**Expt.No:**

**PageNo:**

**Date:**

**EXERCISE-3**

**3. Demonstration of Association rule process on dataset**

**contactlenses.arff using apriori algorithm**

**Expt.No:**

**PageNo:**

**Date:**

**EXERCISE-3**

**3. Demonstration of Association rule process on dataset**

**contactlenses.arff using apriori algorithm**

**Demonstration of Association rule process on dataset**

**contactlenses.arff using apriori algorithm**

**Demonstration of Association rule process on dataset**

**contactlenses.arff using apriori algorithm**

***set1:***

# Perform the basic pre-processing operations on data relation such as removing an attribute and filter attribute bank data.

**Aim** Perform the basic pre-processing operations on data relation such as removing an attribute and filter attribute bank data.

**Dataset:**

@relation bank

@attribute cust {male,female}

@attribute accno {0101,0102,0103,0104,0105,0106,0107,0108,0109,0110,0111,0112,0113,0114,0115}

@attribute bankname {sbi,hdfc,sbh,ab,rbi}

@attribute location {hyd,jmd,antp,pdtr,kdp}

@attribute deposit {yes,no}

@data

male,0101,sbi,hyd,yes

female,0102,hdfc,jmd,no

male,0103,sbh,antp,yes

male,0104,ab,pdtr,yes

female,0105,sbi,jmd,no

male,0106,ab,hyd,yes

female,0107,rbi,jmd,yes

female,0108,hdfc,kdp,no

male,0109,sbh,kdp,yes

male,0110,ab,jmd,no

female,0111,rbi,kdp,yes

male,0112,sbi,jmd,yes

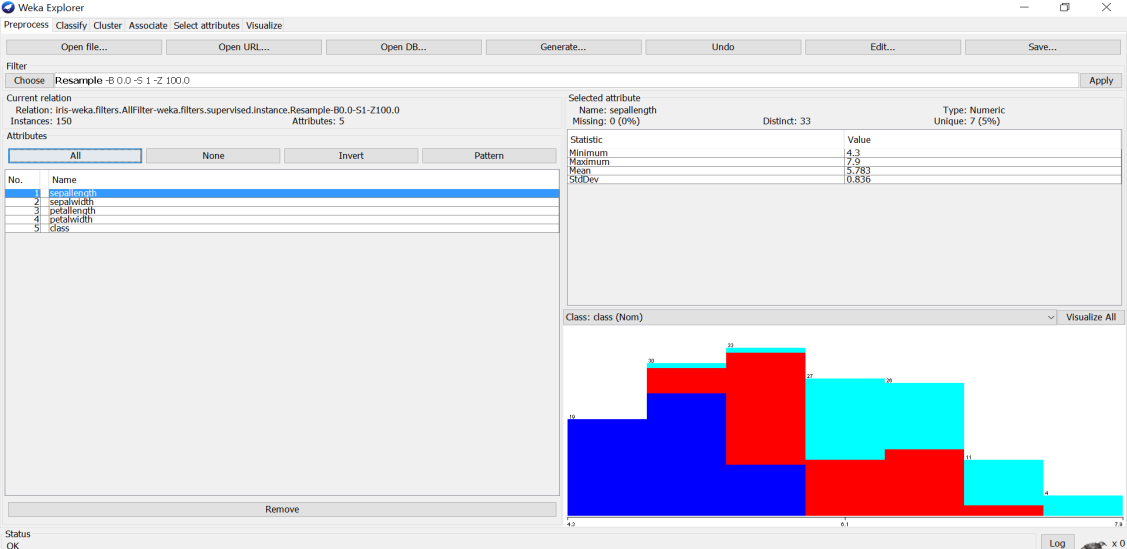
female,0113,rbi,antp,no

male,0114,hdfc,pdtr,yes

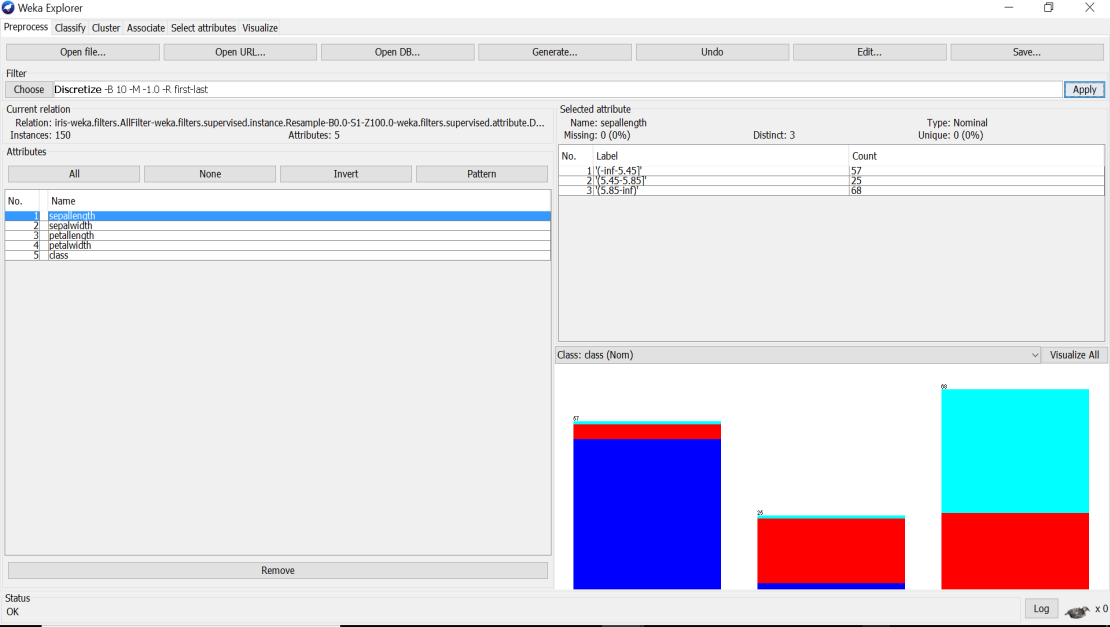
female,0115,sbh,pdtr,no

**Procedure:**

* + 1. For preprocessing the data after selecting the dataset (IRis.arff).
    2. Select Filter option & apply the resample filter & see the below results.



1. Select another filter option & apply the discretization filter, see the below results



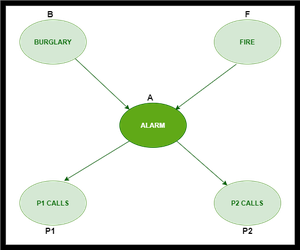
Likewise, we can apply different filters for preprocessing the data & see the results in different dimensions.

# What is Bayesian belief network ?explain with an example

**Bayesian Belief Network**is a graphical representation of different probabilistic relationships among random variables in a particular set. It is a classifier with no dependency on attributes i.e it is condition independent. Due to its feature of joint probability, the probability in Bayesian Belief Network is derived, based on a condition — P(attribute/parent) i.e probability of an attribute, true over parent attribute.

(Note: A classifier assigns data in a collection to desired categories.)

* Consider this example:



* In the above figure, we have an alarm ‘A’ – a node, say installed in a house of a person ‘gfg’, which rings upon two probabilities i.e burglary ‘B’ and fire ‘F’, which are – parent nodes of the alarm node. The alarm is the parent node of two probabilities P1 calls  ‘P1’ & P2 calls ‘P2’ person nodes.
* Upon the instance of burglary and fire, ‘P1’ and ‘P2’ call person ‘gfg’, respectively. But, there are few drawbacks in this case, as sometimes ‘P1’ may forget to call the person ‘gfg’, even after hearing the alarm, as he has a tendency to forget things, quick.  Similarly, ‘P2’, sometimes fails to call the person ‘gfg’, as he is only able to hear the alarm, from a certain distance.

