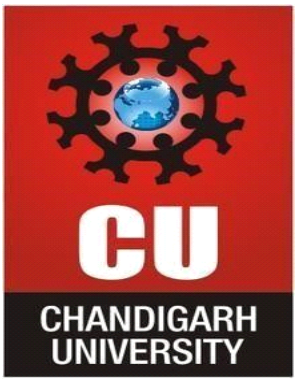
CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



Submitted By: Rajiv Paul	**Submitted To:** Parveen Badoni
Subject Name	Machine Learning Lab
Subject Code	20CSP-317
Branch	BE-CSE
Semester	5th

LAB INDEX

Sr. No	Program	Date	Evaluation				Sign
			LW (12)	VV (8)	FW (10	Total (30	
1.1							
1.2							
1.3							
2.1							
2.2							
2.3							
2.4							
3.1							
3.2							
3.3	To Implement Association Rule Mining.	02/11/22					

EXPERIMENT – 10

Student Name:Rajiv Paul **UID: 20BCS1812**
Branch:CSE **Section/Group: 20BCS_WM-702A**
Semester: 5th **Date of Performance: 26/10/2022**
Subject Name: Machine Learning Lab **Subject Code: 20CSP-317**

1. AIM OF THE EXPERIMENT:

To Implement Association Rule Mining.

2. TASK TO BE DONE:

We will implement Association Rule Mining on our dataset and then view our interpretation values using the Associan rule function.

3. PROGRAM CODE & OUTPUT:

i) Importing libraries

- import pandas as pd
- import numpy as np
- from mlxtend.frequent_patterns import apriori, association_rules
- import warnings
- warnings.filterwarnings('ignore')

```
import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori, association_rules
import warnings
warnings.filterwarnings('ignore')
```

ii) Importing dataset

- df = pd.read_csv(r'C:\Users\ABC\Desktop\Data Science\DataSets\GroceryStoreDataSet_ARM.csv', names=['products'], sep = ',')
- print(pd.DataFrame(df))

```
df = pd.read_csv(r'C:\Users\ABC\Desktop\Data Science\DataSets\GroceryStoreDataSet_ARM.csv', names=['products'], sep = ',')
print(pd.DataFrame(df))
```

```
                      products
0              MILK,BREAD,BISCUIT
1    BREAD,MILK,BISCUIT,CORNFLAKES
2              BREAD,TEA,BOURNVITA
3             JAM,MAGGI,BREAD,MILK
4                MAGGI,TEA,BISCUIT
5              BREAD,TEA,BOURNVITA
6             MAGGI,TEA,CORNFLAKES
7          MAGGI,BREAD,TEA,BISCUIT
8              JAM,MAGGI,BREAD,TEA
9                     BREAD,MILK
10  COFFEE,COCK,BISCUIT,CORNFLAKES
11  COFFEE,COCK,BISCUIT,CORNFLAKES
12          COFFEE,SUGER,BOURNVITA
13               BREAD,COFFEE,COCK
14             BREAD,SUGER,BISCUIT
15         COFFEE,SUGER,CORNFLAKES
16           BREAD,SUGER,BOURNVITA
17              BREAD,COFFEE,SUGER
18              BREAD,COFFEE,SUGER
19      TEA,MILK,COFFEE,CORNFLAKES
```

iii) Checking other parameters of dataset.

- print(df.head())
- print("\nShape of dataset: ", df.shape)

```
print(df.head())
print("\nShape of dataset: ",df.shape)
```

```
                      products
0              MILK,BREAD,BISCUIT
1    BREAD,MILK,BISCUIT,CORNFLAKES
2              BREAD,TEA,BOURNVITA
3             JAM,MAGGI,BREAD,MILK
4                MAGGI,TEA,BISCUIT

Shape of dataset:  (20, 1)
```

iv) Splitting every items in individual columns.

