Knowledge Discovery & Data Mining Lab5

Name : Kunal Sanjay Patil

PRN : 20190802025

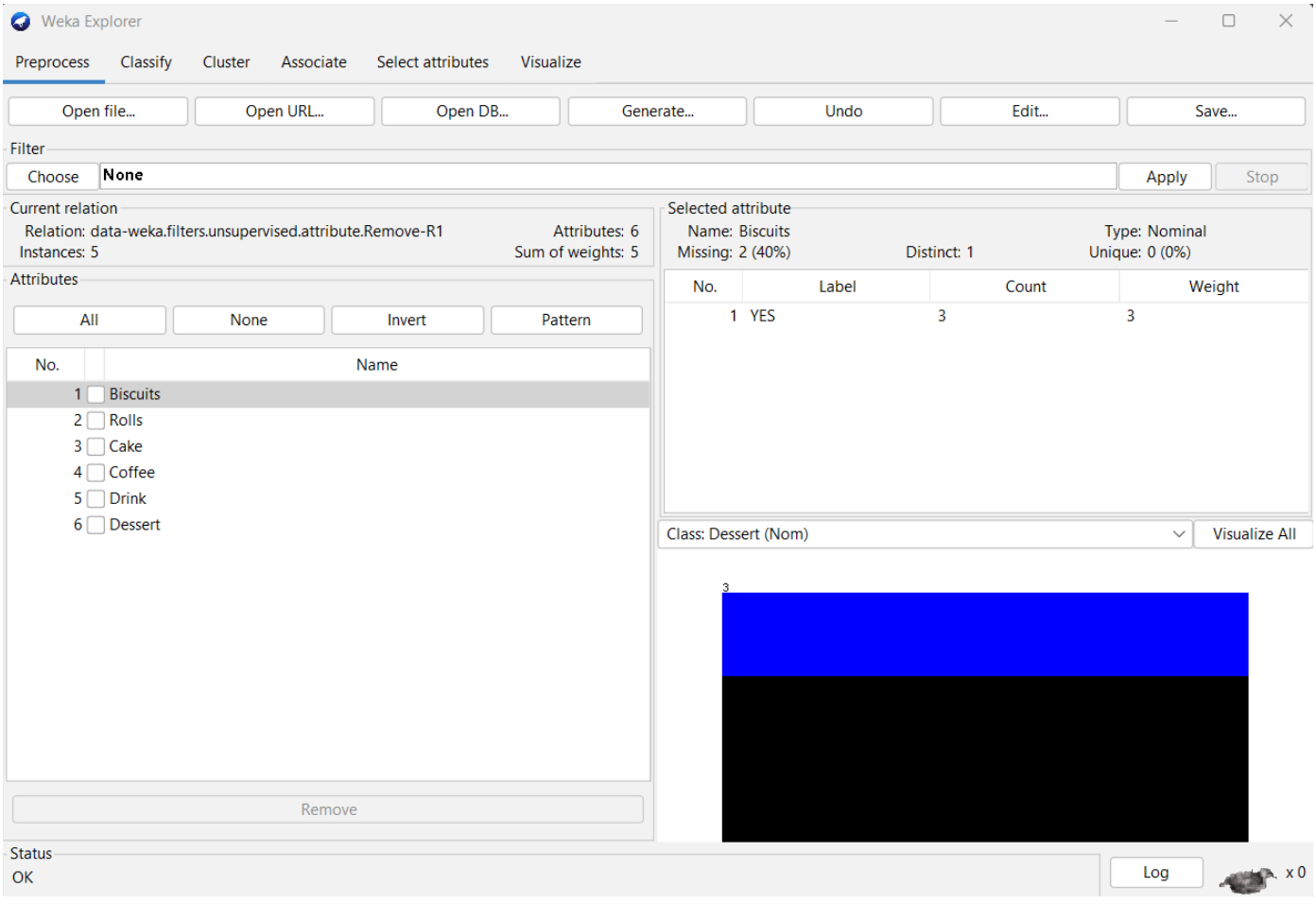
AIM :

