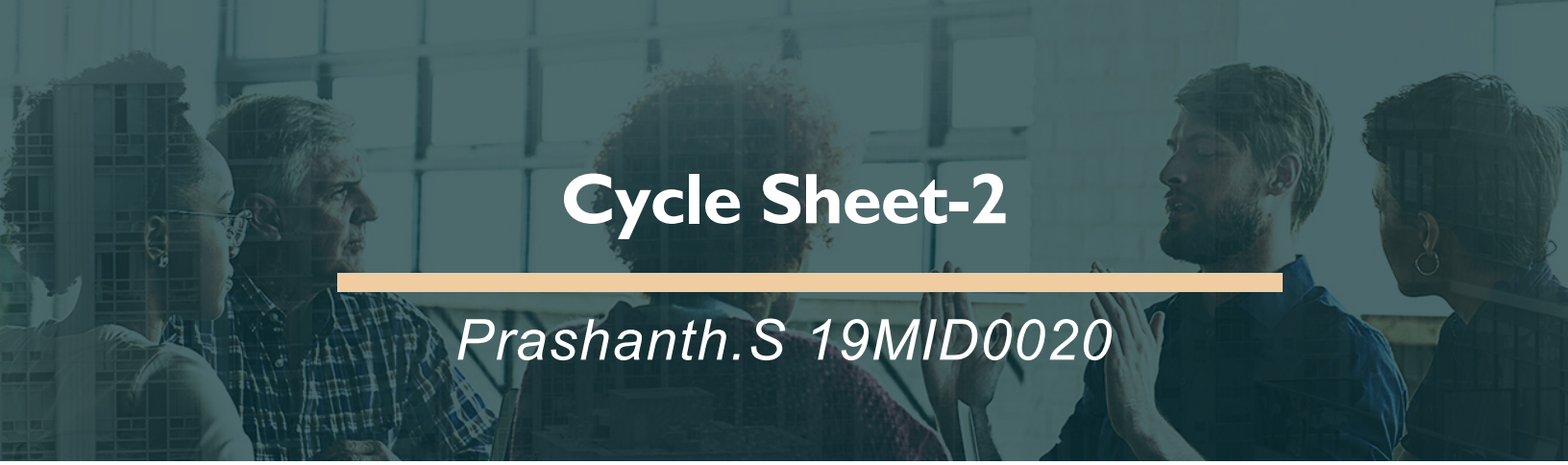


Cycle Sheet-2

Prashanth.S 19MID0020

Question-1

1. Download the employee churn dataset (15000 rows and 10 columns). Perform cluster using K-Means on this data. Identify and characterize the clusters in which the churn rate is higher. (Hint: Study the variables and select the useful ones for cluster analysis. Create clusters (how many? Analyze!). Examine the rate of churn in each cluster and discuss the characteristic of the clusters.)

What is "churn" ??

- When choosing a telecommunication service provider, customers usually have many choices. They can choose any service provider and may move away from the current provider. The percentage of customers moving out and disconnecting the service is known as "churn".
- It is very important to reduce churn for business growth and customer retention. If the churn is high, the business will continually be in search of new customers without a stable customer base.
- The performance of the business will be very unpredictable. Businesses try to keep the customers satisfied, to retain them as long as possible.
- However, in the real world, the customer churn can be as high as 25% annually in the telecommunication industry. Also, the cost of acquiring a new customer is 10 times more than the cost to retain an existing customer. This poses a serious challenge to business owners.

About the Data-set

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   satisfaction_level     14999 non-null  float64
 1   last_evaluation        14999 non-null  float64
 2   number_project         14999 non-null  int64
 3   average_montly_hours   14999 non-null  int64
 4   time_spend_company     14999 non-null  int64
 5   Work_accident          14999 non-null  int64
 6   promotion_last_5years  14999 non-null  int64
 7   Departments            14999 non-null  object
 8   salary                 14999 non-null  object
 9   left                   14999 non-null  int64
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

1)**satisfaction_level** → It is employee satisfaction point, which ranges from 0-1.

2)**last_evaluation** → It is evaluated performance by the employer, which also ranges from 0-1.

3)**number_projects** → How many numbers of projects assigned to an employee?

4)**average_monthly_hours** → How many average numbers of hours worked by an employee in a month?

5)**time_spent_company** → time_spent_company means employee experience. The number of years spent by an employee in the company.

6)**work_accident** → Whether an employee has had a work accident or not.

7)**promotion_last_5years** → Whether an employee has had a promotion in the last 5 years or not.

8)**Departments** → Employee's working department/division.

9)**Salary** → Salary level of the employee such as low, medium and high.

10)**left** →Whether the employee has left the company or not.

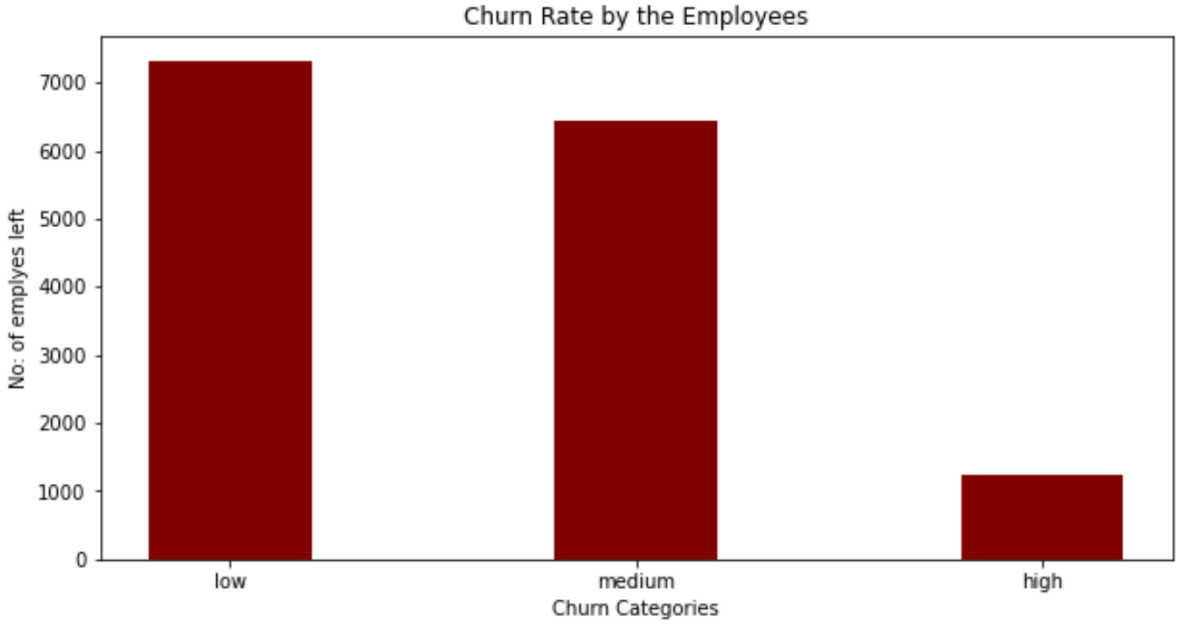
In [4]:
```
## There are no NULL values in the data-set
df.isnull().sum()
```

Out[4]:
```
satisfaction_level       0
last_evaluation          0
number_project           0
average_montly_hours     0
time_spend_company       0
Work_accident            0
promotion_last_5years    0
Departments              0
salary                   0
left                     0
dtype: int64
```

```
salary_var = dict(df['salary'].value_counts())
salary_var
```

```
{'low': 7316, 'medium': 6446, 'high': 1237}
```

```
fig = plt.figure(figsize = (10, 5))

# creating the bar plot
left_level = list(salary_var.keys())
left_count = list(salary_var.values())

plt.bar(left_level, left_count, color ='maroon',width = 0.4)

plt.xlabel("Churn Categories")
plt.ylabel("No: of emplyes left")
plt.title("Churn Rate by the Employees")
plt.show()
```



```
x = df.iloc[:,:-1]  ## in-dependent feature
y = df.iloc[:,-1]   ## dependent feature (left feature)
```

