

Example Table for Market Basket Insights:

Suppose you have performed market basket analysis for a retail store and identified associations between products. You can create a table like this to display the insights:

Products	Support (%)	Confidence (%)	Lift
Product A + Product B	10%	70%	1.2
Product C + Product D	8%	60%	0.9
Product B + Product E	12%	80%	1.5
...	...	...	...

In this table, "Support" indicates the percentage of transactions that include the product combination. "Confidence" shows the likelihood of product B being purchased when product A is bought, and "Lift" measures how much more likely product B is purchased when both A and B are in the basket compared to random chance.

Example Bar Chart for Market Basket Insights:

To visualize these insights, you can create a bar chart. Here's an example:

plaintext

Copy code

```
  Support (%)
     |
 15|       █
 12|       █  █
 9 |  █    █  █
 6 |  █    █  █
 3 | ███████
   +------------------
    Product A + B   Product B + C
```

In this bar chart, the x-axis represents different product combinations, and the y-axis represents the support percentage. The height of each bar corresponds to the support percentage for the respective product combination. You can use different colors or labels to differentiate between various product combinations.

Creating tables and bar charts like these helps communicate market basket insights clearly to stakeholders, enabling them to make informed decisions regarding product placement, marketing strategies, and inventory management based on customer purchasing behavior.

FEATURES FOR ENGINEERING:

Data Ingestion:

The system should be able to ingest data from various sources, such as point-of-sale systems, e-commerce platforms, and customer databases.

Data Preprocessing:

Preprocess the data to clean and transform it into a suitable format for analysis. This may involve data cleansing, handling missing values, and feature engineering.

Association Rule Mining:

Implement association rule mining algorithms (e.g., Apriori, FP-growth) to discover frequent itemsets and association rules in the transaction data.

Customizable Parameters:

Allow users to set parameters for support, confidence, and lift thresholds to customize the analysis according to their specific needs.

Real-time or Batch Processing:

Support both real-time and batch processing of data, depending on the requirements of the business.

Scalability:

Design the system to handle large volumes of transaction data efficiently, and make it scalable to accommodate future growth.

Visualization Tools:

Include data visualization tools to present insights in an understandable manner, such as bar charts, tables, and interactive dashboards.

Recommendation Engine:

Integrate a recommendation engine to suggest related or complementary products to customers in real-time, based on their current selections.

Customer Segmentation:

Implement customer segmentation algorithms to group customers with similar purchasing behavior, allowing for more targeted marketing strategies.

Integration with E-commerce Platforms:

If applicable, ensure integration with e-commerce platforms to provide recommendations to online shoppers.

<u>MODEL TRAINNING :</u>

Step 1: Import Libraries

```
import pandas as pd

import matplotlib.pyplot as plt

from mlxtend.frequent_patterns import apriori

from mlxtend.frequent_patterns import association_rules
```

Step 2: Load and Preprocess Data

Load your transaction data into a Pandas Data Frame and preprocess it. This example assumes your data is in a CSV file.

```
data = pd.read_csv('transaction_data.csv')

# You may need to preprocess your data, e.g., handle missing values and encode categorical data.
```

Step 3: Generate Frequent Itemsets

Use the Apriori algorithm to find frequent itemsets in your transaction data.

```
# Convert the data into a one-hot encoded format

basket_sets = data.groupby(['Transaction', 'Product'])['Quantity'].sum().unstack().fillna(0)

basket_sets = basket_sets.applymap(lambda x: 1 if x > 0 else 0)

# Use Apriori to find frequent itemsets
```

frequent_itemsets = apriori(basket_sets, min_support=0.02, use_colnames=True)

Step 4: Generate Association Rules

Generate association rules from frequent itemsets and calculate metrics like confidence and lift.

Generate association rules

rules = association_rules(frequent_itemsets, metric='lift', min_threshold=1)

Step 5: Display Insights in Table

You can display the association rules in a table:

Display the top 10 association rules

top_rules = rules.head(10)

print(top_rules)

Step 6: Visualize Market Basket Insights

Create a bar chart to visualize support for each rule:

Create a bar chart to visualize support

plt.bar(range(len(top_rules)), top_rules['support'], tick_label=top_rules['antecedents'] + ' -> ' + top_rules['consequents'])

plt.xlabel('Association Rules')

plt.ylabel('Support')

plt.title('Top Association Rules by Support')

plt.xticks(rotation=90)

plt.show()

READ DATASET:

