MARKET BASKET INSIGHTS

TEAM MEMBERS

810021106036:KANIKARAN K

PHASE 3:PROJECT SUBMISSION

This program describes about market basket analysis using python

PROGRAM:

```
# importing the libraries

Import numpy as np

Import pandas as pd

From matplotlib import pyplot as plt

Import seaborn as sns

From csv import reader

From mlxtend.preprocessing import TransactionEncoder

From mlxtend.frequent_patterns import apriori, association_rules

# reading the dataset

Groceries = []

With open('../input/groceries/groceries.csv', 'r') as read_obj:
```

```
    Csv_reader = reader(read_obj)
    For row in csv_reader:
        Groceries.append(row)
 items = set(sum(groceries, []))
df = pd.DataFrame(columns=items)
 for I in range(len(groceries)):
   transaction = []
     for item in items:
         if item in groceries[i]:
            transaction.append(1)
         else:
             transaction.append(0)
     print(transaction)
     df = df.append(transaction, ignore_index=True)
# fitting the list and converting the transactions to true and
false
Encoder = TransactionEncoder()
Transactions = encoder.fit(groceries).transform(groceries)
# converting the true and false to 1 and 0
Transactions = transactions.astype('int')
```

```
# converting the transactions array to a datafrmae
Df = pd.DataFrame(transactions, columns=encoder.columns_)
# viewing the first few rows of the dataframe
Df.head()
# applying the apriori algorithm
Frequent_itemsets = apriori(df,
min_support=0.02,use_colnames=True)
frequent_itemsets['length'] =
frequent_itemsets['itemsets'].apply(lambda x: len(x))
frequent_itemsets
# sorting the dataframe
Frequent_itemsets =
frequent_itemsets.sort_values(by='support', ascending=False)
# finding itemsets having length 2 and minimum support of 2%
Frequent_itemsets[(frequent_itemsets['length'] == 2) &
(frequent_itemsets['support'] >= 0.02)]
# finding itemsets having length more than 1 and minimum
support of 5%
Frequent_itemsets[(frequent_itemsets['length'] > 1) &
(frequent_itemsets['support'] >= 0.05)]
# finding itemsets having length 2 and minimum support of 2%
Frequent_itemsets[(frequent_itemsets['length'] == 2) &
```

```
(frequent_itemsets['support'] >= 0.02)]

# finding top 10 association rules with minimum support of 2%

rules = association_rules(frequent_itemsets, metric='support', min_threshold=0.02)rules

# sorting the rules in the descending order by confidence

rules.sort_values(by='confidence', ascending=False)[0:10]

# finding association rules with minimum support of 2% and having lift more than 1

rules[(rules['support'] >= 0.02) &   (rules['lift'] > 1.0)]

#Create a pie chart to show distribution of transactions

Plt.figure(figsize=[8,8])

Plt.pie(top10,labels=top10.index, autopct = '%0.0f%%',labeldistance=1.3)

Plt.title("Distribution of Transactions by Country")

Plt.show()
```

OUTPUT:

FIRST FEW ROWS OF THE DATAFRAME (5rows×5columns)

	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener	baby cosmetics
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

(122rows×5columns)

	support	itemsets	length
0	0.033452	(UHT-milk)	1
1	0.052466	(beef)	1
2	0.033249	(berries)	1
3	0.026029	(beverages)	1
4	0.080529	(bottled beer)	1
...	...	...	...
117	0.032232	(whipped/sour cream, whole milk)	2
118	0.020742	(yogurt, whipped/sour cream)	2
119	0.056024	(yogurt, whole milk)	2
120	0.023183	(root vegetables, other vegetables, whole milk)	3
121	0.022267	(other vegetables, yogurt, whole milk)	3

TOP 5 ITEMS WITH MINIMUM SUPPORT OF 2%

support	itemsets	length
0.255516	(whole milk)	1
0.193493	(other vegetables)	1
0.183935	(rolls/buns)	1
0.174377	(soda)	1
0.139502	(yogurt)	1

ITEMSETS HAVING LENGTH MORE THAN 1 AND MINIMUM SUPPORT OF 5%

	support	itemsets	length
91	0.074835	(other vegetables, whole milk)	2
103	0.056634	(rolls/buns, whole milk)	2
119	0.056024	(yogurt, whole milk)	2

SORTING THE RULES IN THE DEFENDING ORDER BY CONFIDENCE (5rows×5columns)

antecedents	consequents	antecedent support	consequent support	antecedent support
(other vegetables)	(whole milk)	0.193493	0.255516	0.193493
(whole milk)	(other vegetables)	0.255516	0.193493	0.255516
(rolls/buns)	(whole milk)	0.183935	0.255516	0.183935
(whole milk)	(rolls/buns)	0.255516	0.183935	0.255516
(yogurt)	(whole milk)	0.139502	0.255516	0.139502
...	...	...	...	...

DISTRIBUTION OF TRANSACTION BY COUNTY