**Q)** Find the probability that ‘P1’ is true (P1 has called ‘gfg’), ‘P2’ is true (P2 has called ‘gfg’) when the alarm ‘A’ rang, but no burglary ‘B’ and fire ‘F’ has occurred.

=> **P ( P1, P2, A, ~B, ~F)** [ where- P1, P2 & A are ‘true’ events and ‘~B’ & ‘~F’ are ‘false’ events]

[ **Note:** The values mentioned below are neither calculated nor computed. They have observed values ]

***Burglary ‘B’ –***

* **P (B=T) = 0.001** (‘B’ is true i.e burglary has occurred)
* **P (B=F) = 0.999**(‘B’ is false i.e burglary has not occurred)

***Fire ‘F’ –***

* **P (F=T) = 0.002** (‘F’ is true i.e fire has occurred)
* **P (F=F) = 0.998** (‘F’ is false i.e fire has not occurred)

***Alarm ‘A’ –***

|  |  |  |  |
| --- | --- | --- | --- |
| **B** | **F** | **P (A=T)** | **P (A=F)** |
| T | T | 0.95 | 0.05 |
| T | F | 0.94 | 0.06 |
| F | T | 0.29 | 0.71 |
| F | F | 0.001 | **0.999** |

* The alarm ‘A’ node can be ‘true’ or ‘false’ ( i.e may have rung or may not have rung). It has two parent nodes burglary ‘B’ and fire ‘F’ which can be ‘true’ or ‘false’ (i.e may have occurred or may not have occurred) depending upon different conditions.

***Person ‘P1’ –***

|  |  |  |
| --- | --- | --- |
| **A** | **P (P1=T)** | **P (P1=F)** |
| T | **0.95** | 0.05 |
| F | 0.05 | 0.95 |

* The person ‘P1’ node can be ‘true’ or ‘false’ (i.e may have called the person ‘gfg’ or not) . It has a parent node, the alarm ‘A’, which can be ‘true’ or ‘false’ (i.e may have rung or may not have rung ,upon burglary ‘B’ or fire ‘F’).

***Person ‘P2’ –***

|  |  |  |
| --- | --- | --- |
| **A** | **P (P2=T)** | **P (P2=F)** |
| T | **0.80** | 0.20 |
| F | 0.01 | 0.99 |

* The person ‘P2’ node can be ‘true’ or false’ (i.e may have called the person ‘gfg’ or not). It has a parent node, the alarm ‘A’, which can be ‘true’ or ‘false’ (i.e may have rung or may not have rung, upon burglary ‘B’ or fire ‘F’).

**Solution:** Considering the observed probabilistic scan –

With respect to the question —  **P ( P1, P2, A, ~B, ~F)**, we need to get the probability of ‘P1’. We find it with regard to its parent node – alarm ‘A’. To get the probability of ‘P2’, we find it with regard to its parent node — alarm ‘A’.

We find the probability of alarm ‘A’ node with regard to ‘~B’ & ‘~F’ since burglary ‘B’ and fire ‘F’ are parent nodes of alarm ‘A’.

From the observed probabilistic scan, we can deduce –

**P ( P1, P2, A, ~B, ~F)**

**= P (P1/A) \* P (P2/A) \* P (A/~B~F) \* P (~B) \* P (~F)**

**= 0.95 \* 0.80 \* 0.001 \* 0.999 \* 0.998**

**= 0.00075**

**----------------------------------------------------------------------------------------------------**

# SET2:

# To list all the categorical(or nominal) attributes and the real valued attributes using Weka mining(german credit data)

AIM :To list all the categorical (or nominal) attributes and the real valued attributes using WEKA mining tool.

Tools/ Apparatus: Weka Mining tool.

Procedure:

 Steps for identifying categorical attributes

1.Double click on credit-g.arff file.

2.Select all categorical attributes.

3.Click on invert.

4.Then we get all real valued attributes selected

5.Click on remove

6.Click on visualize all.

Steps for identifying real valued attributes

1.Double click on credit-g.arff file.Select all real valued attributes.

2.Click on invert.