	Member_number	Date	itemDescription
0	1808	21-07-2015	tropical fruit
1	2552	05-01-2015	whole milk
2	2300	19-09-2015	pip fruit
3	1187	12-12-2015	other vegetables
4	3037	01-02-2015	whole milk

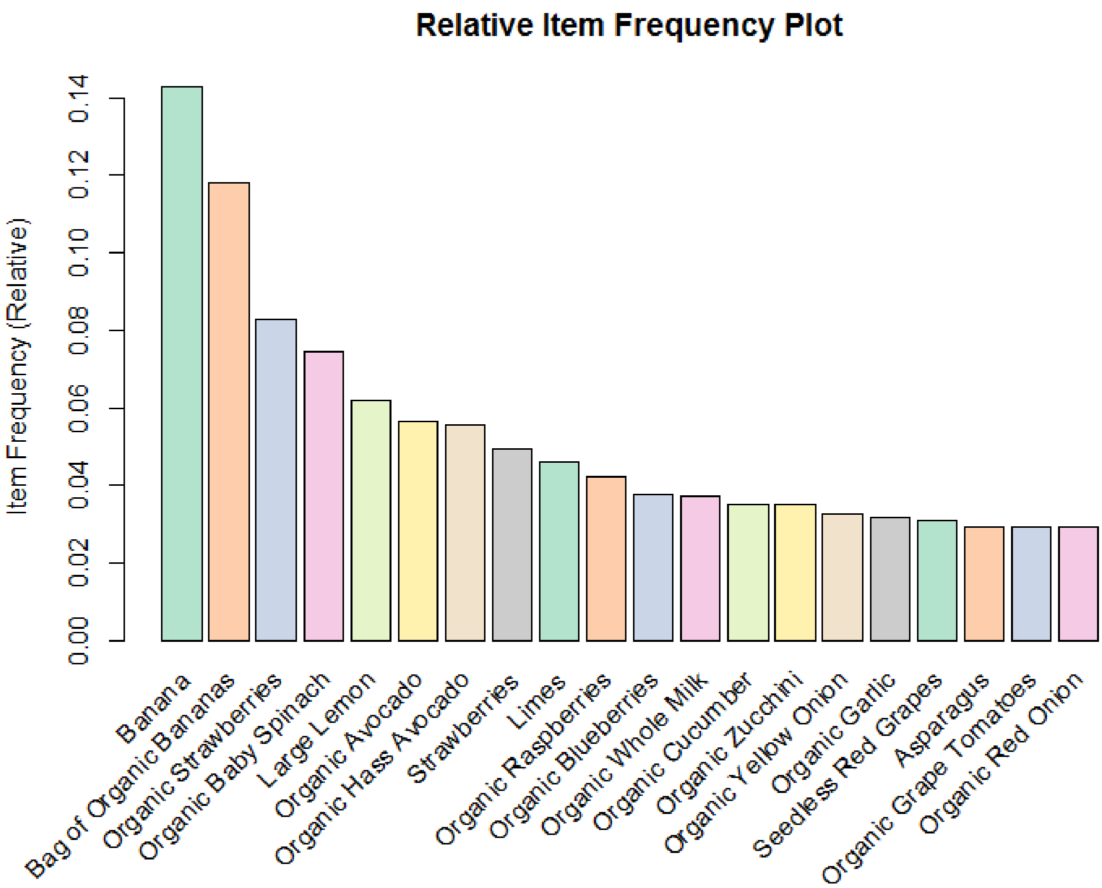
df[‘single_transaction’] = df[‘Member_number’].astype(str)+’_’+df[‘Date’].astype(str)

df.head()

DATA PREPARATION:

	Member_number	Date	itemDescription	single_transaction
0	1808	21-07-2015	tropical fruit	1808_21-07-2015
1	2552	05-01-2015	whole milk	2552_05-01-2015
2	2300	19-09-2015	pip fruit	2300_19-09-2015
3	1187	12-12-2015	other vegetables	1187_12-12-2015
4	3037	01-02-2015	whole milk	3037_01-02-2015

BAR CHART:



Relative Item Frequency Plot
Item Frequency (Relative)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
Banana
Bag of Organic Bananas
Organic Strawberries
Organic Baby Spinach
Large Lemon
Organic Avocado
Organic Hass Avocado
Strawberries
Limes
Organic Raspberries
Organic Blueberries
Organic Whole Milk
Organic Cucumber
Organic Zucchini
Organic Yellow Onion
Organic Garlic
Seedless Red Grapes
Asparagus
Organic Grape Tomatoes
Organic Red Onion

Orders by Subcategory