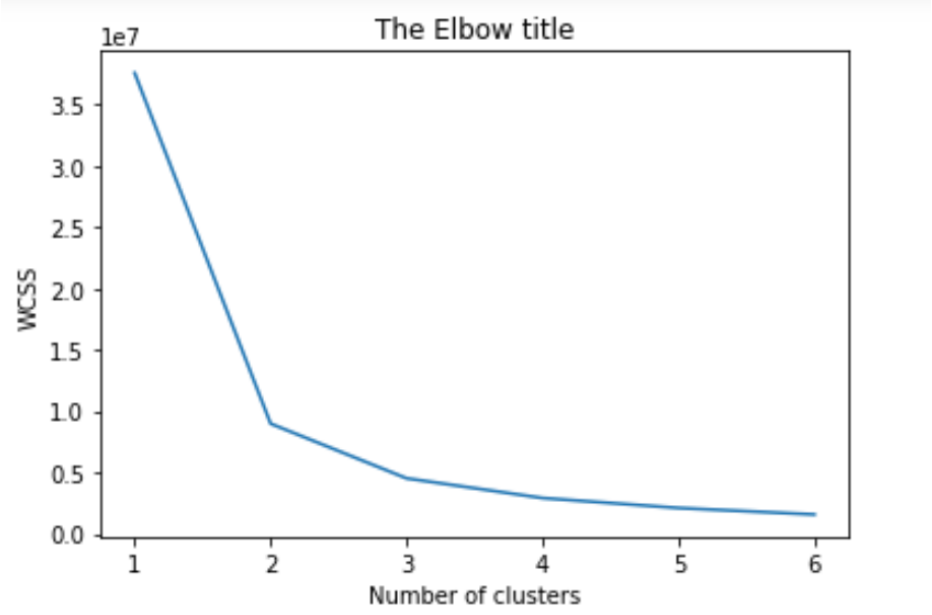
```
number_of_clusters = 3
kmeans = KMeans(number_of_clusters)
kmeans.fit(x)
```

```
KMeans(n_clusters=3)
```

```
identified_clusters = kmeans.fit_predict(x)
identified_clusters
```

```
array([2, 1, 1, ..., 2, 1, 2], dtype=int32)
```

```
wcss=[]
for i in range(1,7):
    kmeans = KMeans(i)
    kmeans.fit(x)
    wcss_iter = kmeans.inertia_
    wcss.append(wcss_iter)

number_clusters = range(1,7)
plt.plot(number_clusters,wcss)
plt.title('The Elbow title')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')

plt.show()
```



From the elbow method, it is found that the number of clusters = 2.

CYCLE SHEET-2

Prashanth.S 19MID0020

Data-Set loaded into Weka

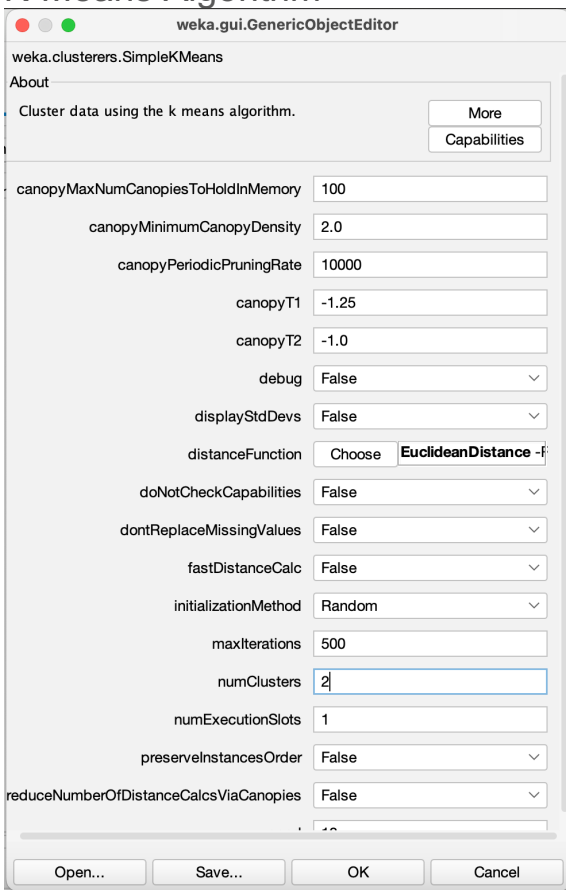
Viewer

Relation: 1_HR_comma_sep

No.	1: satisfaction_level Numeric	2: last_evaluation Numeric	3: number_project Numeric	4: average_montly_hours Numeric	5: time_spend_company Numeric	6: Work_accident Numeric	7: promotion_last_5years Numeric	8: Departments Nominal	9: salary Nominal	10: left Nominal
1	0.38	0.53	2.0	157.0	3.0	0.0	0.0	sales	low	Yes
2	0.8	0.86	5.0	262.0	6.0	0.0	0.0	sales	medium	Yes
3	0.11	0.88	7.0	272.0	4.0	0.0	0.0	sales	medium	Yes
4	0.72	0.87	5.0	223.0	5.0	0.0	0.0	sales	low	Yes
5	0.37	0.52	2.0	159.0	3.0	0.0	0.0	sales	low	Yes
6	0.41	0.5	2.0	153.0	3.0	0.0	0.0	sales	low	Yes
7	0.1	0.77	6.0	247.0	4.0	0.0	0.0	sales	low	Yes
8	0.92	0.85	5.0	259.0	5.0	0.0	0.0	sales	low	Yes
9	0.89	1.0	5.0	224.0	5.0	0.0	0.0	sales	low	Yes
10	0.42	0.53	2.0	142.0	3.0	0.0	0.0	sales	low	Yes
11	0.45	0.54	2.0	135.0	3.0	0.0	0.0	sales	low	Yes
12	0.11	0.81	6.0	305.0	4.0	0.0	0.0	sales	low	Yes
13	0.84	0.92	4.0	234.0	5.0	0.0	0.0	sales	low	Yes
14	0.41	0.55	2.0	148.0	3.0	0.0	0.0	sales	low	Yes
15	0.36	0.56	2.0	137.0	3.0	0.0	0.0	sales	low	Yes
16	0.38	0.54	2.0	143.0	3.0	0.0	0.0	sales	low	Yes
17	0.45	0.47	2.0	160.0	3.0	0.0	0.0	sales	low	Yes
18	0.78	0.99	4.0	255.0	6.0	0.0	0.0	sales	low	Yes
19	0.45	0.51	2.0	160.0	3.0	1.0	1.0	sales	low	Yes
20	0.76	0.89	5.0	262.0	5.0	0.0	0.0	sales	low	Yes
21	0.11	0.83	6.0	282.0	4.0	0.0	0.0	sales	low	Yes
22	0.38	0.55	2.0	147.0	3.0	0.0	0.0	sales	low	Yes
23	0.09	0.95	6.0	304.0	4.0	0.0	0.0	sales	low	Yes
24	0.46	0.57	2.0	139.0	3.0	0.0	0.0	sales	low	Yes
25	0.4	0.53	2.0	158.0	3.0	0.0	0.0	sales	low	Yes
26	0.89	0.92	5.0	242.0	5.0	0.0	0.0	sales	low	Yes
27	0.82	0.87	4.0	239.0	5.0	0.0	0.0	sales	low	Yes
28	0.4	0.49	2.0	135.0	3.0	0.0	0.0	sales	low	Yes
29	0.41	0.46	2.0	128.0	3.0	0.0	0.0	accounting	low	Yes
30	0.38	0.5	2.0	132.0	3.0	0.0	0.0	accounting	low	Yes
31	0.09	0.62	6.0	294.0	4.0	0.0	0.0	accounting	low	Yes
32	0.45	0.57	2.0	134.0	3.0	0.0	0.0	hr	low	Yes
33	0.4	0.51	2.0	145.0	3.0	0.0	0.0	hr	low	Yes

Add instance | Undo | OK | Cancel

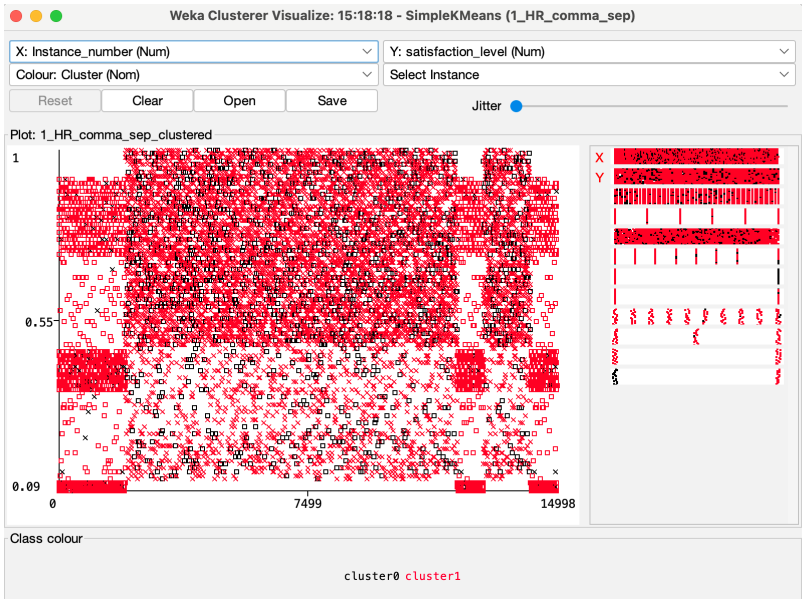
K-Means Algorithm



Number of Clusters=2

15:18:18 - SimpleKMeans

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -
Relation:     1_HR_comma_sep
Instances:    14999
Attributes:   10
              satisfaction_level
              last_evaluation
              number_project
              average_montly_hours
              time_spend_company
              Work_accident
              promotion_last_5years
              Departments
              salary
Ignored:
              left
Test mode:    Classes to clusters evaluation on training data

