Phase-4

Model Development and Evaluation

Date	25 October 2023
Team ID	Proj-212168-Team-2
Project Name	Market Basket Insights
Maximum marks	

Developing and evaluating a market basket analysis model typically involves the use of association rule mining algorithms, such as the Apriori algorithm, and the evaluation of these rules using relevant metrics. You can use programming languages like Python to accomplish this task. Below, I'll provide a step-by-step example of how to develop and evaluate a market basket analysis model using Python, specifically with the mlxtend library.

Steps:

- First, you load your transaction data into a pandas DataFrame. Each row represents a transaction, and each column represents an item, with binary values (1 for purchased, 0 for not purchased).
- Then use the Apriori algorithm to find frequent itemsets based on a minimum support threshold.
- Association rules are generated using a minimum confidence threshold.
- The code then displays the frequent itemsets and association rules.
- You can evaluate the rules based on other metrics like lift, conviction, etc. In the example, we filtered the rules with a minimum lift threshold of 0.5.

Market basket analysis can be a powerful tool for understanding customer behavior and optimizing product recommendations. You can customize the minimum support and confidence thresholds as well as other evaluation criteria to suit your specific business needs.

Formatting the transaction data in a suitable format for analysis

Split the 'Itemname' column in transaction_data into individual items using str.split(', ', expand=True).Concatenate the original DataFrame (transaction_data) with the items DataFrame (items_df) using pd.concat.Drop the original 'Itemname' column since individual items are now in separate columns.Display the resulting DataFrame.

In [4]:
```
df= pd.DataFrame(dataset)
items_df = df['Itemname'].str.split(', ', expand=True)

transaction_data = pd.concat([df, items_df], axis=1)

transaction_data = transaction_data.drop('Itemname', axis=1)


print(transaction_data.head())
```

```
                                   0                             1  \
0  WHITE HANGING HEART T-LIGHT HOLDER           WHITE METAL LANTERN
1             HAND WARMER UNION JACK    HAND WARMER RED POLKA DOT
2       ASSORTED COLOUR BIRD ORNAMENT    POPPY'S PLAYHOUSE BEDROOM
3            JAM MAKING SET WITH JARS  RED COAT RACK PARIS FASHION
4            BATH BUILDING BLOCK WORD                         None

                               2                                    3
\
0  CREAM CUPID HEARTS COAT HANGER  KNITTED UNION FLAG HOT WATER BOTTLE
1                            None                                 None
2       POPPY'S PLAYHOUSE KITCHEN    FELTCRAFT PRINCESS CHARLOTTE DOLL
3  YELLOW COAT RACK PARIS FASHION         BLUE COAT RACK PARIS FASHION
4                            None                                 None

                               4                                   5   \
0  RED WOOLLY HOTTIE WHITE HEART.        SET 7 BABUSHKA NESTING BOXES
1                            None                                None
```

```
0  RED WOOLLY HOTTIE WHITE HEART.        SET 7 BABUSHKA NESTING BOXES
1                             None                                None
2          IVORY KNITTED MUG COSY  BOX OF 6 ASSORTED COLOUR TEASPOONS
3                             None                                None
4                             None                                None

                                6                                7   \
0  GLASS STAR FROSTED T-LIGHT HOLDER                           None
1                               None                           None
2       BOX OF VINTAGE JIGSAW BLOCKS  BOX OF VINTAGE ALPHABET BLOCKS
3                               None                           None
4                               None                           None

                       8                          9   ...   534   535
536  \
0                      None                       None  ...  None  None
None
1                      None                       None  ...  None  None
None


1                      None                       None  ...  None  None
None
2  HOME BUILDING BLOCK WORD  LOVE BUILDING BLOCK WORD  ...  None  None
None
3                      None                       None  ...  None  None
None
4                      None                       None  ...  None  None
None

    537   538   539   540   541   542   543
0  None  None  None  None  None  None  None
1  None  None  None  None  None  None  None
2  None  None  None  None  None  None  None
3  None  None  None  None  None  None  None
4  None  None  None  None  None  None  None

[5 rows x 544 columns]
```

Converting items to Boolean columns:

To prepare the data for association rule mining, we convert the items in the transaction_data DataFrame into boolean columns using one-hot encoding. This is achieved through the pd.get_dummies function, which creates a new DataFrame (df_encoded) with boolean columns representing the presence or absence of each item.

```
In [5]: df_encoded = pd.get_dummies(transaction_data, prefix='', prefix_sep='').groupby(level=0, axis=1).max()
        df_encoded.to_csv('transaction_data_encoded.csv', index=False)
```

Association Rule mining :

Apply the Apriori algorithm to perform association rule mining on the encoded transaction data. The min_support parameter is set to 0.007 to filter out infrequent itemsets. The resulting frequent itemsets are then used to generate association rules based on a minimum confidence threshold of 0.5. Print the generated association rules.

```
[2]:
    frequent_itemsets = apriori(df_encoded, min_support=0.007, use_colnames=True)
    rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5)


    print("Association Rules:")
    print(rules.head())
```

```
Association Rules:
                        antecedents                          consequents  \
0             (CHOCOLATE BOX RIBBONS)           (6 RIBBONS RUSTIC CHARM)
1  (60 CAKE CASES DOLLY GIRL DESIGN)  (PACK OF 72 RETROSPOT CAKE CASES)
2      (60 TEATIME FAIRY CAKE CASES)  (PACK OF 72 RETROSPOT CAKE CASES)
3   (ALARM CLOCK BAKELIKE CHOCOLATE)       (ALARM CLOCK BAKELIKE GREEN)
4   (ALARM CLOCK BAKELIKE CHOCOLATE)        (ALARM CLOCK BAKELIKE PINK)

   antecedent support  consequent support   support  confidence       li
ft  \
0            0.012368            0.039193  0.007036    0.568889  14.5150
44
1            0.018525            0.054529  0.010059    0.543027   9.9584
09
2            0.034631            0.054529  0.017315    0.500000   9.1693
55

3            0.017150            0.042931  0.011379    0.663462  15.4541
51
4            0.017150            0.032652  0.009125    0.532051  16.2947
42

   leverage  conviction  zhangs_metric
0  0.006551    2.228676       0.942766
1  0.009049    2.068984       0.916561
2  0.015427    1.890941       0.922902
3  0.010642    2.843862       0.951613
4  0.008565    2.067210       0.955009
```

Visualization:

Use matplotlib and seaborn libraries to create a scatterplot visualizing the results of the market basket analysis. The plot depicts the relationship between support, confidence, and lift for the generated association rules.

In [3]: 4.2 seconds Explain

```
import matplotlib.pyplot as plt
import seaborn as sns


plt.figure(figsize=(12, 8))
sns.scatterplot(x="support", y="confidence", size="lift", data=rules, hue="lift", palette="viridis", sizes=(20, 200))
plt.title('Market Basket Analysis - Support vs. Confidence (Size = Lift)')
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.legend(title='Lift', loc='upper right', bbox_to_anchor=(1.2, 1))
plt.show()
```