3.Then we get all categorial attributes selected

4.Click on remove

5.Click on visualize all

**Output:**

**Categorical/ Nominal Attributes:**

1. Checking\_status

2.Credit\_history

3.Purpose

4.Savings\_status

5. Employment

6. Personal\_status

7.Other\_parties

8.Property\_magnitude

9. Other\_payment\_plans

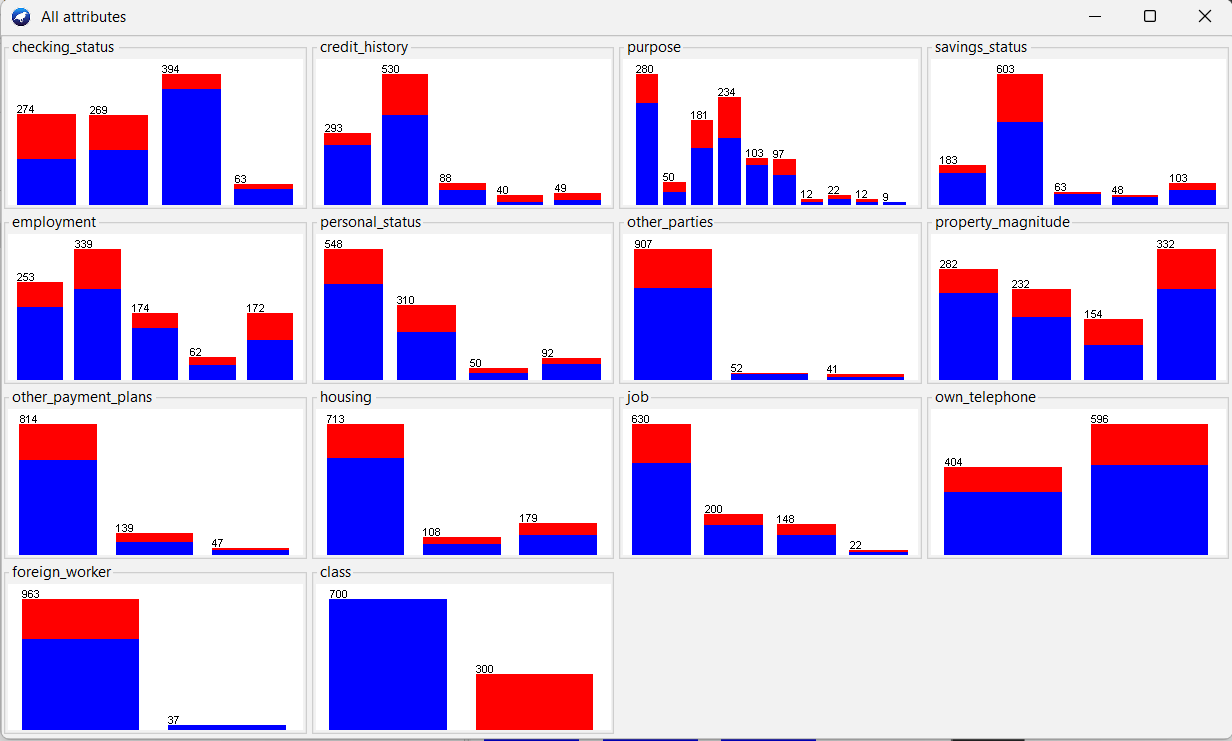
10. Housing

11.Job

12.own\_telephone

13. foreign\_worker

14. Class



**Numeric** **Attributes:**

1. Duration

2. Credit\_amount

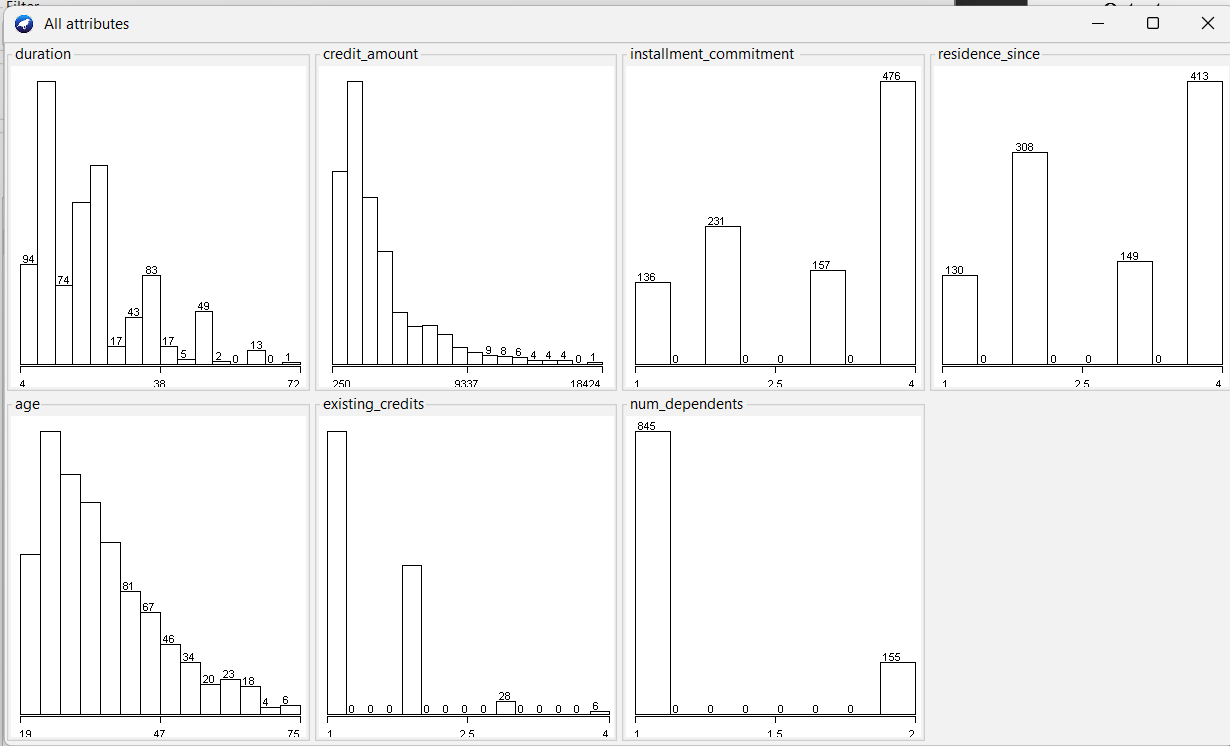
3. Installment\_commitment

4. Residece\_since

5. Age

6. Existing\_credits

7. Num\_dependents.



Result:Hence all categorical and Numerical attributes are displayed

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 |  |  |  |  |
| 2 | 9 | 0 |  |  |  |
| 3 | 3 | 7 | 0 |  |  |
| 4 | 6 | 5 | 9 | 0 |  |
| 5 | 11 | 10 | 2 | 8 | 0 |

# Explain agglomerative methods and construct a dendrogram for below distance matrix using single linkage

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 |  |  |  |  |
| 2 | 9 | 0 |  |  |  |
| 3 | 3 | 7 | 0 |  |  |
| 4 | 6 | 5 | 9 | 0 |  |
| 5 | 11 | 10 | 2 | 8 | 0 |

# SET3:

# Load each dataset into Weka and perform Naive-bayes classification(contact-lenses). Interpret the results obtained

**Dataset:**

@relation contact-lenses

@attribute age {young, pre-presbyopic, presbyopic}

@attribute spectacle-prescrip {myope, hypermetrope}

@attribute astigmatism {no, yes}

@attribute tear-prod-rate {reduced, normal}

@attribute contact-lenses {soft, hard, none}

@data

young,myope,no,reduced,none

young,myope,no,normal,soft

young,myope,yes,reduced,none

young,myope,yes,normal,hard

young,hypermetrope,no,reduced,none

young,hypermetrope,no,normal,soft

young,hypermetrope,yes,reduced,none

young,hypermetrope,yes,normal,hard

pre-presbyopic,myope,no,reduced,none

pre-presbyopic,myope,no,normal,soft

pre-presbyopic,myope,yes,reduced,none

pre-presbyopic,myope,yes,normal,hard

pre-presbyopic,hypermetrope,no,reduced,none

pre-presbyopic,hypermetrope,no,normal,soft

pre-presbyopic,hypermetrope,yes,reduced,none

pre-presbyopic,hypermetrope,yes,normal,none

presbyopic,myope,no,reduced,none

presbyopic,myope,no,normal,none

presbyopic,myope,yes,reduced,none

presbyopic,myope,yes,normal,hard

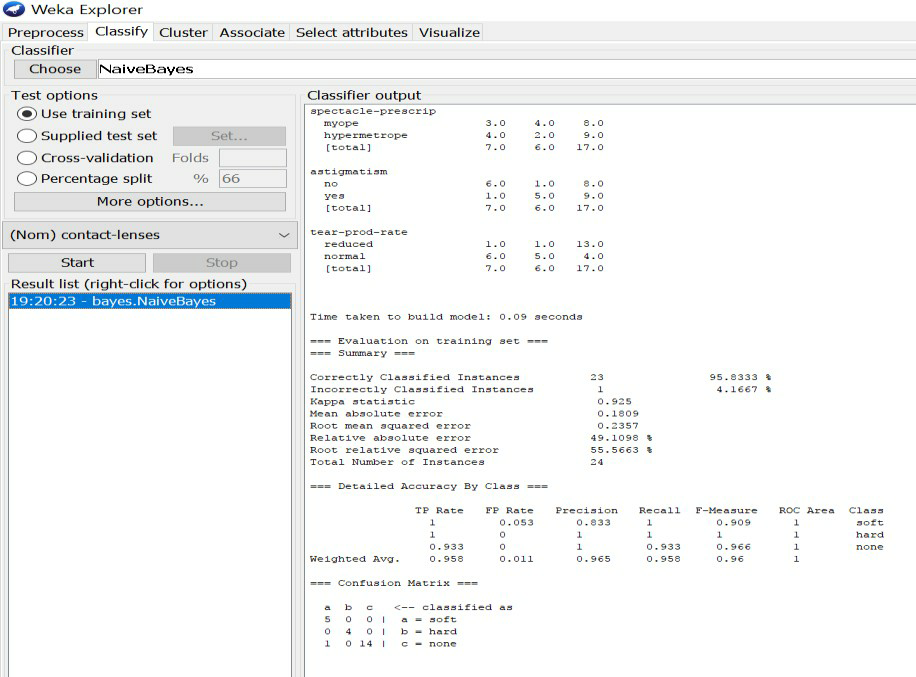
presbyopic,hypermetrope,no,reduced,none

presbyopic,hypermetrope,no,normal,soft

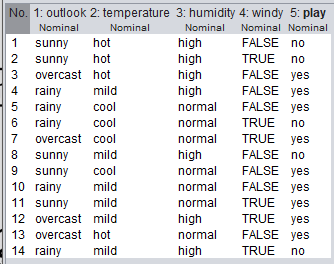
presbyopic,hypermetrope,yes,reduced,none

presbyopic,hypermetrope,yes,normal,none

**Procedure for Naïve-Bayes:**

1. Load the dataset (Contact-Lenses. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under bayes section.
3. In which we selected Naïve-Bayes algorithm & click on start option with ―use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.