=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 33
Within cluster sum of squared errors: 23270.068360804027

Initial starting points (random):

Cluster 0: 0.8,0.38,3,215,6,0,0,support,low
Cluster 1: 0.4,0.57,2,151,3,0,0,support,low

Missing values globally replaced with mean/mode

Final cluster centroids:
                                Cluster#
Attribute               Full Data          0          1
                        (14999.0)  (2169.0) (12830.0)
=============================================================
satisfaction_level         0.6128     0.6483     0.6068
last_evaluation            0.7161     0.7131     0.7166
number_project             3.8031     3.7888     3.8055
average_montly_hours     201.0503   199.8183   201.2586
time_spend_company         3.4982     3.5058      3.497
Work_accident              0.1446          1          0
promotion_last_5years      0.0213      0.035     0.0189
Departments                 sales      sales      sales
salary                        low        low        low



Time taken to build model (full training data) : 0.14 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       2169 ( 14%)
1      12830 ( 86%)


Class attribute: left
Classes to Clusters:

    0    1  <-- assigned to cluster
  169 3402 | Yes
 2000 9428 | No

Cluster 0 <-- Yes
Cluster 1 <-- No

Incorrectly clustered instances :       5402.0   36.0157 %
```



Number of Clusters=3

15:52:09 - SimpleKMeans

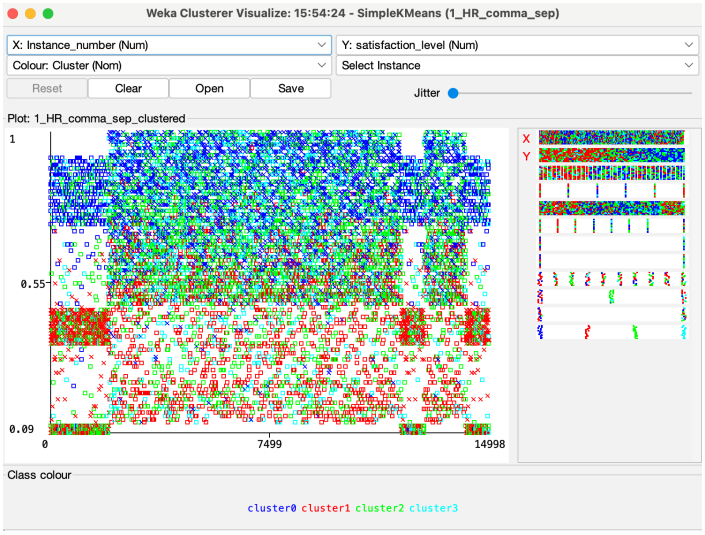
```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -
Relation:     1_HR_comma_sep
Instances:    14999
Attributes:   10
              satisfaction_level
              last_evaluation
              number_project
              average_montly_hours
              time_spend_company
              Work_accident
              promotion_last_5years
              Departments
              salary
Ignored:
              left
Test mode:    Classes to clusters evaluation on training data

=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 38
Within cluster sum of squared errors: 17623.195098110078

Initial starting points (random):

Cluster 0: 0.8,0.38,3,215,6,0,0,support,low
Cluster 1: 0.4,0.57,2,151,3,0,0,support,low
Cluster 2: 0.65,0.98,3,252,2,0,0,product_mng,high

Missing values globally replaced with mean/mode

Final cluster centroids:
                                Cluster#
Attribute               Full Data          0          1          2
                        (14999.0)   (1232.0)   (7034.0)   (6733.0)
=============================================================================
satisfaction_level         0.6128     0.6372     0.5914     0.6307
last_evaluation            0.7161      0.712     0.7135     0.7195
number_project             3.8031     3.7857     3.7924     3.8173
average_montly_hours     201.0503   199.0779   200.9498   201.5163
time_spend_company         3.4982     3.4432     3.4315      3.578
Work_accident              0.1446          1          0     0.1392
promotion_last_5years      0.0213     0.0268      0.007     0.0352
Departments                 sales      sales      sales      sales
salary                        low        low        low     medium



Time taken to build model (full training data) : 0.09 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       1232 (  8%)
1       7034 ( 47%)
2       6733 ( 45%)


Class attribute: left
Classes to Clusters:

    0    1    2  <-- assigned to cluster
   95 2147 1329 | Yes
 1137 4887 5404 | No

Cluster 0 <-- No class
Cluster 1 <-- Yes
Cluster 2 <-- No

Incorrectly clustered instances :       7448.0   49.6566 %
```

CYCLE SHEET-2

Prashanth.S 19MID0020

Number of Clusters=4

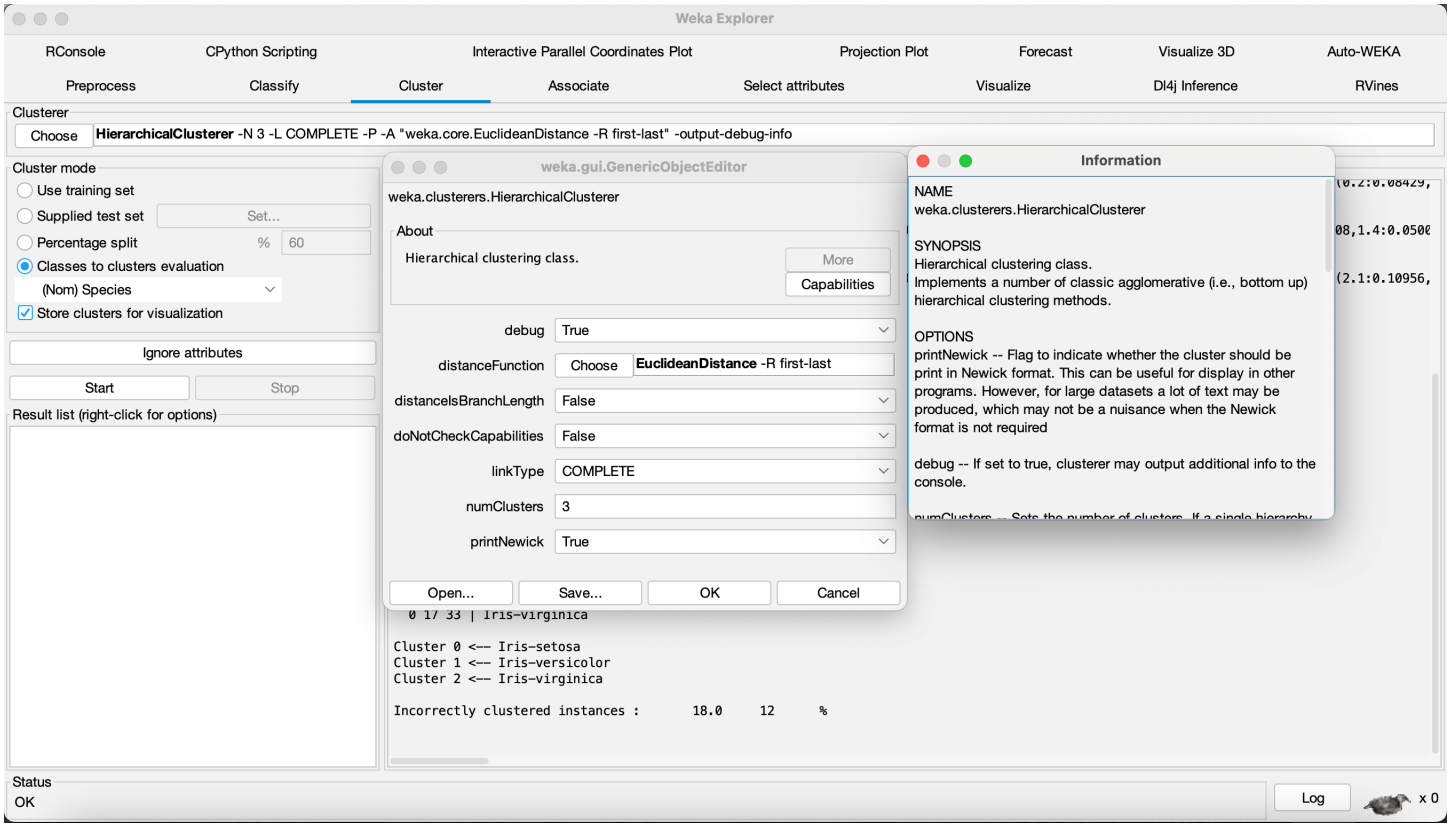
```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 4 -A "weka.core.EuclideanDistance -R first-last" -I 500 -
Relation:     1_HR_comma_sep
Instances:    14999
Attributes:   10
              satisfaction_level
              last_evaluation
              number_project
              average_montly_hours
              time_spend_company
              Work_accident
              promotion_last_5years
              Departments
              salary
Ignored:
              left
Test mode:    Classes to clusters evaluation on training data