- data = list(df["products"].apply(lambda x:x.split(",")))
- pd.DataFrame(data)

```
data = list(df["products"].apply(lambda x:x.split(",") ))
pd.DataFrame(data)
```

	0	1	2	3
0	MILK	BREAD	BISCUIT	None
1	BREAD	MILK	BISCUIT	CORNFLAKES
2	BREAD	TEA	BOURNVITA	None
3	JAM	MAGGI	BREAD	MILK
4	MAGGI	TEA	BISCUIT	None
5	BREAD	TEA	BOURNVITA	None
6	MAGGI	TEA	CORNFLAKES	None
7	MAGGI	BREAD	TEA	BISCUIT
8	JAM	MAGGI	BREAD	TEA
9	BREAD	MILK	None	None
10	COFFEE	COCK	BISCUIT	CORNFLAKES
11	COFFEE	COCK	BISCUIT	CORNFLAKES
12	COFFEE	SUGER	BOURNVITA	None
13	BREAD	COFFEE	COCK	None
14	BREAD	SUGER	BISCUIT	None
15	COFFEE	SUGER	CORNFLAKES	None
16	BREAD	SUGER	BOURNVITA	None
17	BREAD	COFFEE	SUGER	None
18	BREAD	COFFEE	SUGER	None
19	TEA	MILK	COFFEE	CORNFLAKES

v) Transforming the list with one-hot encoding.

- from mlxtend.preprocessing import TransactionEncoder
- a = TransactionEncoder()
- a_data = a.fit(data).transform(data)
- df = pd.DataFrame(a_data,columns=a.columns_)
- df = df.replace(False,0)
- df

```
from mlxtend.preprocessing import TransactionEncoder
a = TransactionEncoder()
a_data = a.fit(data).transform(data)
df = pd.DataFrame(a_data,columns=a.columns_)
df = df.replace(False,0)
df
```

	BISCUIT	BOURNVITA	BREAD	COCK	COFFEE	CORNFLAKES	JAM	MAGGI	MILK	SUGER	TEA
0	True	0	True	0	0	0	0	0	True	0	0
1	True	0	True	0	0	True	0	0	True	0	0
2	0	True	True	0	0	0	0	0	0	0	True
3	0	0	True	0	0	0	True	True	True	0	0
4	True	0	0	0	0	0	0	True	0	0	True
5	0	True	True	0	0	0	0	0	0	0	True
6	0	0	0	0	0	True	0	True	0	0	True
7	True	0	True	0	0	0	0	True	0	0	True
8	0	0	True	0	0	0	True	True	0	0	True
9	0	0	True	0	0	0	0	0	True	0	0
10	True	0	0	True	True	True	0	0	0	0	0
11	True	0	0	True	True	True	0	0	0	0	0
12	0	True	0	0	True	0	0	0	0	True	0
13	0	0	True	True	True	0	0	0	0	0	0
14	True	0	True	0	0	0	0	0	0	True	0
15	0	0	0	0	True	True	0	0	0	True	0
16	0	True	True	0	0	0	0	0	0	True	0
17	0	0	True	0	True	0	0	0	0	True	0
18	0	0	True	0	True	0	0	0	0	True	0
19	0	0	0	0	True	True	0	0	True	0	True

vi) Setting a threshold value for the support value and calculating the support value.

- df = apriori(df, min_support = 0.2, use_colnames = True, verbose = 1)
- df

```
df = apriori(df, min_support = 0.2, use_colnames = True, verbose = 1)
df
```

```
Processing 42 combinations | Sampling itemset size 3
```

	support	itemsets
0	0.35	(BISCUIT)
1	0.2	(BOURNVITA)
2	0.65	(BREAD)
3	0.4	(COFFEE)
4	0.3	(CORNFLAKES)
5	0.25	(MAGGI)
6	0.25	(MILK)
7	0.3	(SUGER)
8	0.35	(TEA)
9	0.2	(BISCUIT, BREAD)
10	0.2	(MILK, BREAD)
11	0.2	(SUGER, BREAD)
12	0.2	(TEA, BREAD)
13	0.2	(CORNFLAKES, COFFEE)
14	0.2	(SUGER, COFFEE)
15	0.2	(MAGGI, TEA)

vii) Interpretation values using the Associan rule function.

- df_ar = association_rules(df, metric = "confidence", min_threshold = 0.6)
- df_ar

```
df_ar = association_rules(df, metric = "confidence", min_threshold = 0.6)
df_ar
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(MILK)	(BREAD)	0.25	0.65	0.2	0.800000	1.230769	0.0375	1.75
1	(SUGER)	(BREAD)	0.30	0.65	0.2	0.666667	1.025641	0.0050	1.05
2	(CORNFLAKES)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.80
3	(SUGER)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.80
4	(MAGGI)	(TEA)	0.25	0.35	0.2	0.800000	2.285714	0.1125	3.25