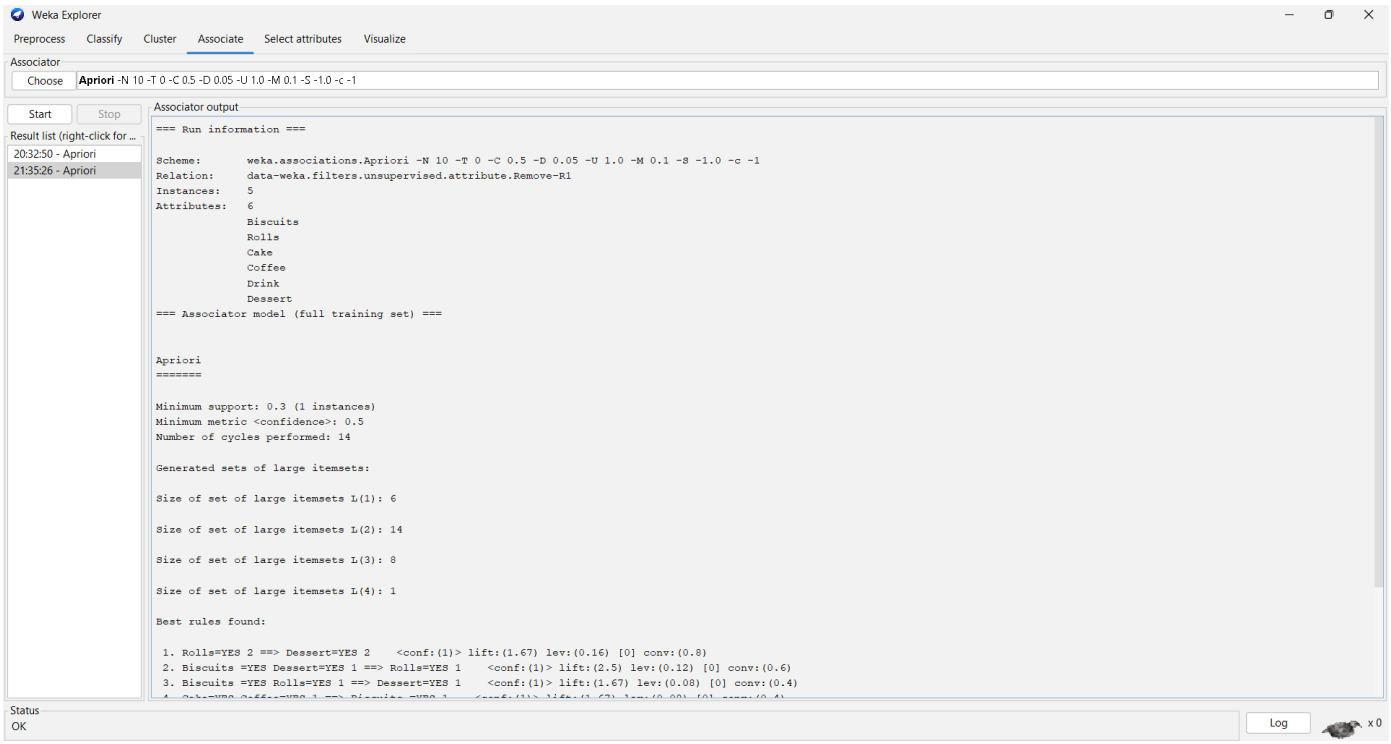
To implement the Apriori Algorithm using WEKA and python.