=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 21
Within cluster sum of squared errors: 17056.39448699175

Initial starting points (random):

Cluster 0: 0.8,0.38,3,215,6,0,0,support,low
Cluster 1: 0.4,0.57,2,151,3,0,0,support,low
Cluster 2: 0.65,0.98,3,252,2,0,0,product_mng,high
Cluster 3: 0.7,0.65,5,202,3,1,0,technical,medium

Missing values globally replaced with mean/mode

Final cluster centroids:
                                Cluster#
Attribute               Full Data          0          1          2          3
                        (14999.0)  (4610.0)  (3697.0)  (4414.0)  (2278.0)
=============================================================================
satisfaction_level         0.6128     0.7884     0.3946     0.6008     0.6351
last_evaluation            0.7161     0.7932     0.6106     0.7237     0.7165
number_project             3.8031     3.9124     3.5055     3.9257     3.827
average_montly_hours     201.0503   211.3555   183.0928   205.0061   201.6743
time_spend_company         3.4982     3.4477     3.4907     3.5517     3.5092
Work_accident              0.1446     0.1534     0.0457     0.0401     0.4899
Departments                 sales      sales      sales    support technical
salary                        low        low        low     medium     medium



Time taken to build model (full training data) : 0.07 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       4610 ( 31%)
1       3697 ( 25%)
2       4414 ( 29%)
3       2278 ( 15%)


Class attribute: left
Classes to Clusters:

    0    1    2    3  <-- assigned to cluster
  674 1620  883  394 | Yes
 3936 2077 3531 1884 | No