# Explain Naïve-Bayers classifiers and predict instance to the class (weather data set) if outlook=rainy,temp=cool, humidity=high, windy=true, play =?



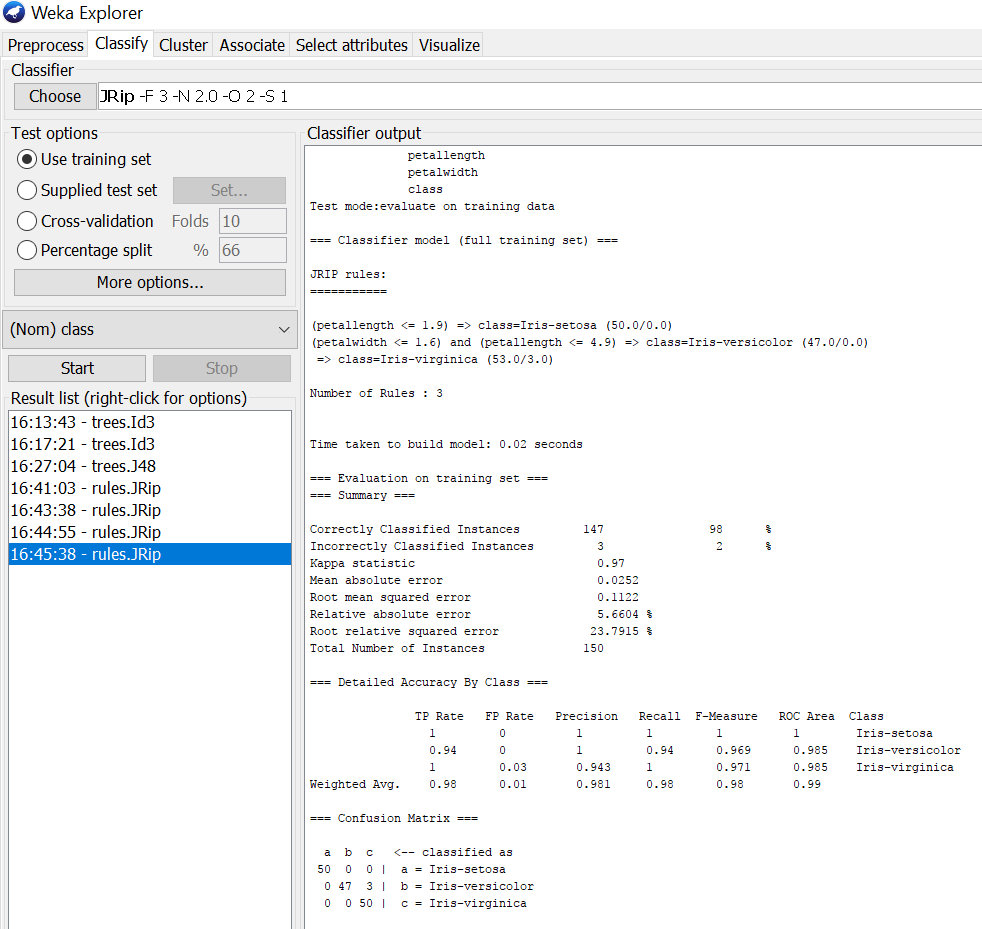
Play=no

# SET4:

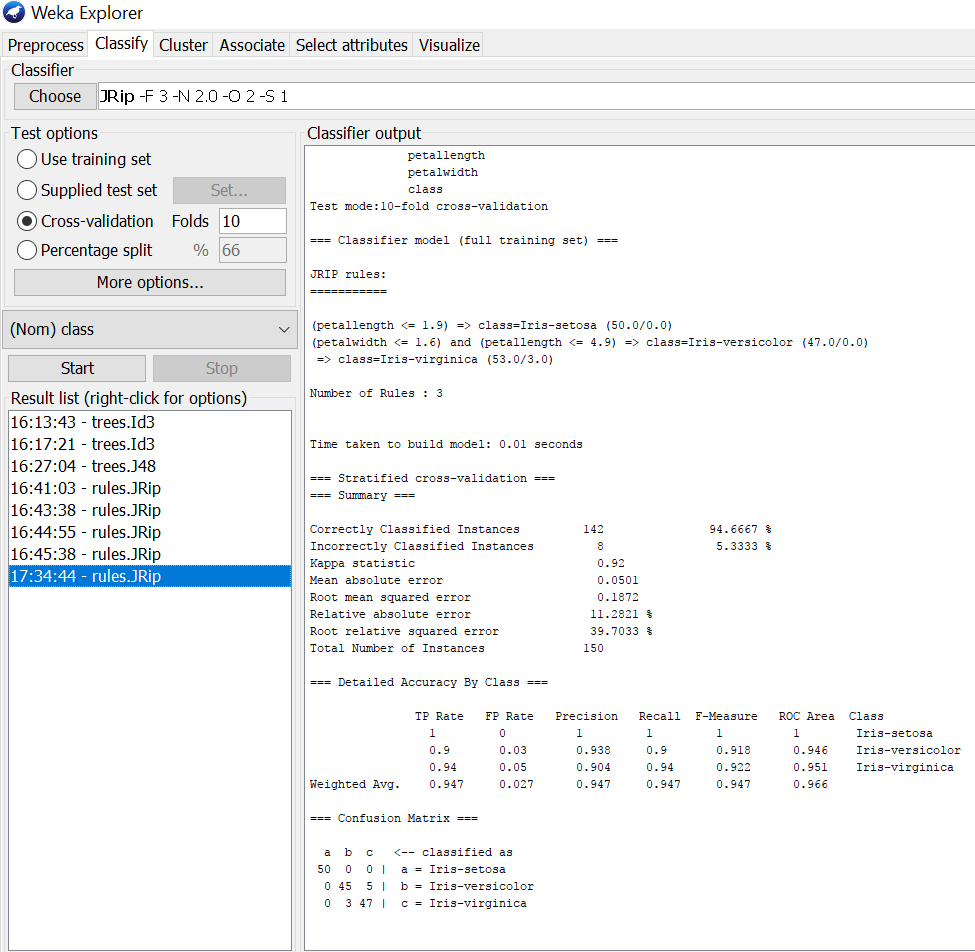
# Demonstrate performing classification on data set Extract if-then rules from the decision tree generated by the classifier, Observethe confusion matrix and derive Accuracy, F-measure, TPrate, FPrate, Precision and Recall values. Apply cross-validation strategy with various fold levels and compare the accuracy results.(iris data)

## Procedure**:**

1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under rules section.
3. In which we selected JRip (If-then) algorithm & click on start option with ―use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists ofF-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.

