Association Rule Mining

Name - Amith Korada

Course - AI & ML
(Batch - 4)

Duration - 12 Months

Problem Statement - Using Apriori algorithm try to find the rules that describe the relation between each of the products that were brought by the customers a

Prerequisites -

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has the latest version of python. The following URL https://www.python.org/downloads/ can be referred to as download python.

The second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this URL https://www.anaconda.com/download/ You will also need to download and install the below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run the below commands in command prompt/terminal to install these packages pip install -U sci-kit-learn pip install NumPy pip install scipy if you have chosen to install anaconda then run the below commands in anaconda prompt to install these packages conda install -c sci-kit-learn conda install -c anaconda numpy conda install -c anaconda scipy.

1. Importing necessary libraries-

```
import numpy as np
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

2. Loading the dataset-

```
df = pd.read_csv("Store_data.csv",header=None)
```

```
df.head(20)
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	sr
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
5	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

```
df.values[0,:]
```

```
array(['shrimp', 'almonds', 'avocado', 'vegetables mix', 'green grapes',
       'whole weat flour', 'yams', 'cottage cheese', 'energy drink',
       'tomato juice', 'low fat yogurt', 'green tea', 'honey', 'salad',
       'mineral water', 'salmon', 'antioxydant juice', 'frozen smoothie',
       'spinach', 'olive oil'], dtype=object)
```

```
df.shape
```

```
(7501, 20)
```

3. One-hot Encoding-

```
items = []
```

```
for index, row in df.iterrows():
    clean_items = [item for item in row if item is not np.nan]
    items.append(clean_items)
```

```
tran_encod = TransactionEncoder()
tran_encod_list = tran_encod.fit(items).transform(items)
transaction_df = pd.DataFrame(tran_encod_list, columns=tran_encod.columns_)
transaction_df.head()
```

	asparagus	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	black tea	blueberries	...	turkey	vegetables mix	water spray	white wine	whole weat flour	whole wheat pasta	wh wh
0	False	True	True	False	True	False	False	False	False	False	...	False	True	False	False	True	False	Fa
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fa
3	False	False	False	False	True	False	False	False	False	False	...	True	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	T

5 rows × 120 columns

4. Calculating item frequency-

```
item_count = {}
for col in transaction_df.columns:
    item_count[col] = transaction_df[col].sum()
item_freq_df = pd.DataFrame(data=list(item_count.values()), index=list(item_count.keys()), columns=['frequency']).sort_values(by=
```

```
item_freq_df.shape
```

```
(120, 1)
```

```
item_freq_df.head(10)
```

	frequency
mineral water	3576
eggs	2696
spaghetti	2612
french fries	2564
chocolate	2458
green tea	1982
milk	1944
ground beef	1474
frozen vegetables	1430
pancakes	1426

5. Frequent Itemsets-

```
freq_itemsets = apriori(transaction_df, min_support=0.03, use_colnames=True)
freq_itemsets
```

	support	itemsets
0	0.033329	(avocado)
1	0.033729	(brownies)
2	0.087188	(burgers)
3	0.030129	(butter)
4	0.081056	(cake)
5	0.046794	(champagne)
6	0.059992	(chicken)
7	0.163845	(chocolate)
8	0.080389	(cookies)
9	0.051060	(cooking oil)
10	0.031862	(cottage cheese)
11	0.179709	(eggs)
12	0.079323	(escalope)
13	0.170911	(french fries)
14	0.043061	(fresh bread)
15	0.063325	(frozen smoothie)

6. Association rules-

```
rules = association_rules(freq_itemsets, metric='confidence',min_threshold=0.2)
rules.head(10)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(chocolate)	(eggs)	0.163845	0.179709	0.033196	0.202604	1.127397	0.003751	1.028711
1	(chocolate)	(french fries)	0.163845	0.170911	0.034395	0.209927	1.228284	0.006393	1.049383
2	(french fries)	(chocolate)	0.170911	0.163845	0.034395	0.201248	1.228284	0.006393	1.046827
3	(milk)	(chocolate)	0.129583	0.163845	0.032129	0.247942	1.513276	0.010898	1.111823
4	(mineral water)	(chocolate)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256
5	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
6	(chocolate)	(spaghetti)	0.163845	0.174110	0.039195	0.239219	1.373952	0.010668	1.085581
7	(spaghetti)	(chocolate)	0.174110	0.163845	0.039195	0.225115	1.373952	0.010668	1.079070
8	(eggs)	(french fries)	0.179709	0.170911	0.036395	0.202522	1.184961	0.005681	1.039640
9	(french fries)	(eggs)	0.170911	0.179709	0.036395	0.212949	1.184961	0.005681	1.042232

```
rules.sort_values('lift', ascending=False)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
19	(spaghetti)	(ground beef)	0.174110	0.098254	0.039195	0.225115	2.291162	0.022088	1.163716
18	(ground beef)	(spaghetti)	0.098254	0.174110	0.039195	0.398915	2.291162	0.022088	1.373997
17	(ground beef)	(mineral water)	0.098254	0.238368	0.040928	0.416554	1.747522	0.017507	1.305401
15	(frozen vegetables)	(mineral water)	0.095321	0.238368	0.035729	0.374825	1.572463	0.013007	1.218270
23	(spaghetti)	(milk)	0.174110	0.129583	0.035462	0.203675	1.571779	0.012900	1.093043
22	(milk)	(spaghetti)	0.129583	0.174110	0.035462	0.273663	1.571779	0.012900	1.137061
21	(mineral water)	(milk)	0.238368	0.129583	0.047994	0.201342	1.553774	0.017105	1.089850
20	(milk)	(mineral water)	0.129583	0.238368	0.047994	0.370370	1.553774	0.017105	1.209650
3	(milk)	(chocolate)	0.129583	0.163845	0.032129	0.247942	1.513276	0.010898	1.111823
24	(pancakes)	(mineral water)	0.095054	0.238368	0.033729	0.354839	1.488616	0.011071	1.180529
25	(mineral water)	(spaghetti)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008
26	(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314
7	(spaghetti)	(chocolate)	0.174110	0.163845	0.039195	0.225115	1.373952	0.010668	1.079070
6	(chocolate)	(spaghetti)	0.163845	0.174110	0.039195	0.239219	1.373952	0.010668	1.085581
5	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
4	(mineral water)	(chocolate)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256
10	(milk)	(eggs)	0.129583	0.179709	0.030796	0.237654	1.322437	0.007509	1.076009