Cluster 0 <-- No
Cluster 1 <-- Yes
Cluster 2 <-- No class
Cluster 3 <-- No class

Incorrectly clustered instances :       9443.0   62.9575 %
```



Question-2

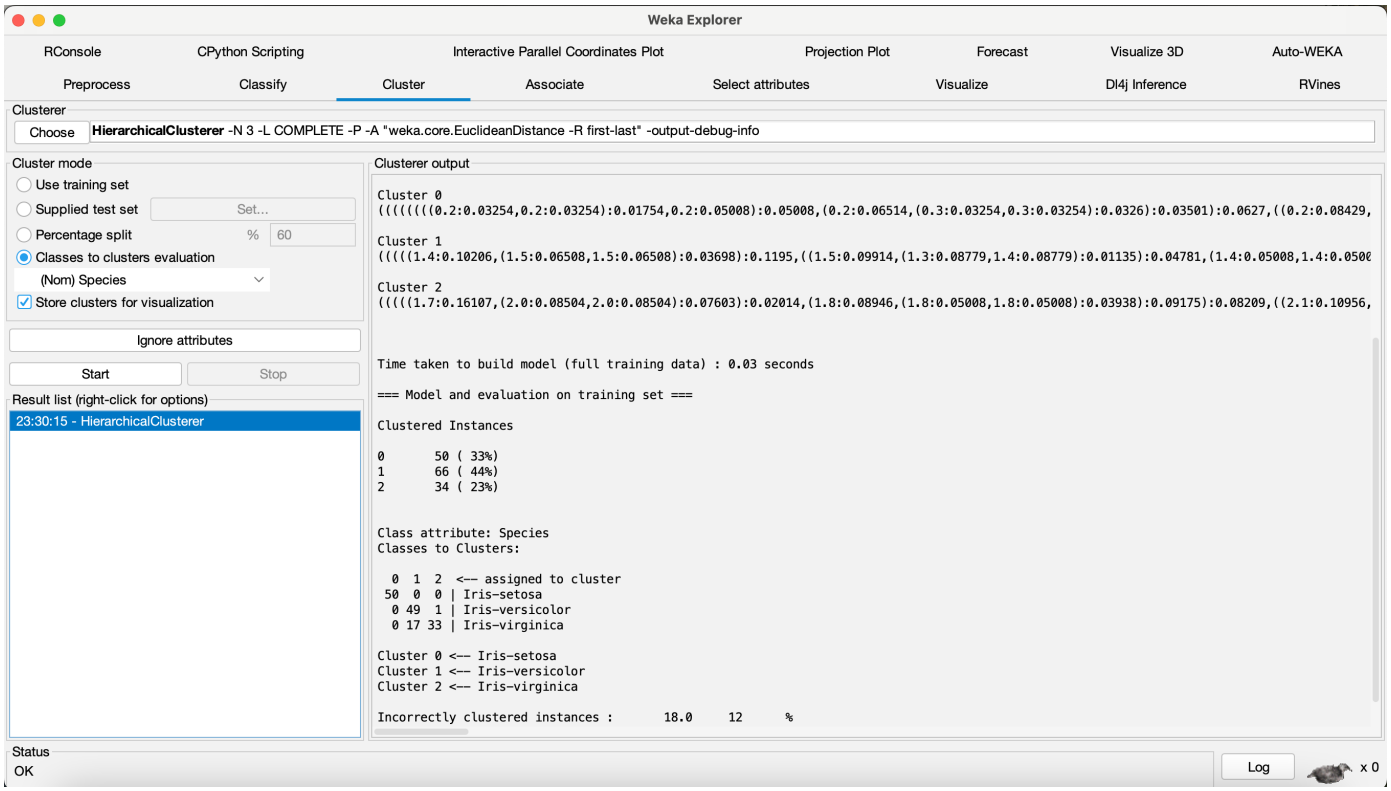
2. Consider the any one suitable dataset available in UCI. Find clusters in this dataset using Hierarchical clustering technique having the following properties.

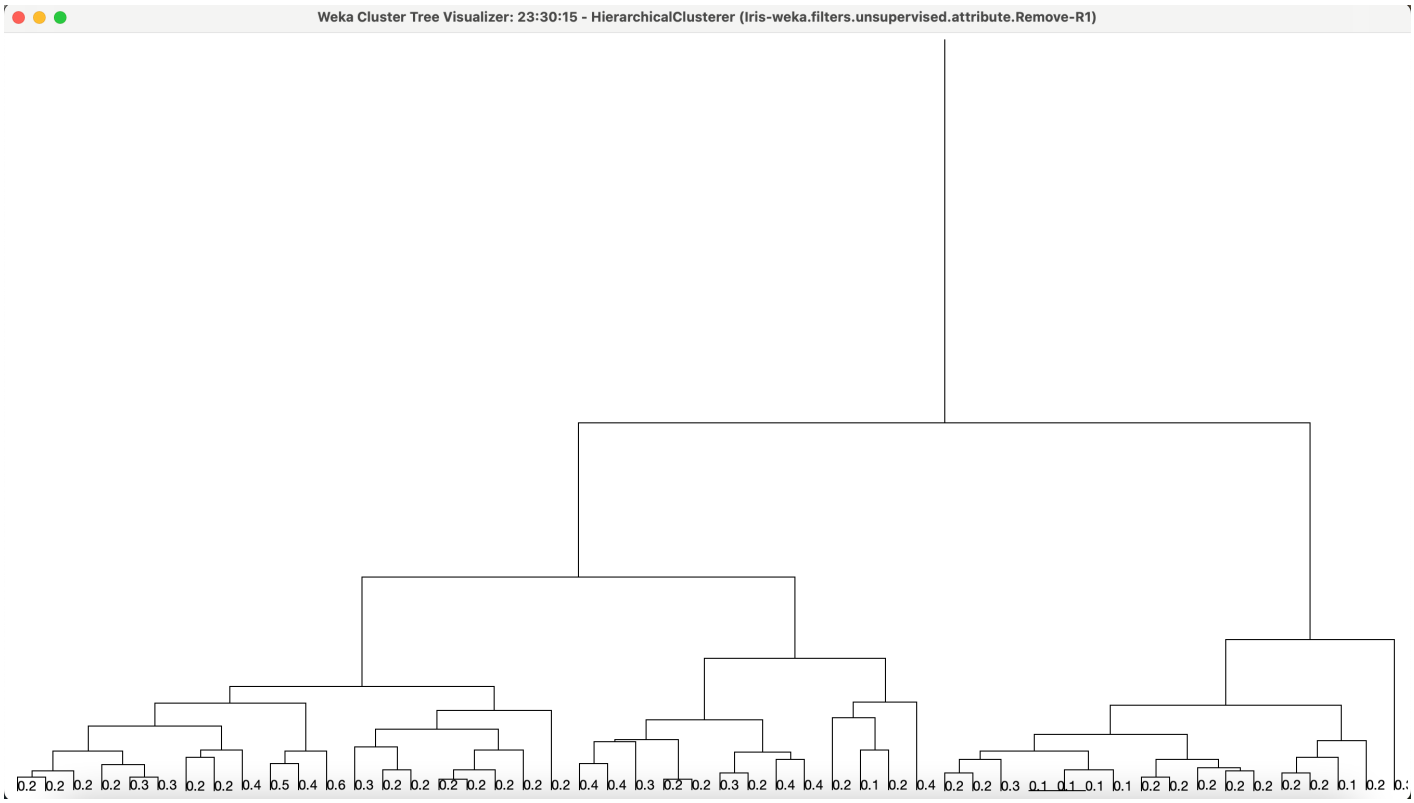
Number of clusters: 3
Distance measure: Euclidean distance.
List the members of each cluster.
Draw the Cluster tree diagram



CYCLE SHEET-2

Prashanth.S 19MID0020





Question-3

3. Download the wholesale customer dataset from UCI ML. The data set refers to clients of a wholesale distributor. It includes the annual spending in monetary units on diverse product categories. Segment the customers (find clusters in the dataset using DBSCAN clustering algorithm) for possible offers. Analyse the noise points in the dataset

About the Data-set

Attribute Information:

1) FRESH: annual spending (m.u.) on fresh products (Continuous);
2) MILK: annual spending (m.u.) on milk products (Continuous);
3) GROCERY: annual spending (m.u.)on grocery products (Continuous);
4) FROZEN: annual spending (m.u.)on frozen products (Continuous)
5) DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Continuous)
6) DELICATESSEN: annual spending (m.u.)on and delicatessen products (Continuous);
7) CHANNEL: customersâ€™ Channel - Horeca (Hotel/Restaurant/CafÃ©) or Retail channel (Nominal)
8) REGION: customersâ€™ Region â€“ Lisnon, Oporto or Other (Nominal)
Descriptive Statistics:

(Minimum, Maximum, Mean, Std. Deviation)
FRESH (3, 112151, 12000.30, 12647.329)
MILK (55, 73498, 5796.27, 7380.377)
GROCERY (3, 92780, 7951.28, 9503.163)
FROZEN (25, 60869, 3071.93, 4854.673)
DETERGENTS_PAPER (3, 40827, 2881.49, 4767.854)
DELICATESSEN (3, 47943, 1524.87, 2820.106)

REGION Frequency
Lisbon 77
Oporto 47
Other Region 316
Total 440

CHANNEL Frequency
Horeca 298
Retail 142
Total 440

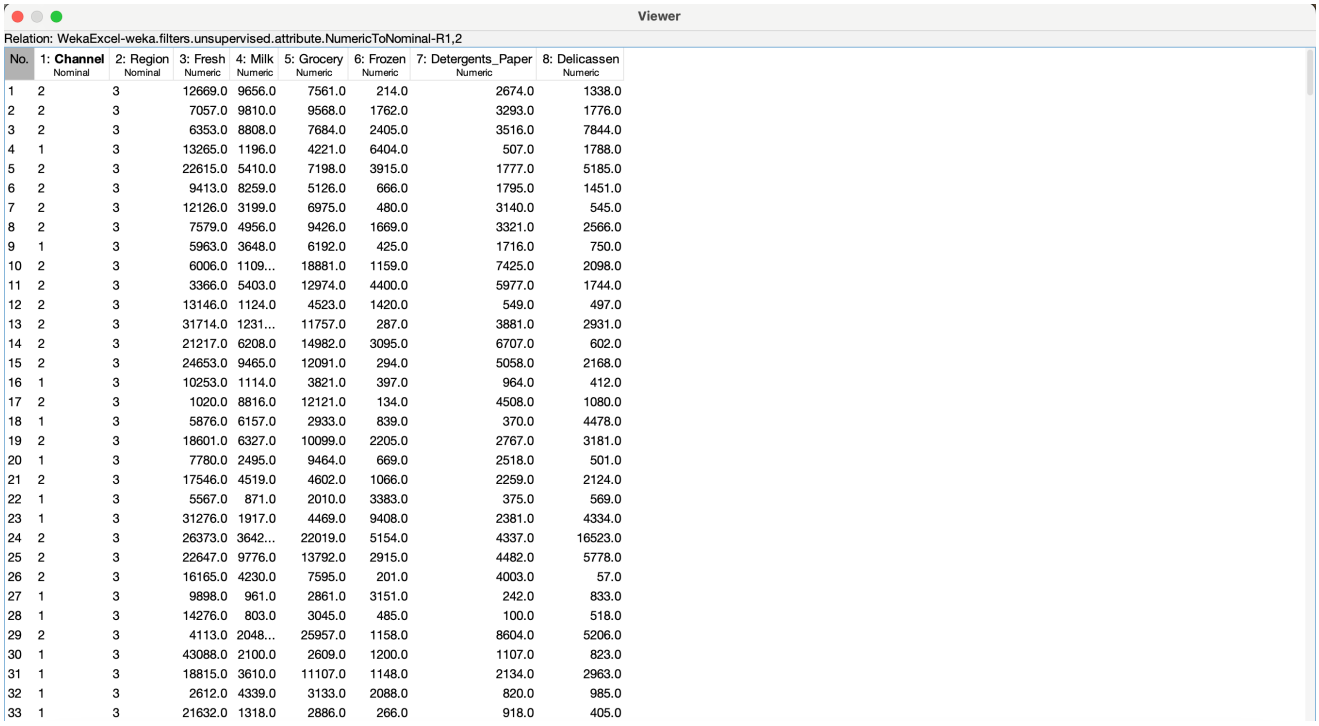
```
df3 = pd.read_excel('3_Wholesale customers data.xls')
df3.head()
```

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185

```
df3.info()
```

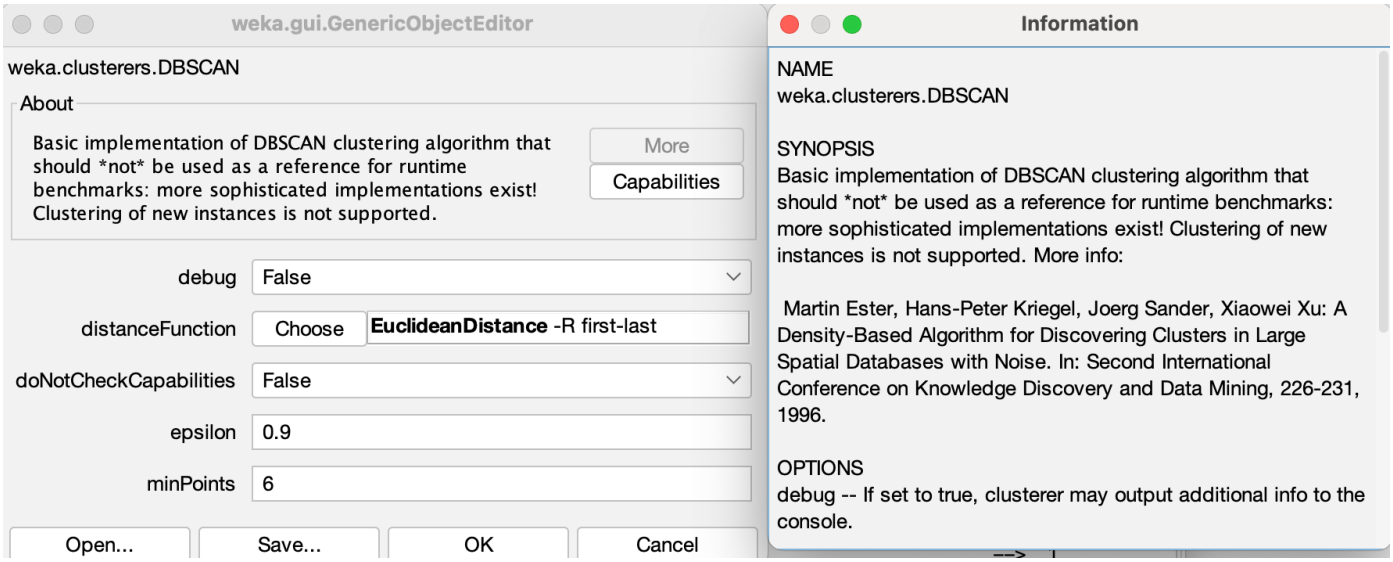
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440 entries, 0 to 439
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Channel           440 non-null    int64
 1   Region            440 non-null    int64
 2   Fresh             440 non-null    int64
 3   Milk              440 non-null    int64
 4   Grocery           440 non-null    int64
 5   Frozen            440 non-null    int64
 6   Detergents_Paper  440 non-null    int64
 7   Delicassen        440 non-null    int64
dtypes: int64(8)
memory usage: 27.6 KB
```