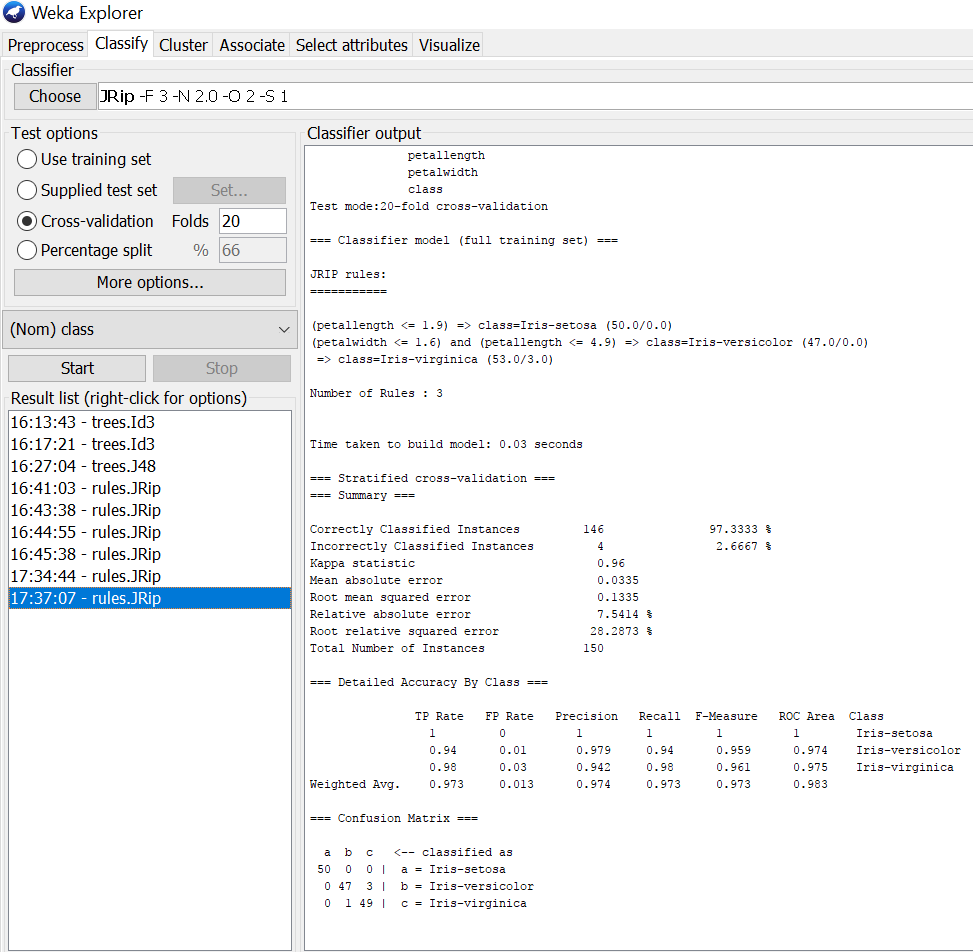


# Using Cross-Validation Strategy with 10 folds:

Here, we enabled cross-validation test option with 10 folds & clicked start button as represented below.

## Using Cross-Validation Strategy with 20 folds:

Here, we enabled cross-validation test option with 20 folds & clicked start button as represented below.



If we see the above results of cross validation with 10 folds & 20 folds. As per our observation the error rate is lesser with 20 folds got 97.3% correctness when compared to 10 folds got 94.6% correctness.

# Explain divisive method and construct a dendrogram for below distance matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 |  |  |  |  |
| B | 1 | 0 |  |  |  |
| C | 2 | 2 | 0 |  |  |
| D | 2 | 4 | 1 | 0 |  |
| E | 3 | 3 | 5 | 3 | 0 |

**1. Divisive clustering**

Divisive clustering, also known as the top-down clustering method assigns all of the observations to a single cluster and then partition the cluster into two least similar clusters.