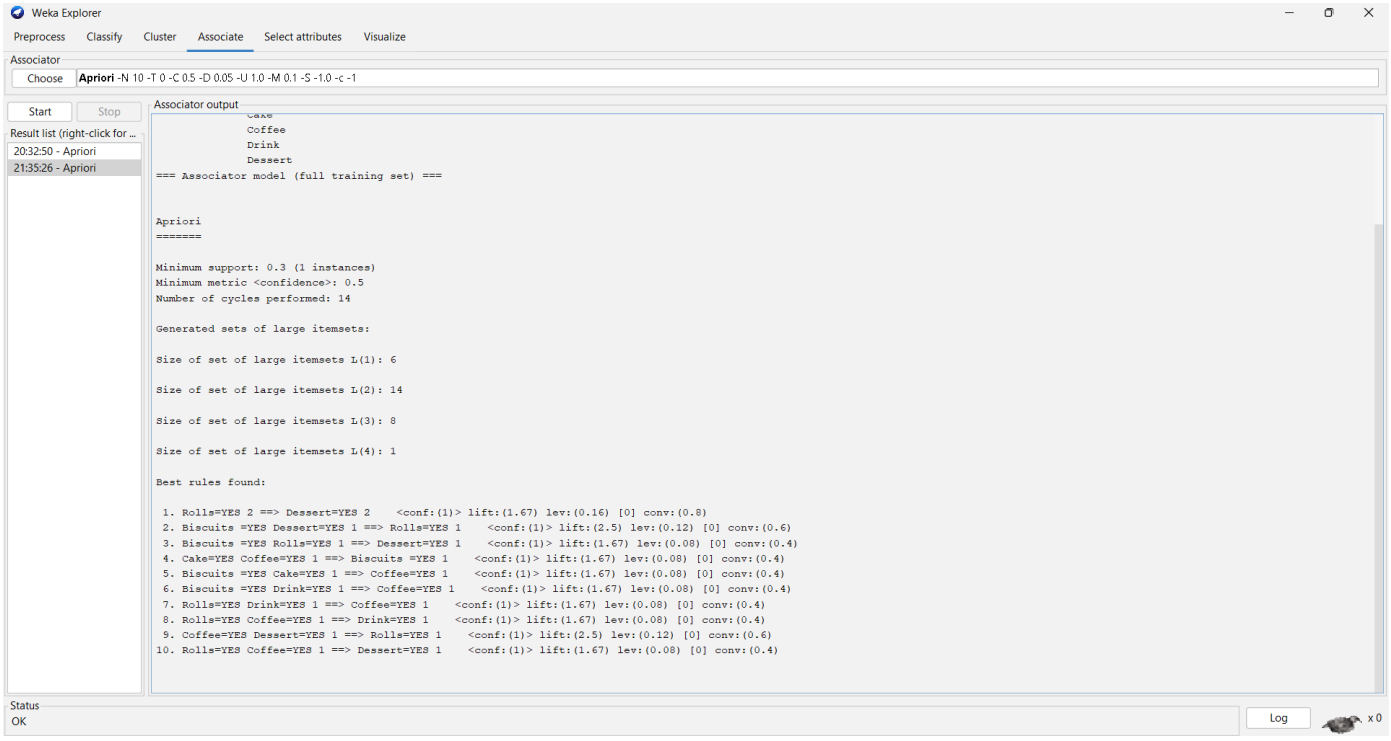
Apriori Algorithm using WEKA :-

Dataset used :-

	A	B	C	D	E	F	G
1	Transaction Id	Biscuits	Rolls	Cake	Coffee	Drink	Dessert
2	1	YES		YES	YES		
3	2			YES		YES	YES
4	3	YES			YES	YES	
5	4	YES	YES				YES
6	5		YES		YES	YES	YES







Apriori Algorithm using python :-

In [1]:

```
pip install apyori
```

```
Requirement already satisfied: apyori in c:\users\kunal\anaconda3\lib\site-p
ackages (1.1.2)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
import pandas as pd
from apyori import apriori
```

In [3]:

```
data = pd.read_csv('Market_Basket_Optimisation.csv', header=None)
```

In [4]:

```
data.head()
```

Out[4]:

	0	1	2	3	4	5	6	7	8	9	10
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	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

In [5]:

```
data.shape
```

Out[5]:

```
(7501, 20)
```

In [6]:

```
transactions=[]
for i in range(0,7501):
    transactions.append([str(data.values[i,j]) for j in range(0,20)])
```

In [7]:

```
association_rules = apriori(transactions, min_support=0.005, min_confidence=0.2,min_lift =
association_results = list(association_rules)
```

In [8]:

```
print(len(association_results))
```

```
38
```

In [9]:

```
association_results
```

Out[9]:

```
[RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), sup
port=0.005732568990801226, ordered_statistics=[OrderedStatistic(items_base
=frozenset({'mushroom cream sauce'}), items_add=frozenset({'escalope'}), c
onfidence=0.3006993006993007, lift=3.790832696715049)]),
 RelationRecord(items=frozenset({'pasta', 'escalope'}), support=0.00586588
4548726837, ordered_statistics=[OrderedStatistic(items_base=frozenset({'pa
sta'}), items_add=frozenset({'escalope'}), confidence=0.3728813559322034,
lift=4.700811850163794)]),
 RelationRecord(items=frozenset({'herb & pepper', 'ground beef'}), support
=0.015997866951073192, ordered_statistics=[OrderedStatistic(items_base=fro
zenset({'herb & pepper'}), items_add=frozenset({'ground beef'}), confidenc
e=0.3234501347708895, lift=3.2919938411349285)]),
 RelationRecord(items=frozenset({'tomato sauce', 'ground beef'}), support=
0.005332622317024397, ordered_statistics=[OrderedStatistic(items_base=froz
enset({'tomato sauce'}), items_add=frozenset({'ground beef'}), confidence=
0.3773584905660377, lift=3.840659481324083)]),
 RelationRecord(items=frozenset({'olive oil', 'whole wheat pasta'}), suppo
rt=0.007998933475536596, ordered_statistics=[OrderedStatistic(items_base=f
```

In [10]:

```
results = pd.DataFrame(association_results)
results.head()
```

Out[10]:

	items	support	ordered_statistics
0	(escalope, mushroom cream sauce)	0.005733	[((mushroom cream sauce), (escalope), 0.300699...
1	(pasta, escalope)	0.005866	[((pasta), (escalope), 0.3728813559322034, 4.7...
2	(herb & pepper, ground beef)	0.015998	[((herb & pepper), (ground beef), 0.3234501347...
3	(tomato sauce, ground beef)	0.005333	[((tomato sauce), (ground beef), 0.37735849056...
4	(olive oil, whole wheat pasta)	0.007999	[((whole wheat pasta), (olive oil), 0.27149321...