Data-Set loaded into Weka

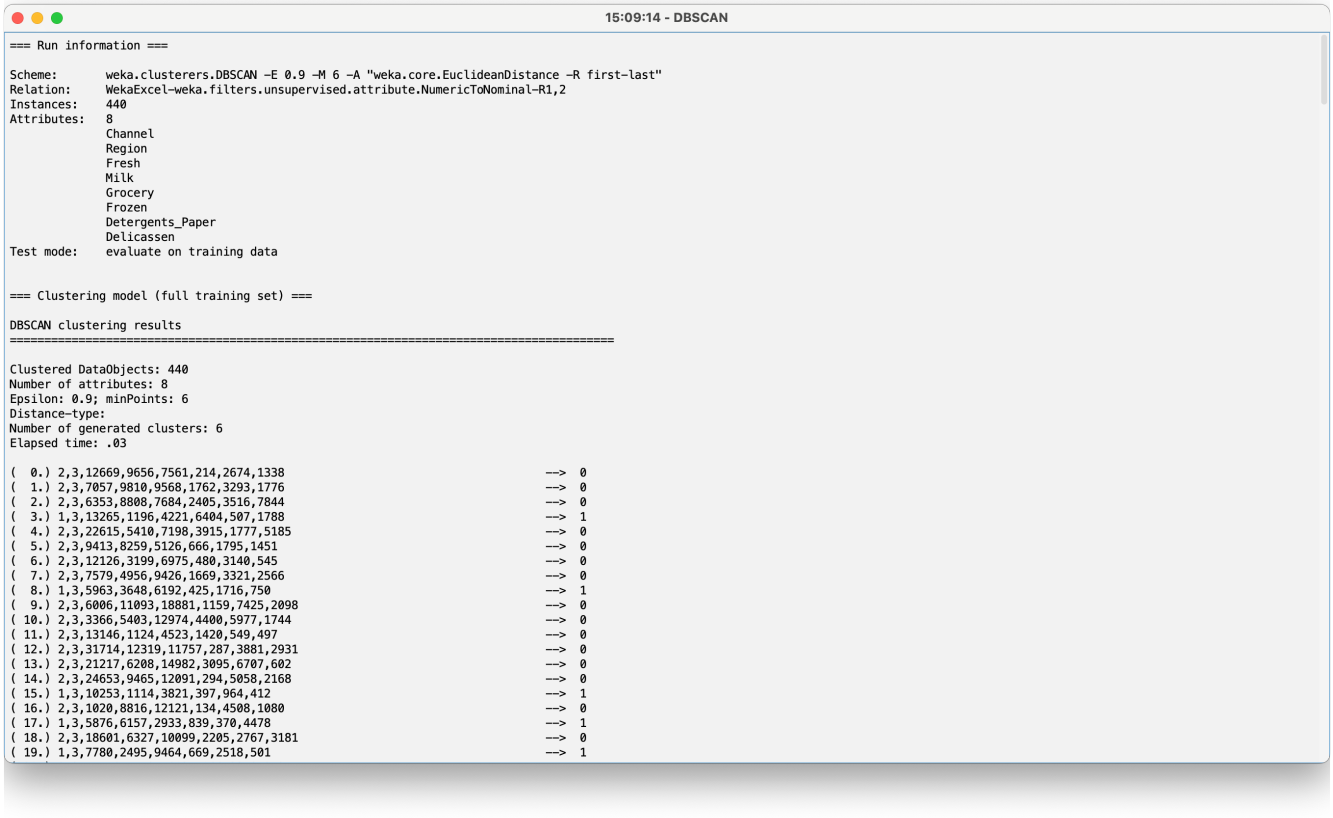


No.	1: Channel Nominal	2: Region Nominal	3: Fresh Numeric	4: Milk Numeric	5: Grocery Numeric	6: Frozen Numeric	7: Detergents_Paper Numeric	8: Delicassen Numeric
1	2	3	12669.0	9656.0	7561.0	214.0	2674.0	1338.0
2	2	3	7057.0	9810.0	9568.0	1762.0	3293.0	1776.0
3	2	3	6353.0	8808.0	7684.0	2405.0	3516.0	7844.0
4	1	3	13265.0	1196.0	4221.0	6404.0	507.0	1788.0
5	2	3	22615.0	5410.0	7198.0	3915.0	1777.0	5185.0
6	2	3	9413.0	8259.0	5126.0	666.0	1795.0	1451.0
7	2	3	12126.0	3199.0	6975.0	480.0	3140.0	545.0
8	2	3	7579.0	4956.0	9426.0	1669.0	3321.0	2566.0
9	1	3	5963.0	3648.0	6192.0	425.0	1716.0	750.0
10	2	3	6006.0	1109...	18881.0	1159.0	7425.0	2098.0
11	2	3	3366.0	5403.0	12974.0	4400.0	5977.0	1744.0
12	2	3	13146.0	1124.0	4523.0	1420.0	549.0	497.0
13	2	3	31714.0	1231...	11757.0	287.0	3881.0	2931.0
14	2	3	21217.0	6208.0	14982.0	3095.0	6707.0	602.0
15	2	3	24653.0	9465.0	12091.0	294.0	5058.0	2168.0
16	1	3	10253.0	1114.0	3821.0	397.0	964.0	412.0
17	2	3	1020.0	8816.0	12121.0	134.0	4508.0	1080.0
18	1	3	5876.0	6157.0	2933.0	839.0	370.0	4478.0
19	2	3	18601.0	6327.0	10099.0	2205.0	2767.0	3181.0
20	1	3	7780.0	2495.0	9464.0	669.0	2518.0	501.0
21	2	3	17546.0	4519.0	4602.0	1066.0	2259.0	2124.0
22	1	3	5567.0	871.0	2010.0	3383.0	375.0	569.0
23	1	3	31276.0	1917.0	4469.0	9408.0	2381.0	4334.0
24	2	3	26373.0	3642...	22019.0	5154.0	4337.0	16523.0
25	2	3	22647.0	9776.0	13792.0	2915.0	4482.0	5778.0
26	2	3	16165.0	4230.0	7595.0	201.0	4003.0	57.0
27	1	3	9898.0	961.0	2861.0	3151.0	242.0	833.0
28	1	3	14276.0	803.0	3045.0	485.0	100.0	518.0
29	2	3	4113.0	2048...	25957.0	1158.0	8604.0	5206.0
30	1	3	43088.0	2100.0	2609.0	1200.0	1107.0	823.0
31	1	3	18815.0	3610.0	11107.0	1148.0	2134.0	2963.0
32	1	3	2612.0	4339.0	3133.0	2088.0	820.0	985.0
33	1	3	21632.0	1318.0	2886.0	266.0	918.0	405.0