# SET5:

# To create a Decision tree by cross validation training data set using Weka mining tool

**Dataset:**

@relation contact-lenses

@attribute age {young, pre-presbyopic, presbyopic}

@attribute spectacle-prescrip {myope, hypermetrope}

@attribute astigmatism {no, yes}

@attribute tear-prod-rate {reduced, normal}

@attribute contact-lenses {soft, hard, none}

@data

young,myope,no,reduced,none

young,myope,no,normal,soft

young,myope,yes,reduced,none

young,myope,yes,normal,hard

young,hypermetrope,no,reduced,none

young,hypermetrope,no,normal,soft

young,hypermetrope,yes,reduced,none

young,hypermetrope,yes,normal,hard

pre-presbyopic,myope,no,reduced,none

pre-presbyopic,myope,no,normal,soft

pre-presbyopic,myope,yes,reduced,none

pre-presbyopic,myope,yes,normal,hard

pre-presbyopic,hypermetrope,no,reduced,none

pre-presbyopic,hypermetrope,no,normal,soft

pre-presbyopic,hypermetrope,yes,reduced,none

pre-presbyopic,hypermetrope,yes,normal,none

presbyopic,myope,no,reduced,none

presbyopic,myope,no,normal,none

presbyopic,myope,yes,reduced,none

presbyopic,myope,yes,normal,hard

presbyopic,hypermetrope,no,reduced,none

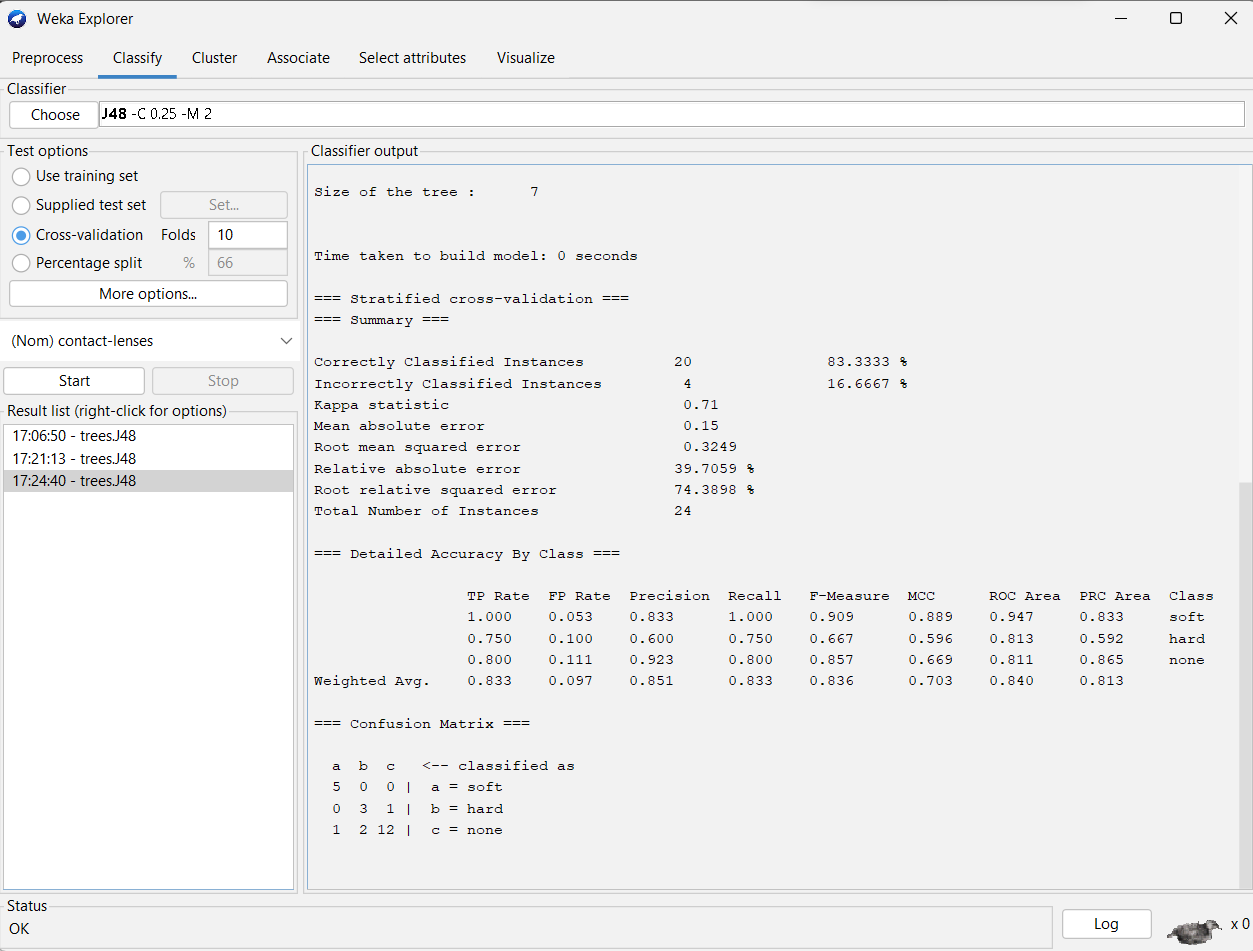
presbyopic,hypermetrope,no,normal,soft

presbyopic,hypermetrope,yes,reduced,none

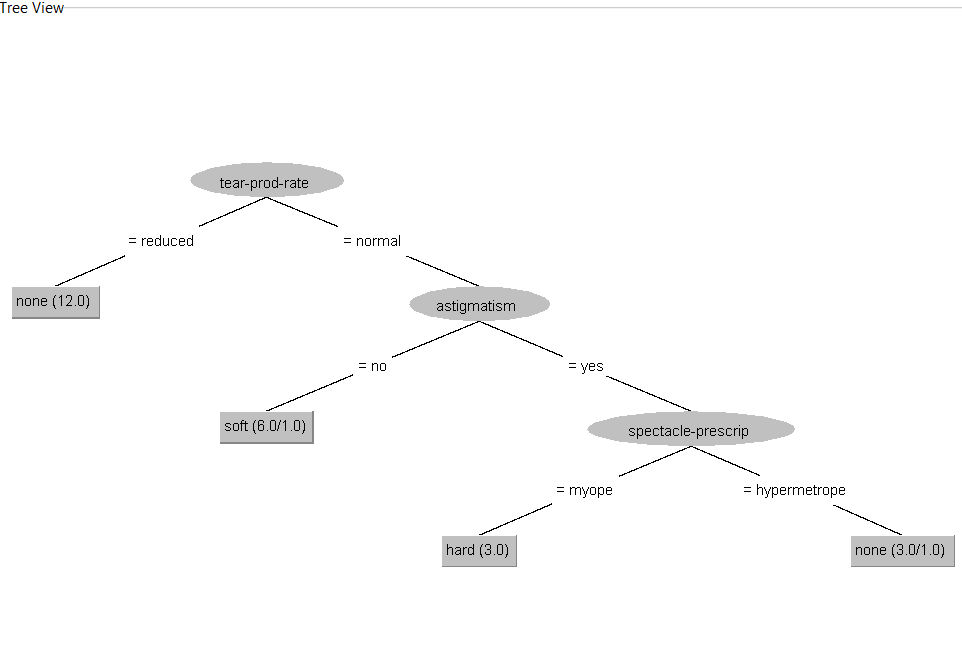
presbyopic,hypermetrope,yes,normal,none

Creating a decision tree using the Weka data mining tool with cross-validation involves several steps. Below is a step-by-step guide on how to do this:

1. **Prepare your Data**: You need to have your dataset in a compatible format (ARFF, CSV, etc.). If your data is not in ARFF format, you can use Weka's data preprocessing tools to convert it.
2. **Open Weka**: Launch the Weka GUI from your installed software.
3. **Load Data**: Click on the "Explorer" tab and then click on the "Open file" button to load your dataset.
4. **Select Classifier (Decision Tree)**: Click on the "Classify" tab. From the "Choose" section, select "trees" and then "J48" which is Weka's implementation of the C4.5 decision tree algorithm.
5. **Set Cross-Validation**: In the "Classifier" section, click on "Cross-validation" to configure the cross-validation settings.
   * Choose the number of folds (e.g., 10) for cross-validation.
   * Optionally, you can set the random seed for reproducibility.
6. **Run Cross-Validation**: Click on the "Start" button to run the cross-validation process. Weka will partition your dataset into folds, train the decision tree on multiple subsets, and evaluate its performance.



1. **View Results**: Once the cross-validation is complete, you will see the results in the "Result list" panel. You can click on individual runs to see the detailed results for each fold.
2. **Visualize the Decision Tree**: To visualize the generated decision tree, you can right-click on the J48 model in the "Result list" and select "Visualize tree". This will show you the graphical representation of the decision tree.



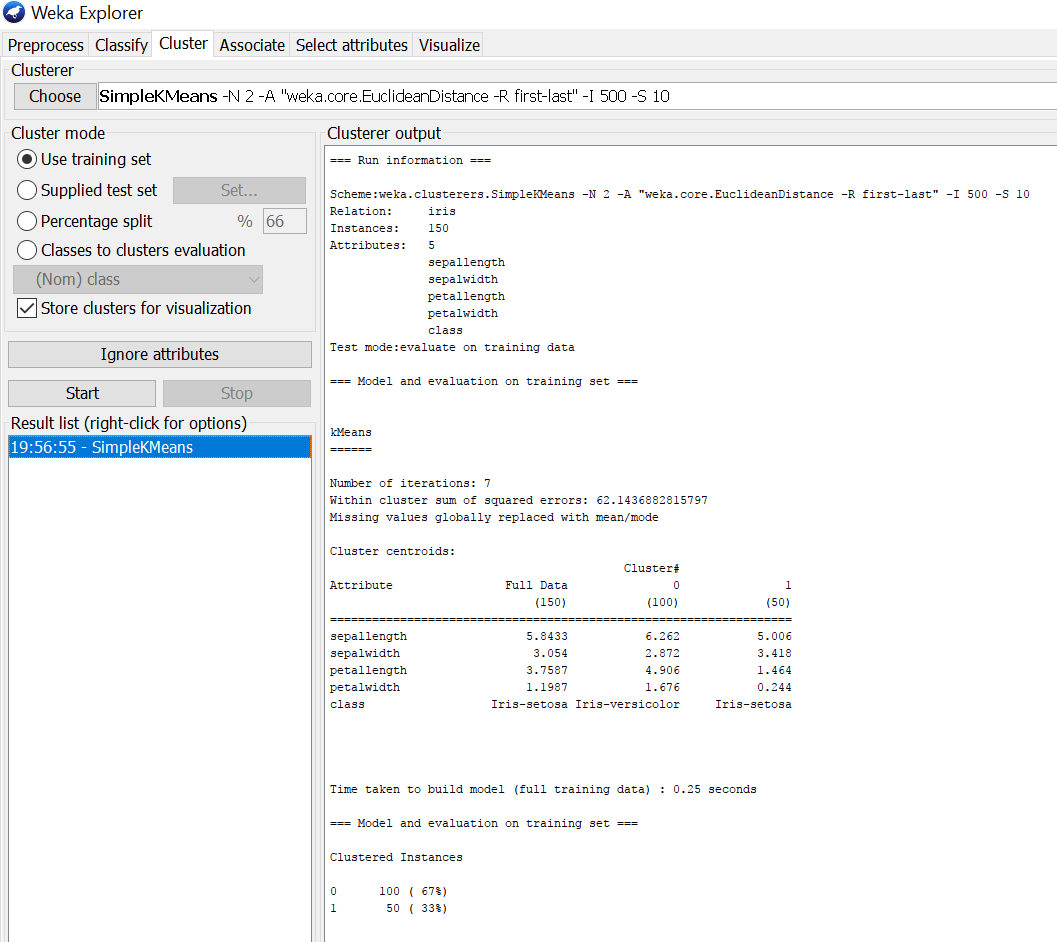
# Describe the general approaches to solve a classification problem

