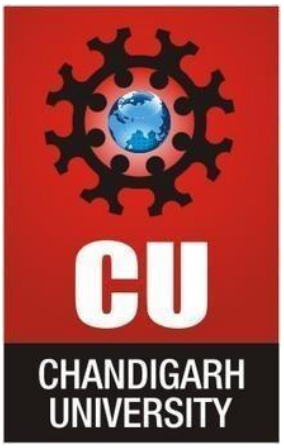
DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

CHANDIGARH UNIVERSITY
UNIVERSITY INSTITUTE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



Submitted By: Lipakshi	**Submitted To:** Navneet Kaur
Subject Name	Machine Learning Lab
Subject Code	20CSP-317
Branch	Computer Science
Semester	5th

UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CU CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Subject Name: Machine Learning Lab

Subject Code: 20CSP-317

Submitted to:

Faculty name: Navneet Kaur

Submitted by:

Name: Lipakshi

UID: 20BCS5082

Section: 607

Group: B

Ex. No	List of Experiments	Date	Conduc t (MM: 12)	Viva (MM : 10)	Record (MM: 8)	Total (MM: 30)	Remarks/Signature
1.1	Implement Exploratory Data Analysis on any data set.						
1.2	Implement Data Visualization.	23-08-2022					
1.3	Data analysis of any data set via graphs using linear regression.						
1.4	Implement support Vector machine on any data set and analyse the accuracy with logistic regression.	10-10-2022					
2.2	Implement Naive Bayes on any Data Set.	10-10-2022					
2.3							

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

2.4							
3.1	Implement K-Means	07-11-2022					
3.2	Implement PCA	07-11-2022					
3.3	Implement Association Rules Minning	07-11-2022					

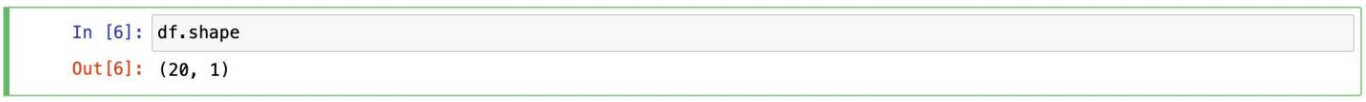
Experiment 10

Q1. Task to be done/ Which logistics used: Implement Association Rules Minning.

 # Code: import pandas as pd import numpy as np from

mlxtend.frequent_patterns import apriori, association_rules

df = pd.read_csv('GroceryStoreDataSet.csv', names = ['products'], sep = ',') df.head()



```
In [3]: import pandas as pd
        import numpy as np
        from mlxtend.frequent_patterns import apriori, association_rules
```

```
In [5]: df = pd.read_csv('GroceryStoreDataSet.csv', names = ['products'], sep = ',')
        df.head()
```

Out[5]:

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

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

```
In [7]: data = list(df["products"].apply(lambda x:x.split(",") ))
        data
```

```
Out[7]: [['MILK', 'BREAD', 'BISCUIT'],
         ['BREAD', 'MILK', 'BISCUIT', 'CORNFLAKES'],
         ['BREAD', 'TEA', 'BOURNVITA'],
         ['JAM', 'MAGGI', 'BREAD', 'MILK'],
         ['MAGGI', 'TEA', 'BISCUIT'],
         ['BREAD', 'TEA', 'BOURNVITA'],
         ['MAGGI', 'TEA', 'CORNFLAKES'],
         ['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],
         ['JAM', 'MAGGI', 'BREAD', 'TEA'],
         ['BREAD', 'MILK'],
         ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
         ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
         ['COFFEE', 'SUGER', 'BOURNVITA'],
         ['BREAD', 'COFFEE', 'COCK'],
         ['BREAD', 'SUGER', 'BISCUIT'],
         ['COFFEE', 'SUGER', 'CORNFLAKES'],
         ['BREAD', 'SUGER', 'BOURNVITA'],
         ['BREAD', 'COFFEE', 'SUGER'],
         ['BREAD', 'COFFEE', 'SUGER'],
         ['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]
```

#Let's transform 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

In [8]:
```
#Let's transform 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
```

Out[8]:

	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

#set a threshold value for the support value and calculate the support value.
df = apriori(df, min_support = 0.2, use_colnames = True, verbose = 1) df

In [9]:
```
#set a threshold value for the support value and calculate the support value.
df = apriori(df, min_support = 0.2, use_colnames = True, verbose = 1)
df
```

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

```
/Users/apple/opt/anaconda3/lib/python3.9/site-packages/mlxtend/frequent_patterns/fpcommon.py:111: DeprecationWarning: DataFrames with non-bool types result in worse computationalperformance and their support might be discontinued in the future.Please use a DataFrame with bool type
  warnings.warn(
```

Out[9]:

	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	(COFFEE, CORNFLAKES)
14	0.2	(COFFEE, SUGER)
15	0.2	(TEA, MAGGI)

#Let's view our interpretation values using the Associan rule function. df_ar = association_rules(df, metric = "confidence", min_threshold = 0.6) df_ar

```
In [10]: #Let's view our interpretation values using the Associan rule function.
         df_ar = association_rules(df, metric = "confidence", min_threshold = 0.6)
         df_ar
```

Out[10]:

	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

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1			
2			
3			
4			