DB-Scan Function



Output



15:09:14 - DBSCAN

```
( 20.) 2,3,17546,4519,4602,1066,2259,2124                    -->  0
( 21.) 1,3,5567,871,2010,3383,375,569                        -->  1
( 22.) 1,3,31276,1917,4469,9408,2381,4334                    -->  1
( 23.) 2,3,26373,36423,22019,5154,4337,16523                 -->  0
( 24.) 2,3,22647,9776,13792,2915,4482,5778                   -->  0
( 25.) 2,3,16165,4230,7595,201,4003,57                       -->  0
( 26.) 1,3,9898,961,2861,3151,242,833                        -->  1
( 27.) 1,3,14276,803,3045,485,100,518                        -->  1
( 28.) 2,3,4113,20484,25957,1158,8604,5206                   -->  0
( 29.) 1,3,43088,2100,2609,1200,1107,823                     -->  1
( 30.) 1,3,18815,3610,11107,1148,2134,2963                   -->  1
( 31.) 1,3,2612,4339,3133,2088,820,985                       -->  1
( 32.) 1,3,21632,1318,2886,266,918,405                       -->  1
( 33.) 1,3,29729,4786,7326,6130,361,1083                     -->  1
( 34.) 1,3,1502,1979,2262,425,483,395                        -->  1
( 35.) 2,3,688,5491,11091,833,4239,436                       -->  0
( 36.) 1,3,29955,4362,5428,1729,862,4626                     -->  1
( 37.) 2,3,15168,10556,12477,1920,6506,714                   -->  0
( 38.) 2,3,4591,15729,16709,33,6956,433                      -->  0
( 39.) 1,3,56159,555,902,10002,212,2916                      -->  1
( 40.) 1,3,24025,4332,4757,9510,1145,5864                    -->  1
( 41.) 1,3,19176,3065,5956,2033,2575,2802                    -->  1
( 42.) 2,3,10850,7555,14961,188,6899,46                      -->  0
( 43.) 2,3,630,11095,23998,787,9529,72                       -->  0
( 44.) 2,3,9670,7027,10471,541,4618,65                       -->  0
( 45.) 2,3,5181,22044,21531,1740,7353,4985                   -->  0
( 46.) 2,3,3103,14069,21955,1668,6792,1452                   -->  0
( 47.) 2,3,44466,54259,55571,7782,24171,6465                 -->  0
( 48.) 2,3,11519,6152,10868,584,5121,1476                    -->  0
( 49.) 2,3,4967,21412,28921,1798,13583,1163                  -->  0
( 50.) 1,3,6269,1095,1980,3860,609,2162                      -->  1
( 51.) 1,3,3347,4051,6996,239,1538,301                       -->  1
( 52.) 2,3,40721,3916,5876,532,2587,1278                     -->  0
( 53.) 2,3,491,10473,11532,744,5611,224                      -->  0
( 54.) 1,3,27329,1449,1947,2436,204,1333                     -->  1
( 55.) 1,3,5264,3683,5005,1057,2024,1130                     -->  1
( 56.) 2,3,4098,29892,26866,2616,17740,1340                  -->  0
( 57.) 2,3,5417,9933,10487,38,7572,1282                      -->  0
( 58.) 1,3,13779,1970,1648,596,227,436                       -->  1
( 59.) 1,3,6137,5360,8040,129,3084,1603                      -->  1
( 60.) 2,3,8590,3045,7854,96,4095,225                        -->  0
( 61.) 2,3,35942,38369,59598,3254,26701,2017                 -->  0
( 62.) 2,3,7823,6245,6544,4154,4074,964                      -->  0
( 63.) 2,3,9396,11601,15775,2896,7677,1295                   -->  0
( 64.) 1,3,4760,1227,3250,3724,1247,1145                     -->  1
( 65.) 2,3,85,20959,45828,36,24231,1423                      -->  0
( 66.) 1,3,9,1534,7417,175,3468,27                           -->  1
( 67.) 2,3,19913,6759,13462,1256,5141,834                    -->  0
( 68.) 1,3,2446,7260,3993,5870,788,3095                      -->  1
```

15:09:14 - DBSCAN

```
(409.) 1,3,8708,3634,6100,2349,2123,5137                     -->  1
(410.) 1,3,6633,2096,4563,1389,1860,1892                     -->  1
(411.) 1,3,2126,3289,3281,1535,235,4365                      -->  1
(412.) 1,3,97,3605,12400,98,2970,62                          -->  1
(413.) 1,3,4983,4859,6633,17866,912,2435                     -->  1
(414.) 1,3,5969,1990,3417,5679,1135,290                      -->  1
(415.) 2,3,7842,6046,8552,1691,3540,1874                     -->  0
(416.) 2,3,4389,10940,10908,848,6728,993                     -->  0
(417.) 1,3,5065,5499,11055,364,3485,1063                     -->  1
(418.) 2,3,660,8494,18622,133,6740,776                       -->  0
(419.) 1,3,8861,3783,2223,633,1580,1521                      -->  1
(420.) 1,3,4456,5266,13227,25,6818,1393                      -->  1
(421.) 2,3,17063,4847,9053,1031,3415,1784                    -->  0
(422.) 1,3,26400,1377,4172,830,948,1218                      -->  1
(423.) 2,3,17565,3686,4657,1059,1803,668                     -->  0
(424.) 2,3,16980,2884,12232,874,3213,249                     -->  0
(425.) 1,3,11243,2408,2593,15348,108,1886                    -->  1
(426.) 1,3,13134,9347,14316,3141,5079,1894                   -->  1
(427.) 1,3,31012,16687,5429,15082,439,1163                   -->  1
(428.) 1,3,3047,5970,4910,2198,850,317                       -->  1
(429.) 1,3,8607,1750,3580,47,84,2501                         -->  1
(430.) 1,3,3097,4230,16483,575,241,2080                      -->  1
(431.) 1,3,8533,5506,5160,13486,1377,1498                    -->  1
(432.) 1,3,21117,1162,4754,269,1328,395                      -->  1
(433.) 1,3,1982,3218,1493,1541,356,1449                      -->  1
(434.) 1,3,16731,3922,7994,688,2371,838                      -->  1
(435.) 1,3,29703,12051,16027,13135,182,2204                  -->  1
(436.) 1,3,39228,1431,764,4510,93,2346                       -->  1
(437.) 2,3,14531,15488,30243,437,14841,1867                  -->  0
(438.) 1,3,10290,1981,2232,1038,168,2125                     -->  1
(439.) 1,3,2787,1698,2510,65,477,52                          -->  1


Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      105 ( 24%)
1      210 ( 48%)
2       59 ( 13%)
3       18 (  4%)
4       19 (  4%)
5       28 (  6%)

Unclustered instances : 1
```

Question-4

4. Use the Apriori technique to determine the association rules in the AdultUCI dataset. Choose education, marital-status, occupation, race, sex and native-country as the parameters of the given dataset for analysis. Perform several iterations to determine the min_support that can be chosen if the number of association rules that can be considered for further analysis is between 25 and 30.AdultUCI dataset can be downloaded from web or construct the dataset

About the Data-set

Data Set Information:

Extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0))

Prediction task is to determine whether a person makes over 50K a year.

Attribute Information:

Listing of attributes:

>50K, <=50K.

age: continuous.
workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt: continuous.
education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: continuous.
marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex: Female, Male.
capital-gain: continuous.
capital-loss: continuous.
hours-per-week: continuous.
native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Apriori functions

delta → Iteratively Decreasing the support by this factor.

lowerBoundMinSupport → Lower Bound for Minimum Support (10%)

upperBoundMinSupport → Upper Bound for Minimum Support (90%)

minMetric → Confidence > 90%

numRules → Number of Rules to be generated

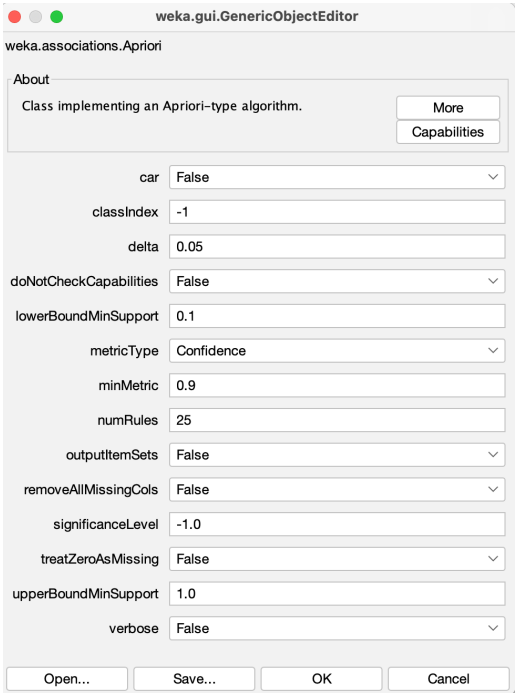
outputItemSets → This will display the item-sets with at-least support of 2.

L(1), these are the item-sets with 1 item.

L(2), these are the item-sets with 2 items.