# **SET6:**

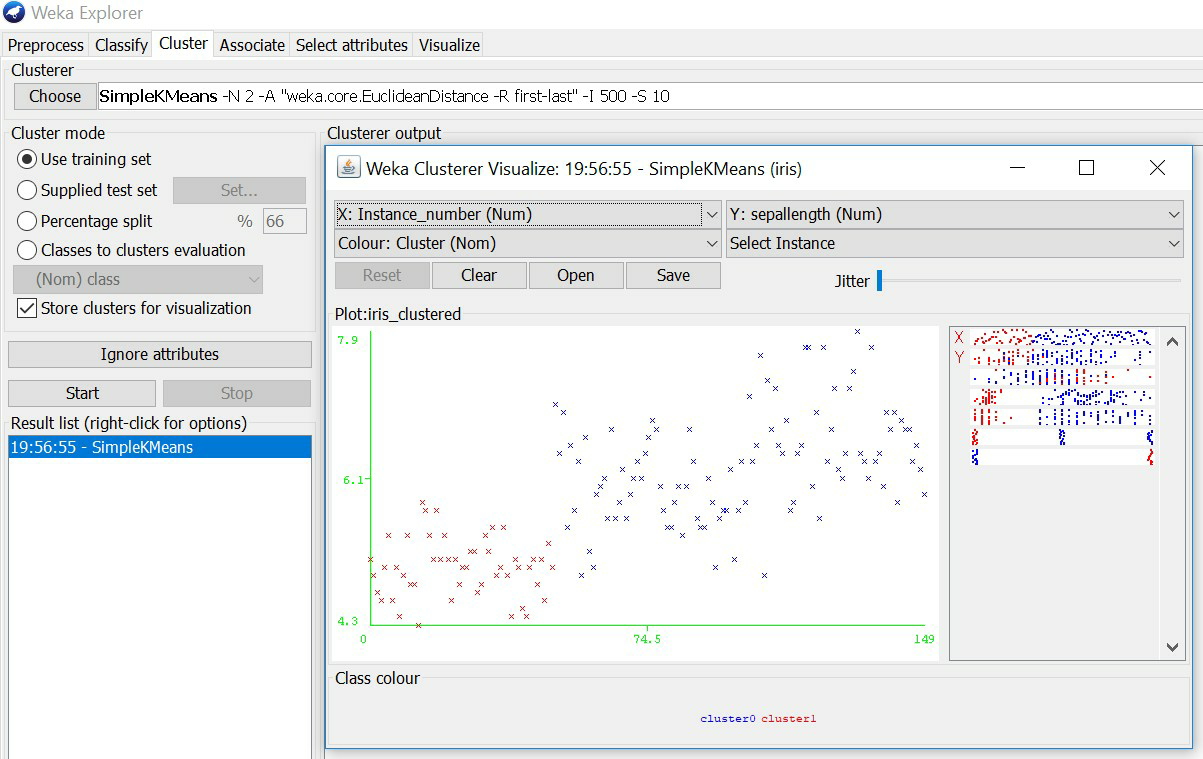
# Demonstrate performing clustering on data sets Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters). Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.

**Procedure:**

1. Load the dataset (Iris.arff) into weka tool
2. Go to cluster option & in left-hand navigation bar we can see different clustering algorithms under lazy section.
3. In which we selected Simple K-Means algorithm & click on start option with ―use training set‖ test option enabled.
4. Then we will get the sum of squared errors, centroids, No. of iterations & clustered instances as represented below.



1. If we right click on simple k means, we will get more options in which ―Visualize cluster assignments‖ should be selected for getting cluster visualization as shown below.



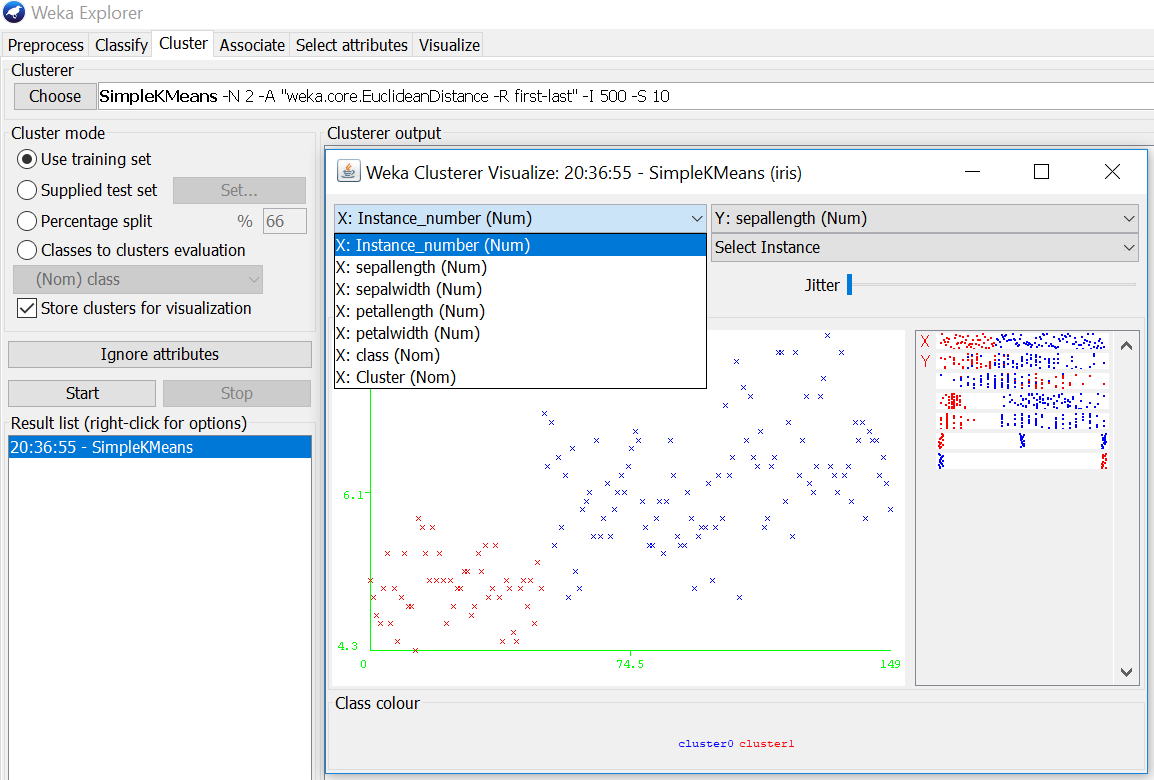
# Consider the following data, apply KNN algorithm to find class when acid durability=3 and strength=7