CYCLE SHEET-2

Prashanth.S 19MID0020




```
=== Run information ===

Scheme:       weka.associations.Apriori -N 25 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:     adult-weka.filters.unsupervised.attribute.Remove-R3,5
Instances:    7981
Attributes:   13
              age
              workclass
              education
              marital-status
              occupation
              relationship
              race
              sex
              capitalgain
              capitalloss
              hoursperweek
              native-country
              class
=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.65 (5188 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 7

Generated sets of large itemsets:

Size of set of large itemsets L(1): 7

Size of set of large itemsets L(2): 10

Size of set of large itemsets L(3): 6

Size of set of large itemsets L(4): 1

Best rules found:

 1. class=<=50K 6075 ==> capitalloss=0 5885    <conf:(0.97)> lift:(1.02) lev:(0.01) [102] conv:(1.53)
 2. native-country=United-States class=<=50K 5392 ==> capitalloss=0 5223    <conf:(0.97)> lift:(1.02) lev:(0.01) [90] conv:(1.53)
 3. capitalgain=0 class=<=50K 5821 ==> capitalloss=0 5631    <conf:(0.97)> lift:(1.02) lev:(0.01) [90] conv:(1.47)
 4. class=<=50K 6075 ==> capitalgain=0 5821    <conf:(0.96)> lift:(1.04) lev:(0.03) [243] conv:(1.95)
 5. capitalloss=0 class=<=50K 5885 ==> capitalgain=0 5631    <conf:(0.96)> lift:(1.04) lev:(0.03) [228] conv:(1.89)
 6. workclass=Private 5539 ==> capitalloss=0 5297    <conf:(0.96)> lift:(1) lev:(0) [24] conv:(1.1)
 7. native-country=United-States 7124 ==> capitalloss=0 6781    <conf:(0.95)> lift:(1) lev:(-0) [0] conv:(1)
 8. race=White 6823 ==> capitalloss=0 6481    <conf:(0.95)> lift:(1) lev:(-0) [-13] conv:(0.96)
 9. race=White native-country=United-States 6242 ==> capitalloss=0 5920    <conf:(0.95)> lift:(1) lev:(-0) [-21] conv:(0.93)
10. capitalgain=0 7327 ==> capitalloss=0 6943    <conf:(0.95)> lift:(1) lev:(-0) [-31] conv:(0.92)
11. capitalgain=0 native-country=United-States 6533 ==> capitalloss=0 6190    <conf:(0.95)> lift:(1) lev:(-0) [-28] conv:(0.91)
12. race=White capitalgain=0 6228 ==> capitalloss=0 5886    <conf:(0.95)> lift:(0.99) lev:(-0.01) [-42] conv:(0.87)
13. race=White capitalgain=0 native-country=United-States 5695 ==> capitalloss=0 5373    <conf:(0.94)> lift:(0.99) lev:(-0.01) [-47] conv:(0.85)
14. class=<=50K 6075 ==> capitalgain=0 capitalloss=0 5631    <conf:(0.93)> lift:(1.07) lev:(0.04) [346] conv:(1.78)
15. native-country=United-States 7124 ==> capitalgain=0 6533    <conf:(0.92)> lift:(1) lev:(-0) [-7] conv:(0.99)
16. race=White 6823 ==> native-country=United-States 6242    <conf:(0.91)> lift:(1.02) lev:(0.02) [151] conv:(1.26)
17. race=White capitalgain=0 6228 ==> native-country=United-States 5695    <conf:(0.91)> lift:(1.02) lev:(0.02) [135] conv:(1.25)
18. capitalloss=0 7597 ==> capitalgain=0 6943    <conf:(0.91)> lift:(1) lev:(-0) [-31] conv:(0.95)
19. race=White capitalloss=0 6481 ==> native-country=United-States 5920    <conf:(0.91)> lift:(1.02) lev:(0.02) [134] conv:(1.24)
20. capitalloss=0 native-country=United-States 6781 ==> capitalgain=0 6190    <conf:(0.91)> lift:(0.99) lev:(-0) [-35] conv:(0.94)
21. race=White capitalgain=0 capitalloss=0 5886 ==> native-country=United-States 5373    <conf:(0.91)> lift:(1.02) lev:(0.01) [119] conv:(1.23)
22. race=White 6823 ==> capitalgain=0 6228    <conf:(0.91)> lift:(0.99) lev:(-0) [-35] conv:(0.94)
23. race=White native-country=United-States 6242 ==> capitalgain=0 5695    <conf:(0.91)> lift:(0.99) lev:(-0) [-35] conv:(0.93)
24. race=White capitalloss=0 6481 ==> capitalgain=0 5886    <conf:(0.91)> lift:(0.99) lev:(-0.01) [-63] conv:(0.89)
25. race=White capitalloss=0 native-country=United-States 5920 ==> capitalgain=0 5373    <conf:(0.91)> lift:(0.99) lev:(-0.01) [-61] conv:(0.89)
```

Question-5

5. From the given data, they wish to find the items that were purchased most frequently. They also wish to determine the item(s) which encouraged the customer to purchase additional item(s). Such analysis is commonly termed as **Market Basket Analysis**, where the interesting associations between various items are determined.

The analysis that leads to determining purchase behaviour of customers arises from the items attribute. The marketing team seeks to study the items attribute more closely to determine associations between various items.

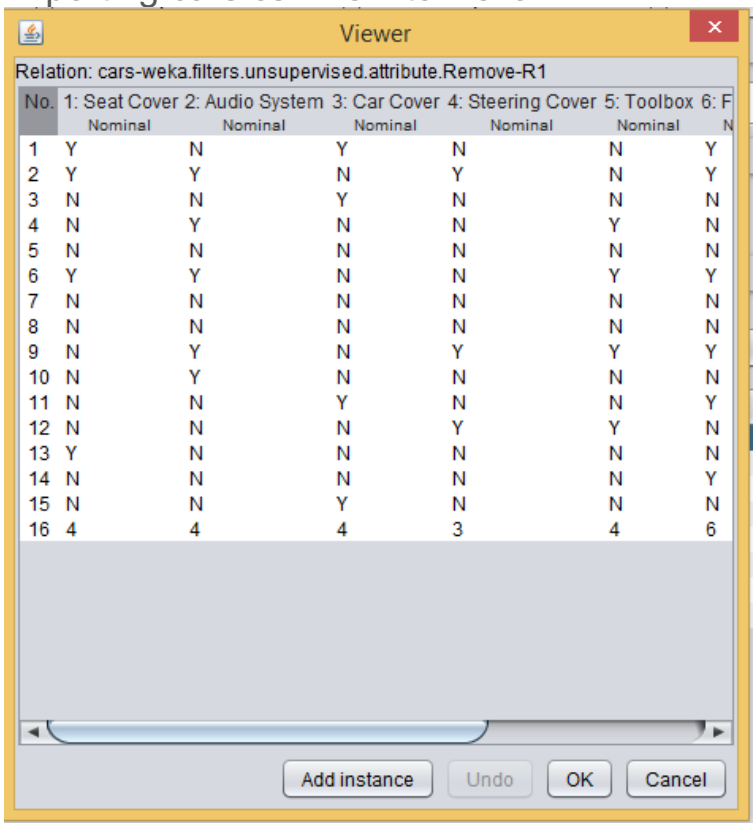
From the given sample data set, the most frequently purchased item can be determined using a frequency table, as shown below

Transaction	Seat Cover	Audio system	Car cover	Steering Cover	Toolbox	Foot mats	Mud flaps	Window tint
1	Y		Y			Y	Y	
2	Y	Y		Y		Y		
3			Y					Y
4		Y					Y	Y
5					Y			
6	Y					Y		
7		Y			Y			Y
8							Y	
9				Y		Y		
10		Y			Y			Y
11			Y			Y		
12				Y				
13	Y				Y			
14						Y		
15			Y					
Total	**4**	**4**	**4**	**3**	**4**	**6**	**3**	**4**

Creating the data-set

Transaction	Seat Cover	Audio System	Car Cover	Steering Cover	Toolbox	Foot mats	Mud flaps	Window Tints
1	Y	N	Y	N	N	Y	Y	N
2	Y	Y	N	Y	N	Y	N	N
3	N	N	Y	N	N	N	N	Y
4	N	Y	N	N	Y	N	Y	Y
5	N	N	N	N	N	N	N	N
6	Y	Y	N	N	Y	Y	N	N
7	N	N	N	N	N	N	N	Y
8	N	N	N	N	N	N	Y	N
9	N	Y	N	Y	Y	Y	N	N
10	N	Y	N	N	N	N	N	Y
11	N	N	Y	N	N	Y	N	N
12	N	N	N	Y	Y	N	N	N
13	Y	N	N	N	N	N	N	N
14	N	N	N	N	N	Y	N	N
15	N	N	Y	N	N	N	N	N
Total	4	4	4	3	4	6	3	4

Importing cars.csv file into weka



Apriori Report

Associator output

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:     cars-weka.filters.unsupervised.attribute.Remove-R1
Instances:    16
Attributes:   8
              Seat Cover
              Audio System
              Car Cover
              Steering Cover
              Toolbox
              Foot mats
              Mud flaps
              Window Tints
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.45 (7 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 11

Generated sets of large itemsets:

Size of set of large itemsets L(1): 8
```