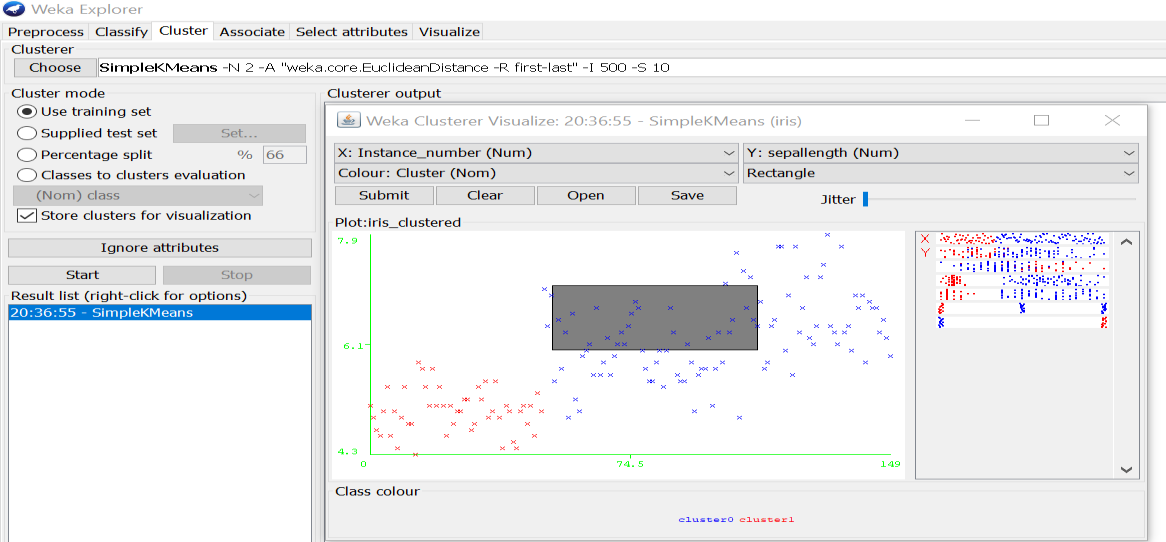
|  |  |  |  |
| --- | --- | --- | --- |
| name | acid durability | strength | class |
| Type1 | 7 | 7 | Bad |
| Type2 | 7 | 4 | Bad |
| Type3 | 3 | 4 | Good |
| Type4 | 1 | 4 | Good |

# SET7:

# Explore visualization features of Weka to visualize the clusters. Derive interesting insights and explain.

* + If we right click on simple k means, we will get more options in which ―Visualize cluster assignments‖ should be selected for getting cluster visualization as shown below.
  + In that cluster visualization we are having different features to explore by changing the
  + X-axis, Y-axis, Color, Jitter& Select instance (Rectangle, Polygon & Polyline) for getting different sets of cluster outputs.



* + As shown in above screenshot, all the dataset (Iris.arff) tuples are represented in X-axis & in similar way it will represented for y-axis also. For each cluster, the color will be different. In the above figure, there are two clusters which are represented in blue & red colors.
  + In the select instance we can select different shapes for choosing clustered area as shown in below screenshot, rectangle shape is selected
  + By this visualization feature we can observe different clustering outputs for an dataset by changing those X-axis, Y-axis, Color & Jitter options.

# Explain agglomerative methods and construct a dendrogram for below distance matrix using complete linkage

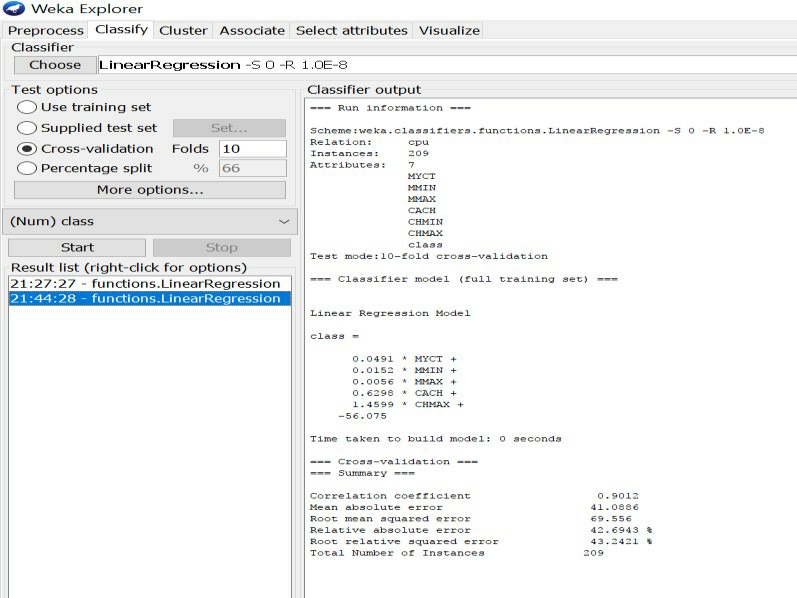
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | | 5 |
| 1 | 0 |  |  |  | |  |
| 2 | 9 | 0 |  |  | |  |
| 3 | 3 | 7 | 0 |  | |  |
| 4 | 6 | 5 | 9 | 0 | |  |
| 5 | 11 | 10 | 2 | 8 | 0 | |

# SET8:

# Use options cross-validation and percentage split and repeat running the Linear Regression Model. Observe the results and meaningful results.

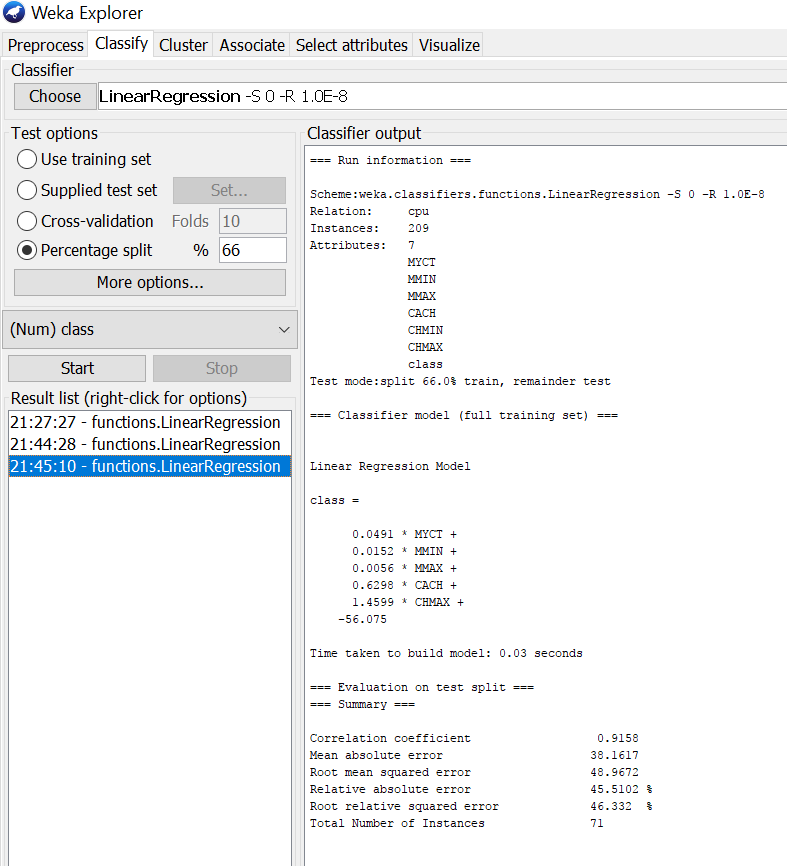
# Aim: Use options cross-validation and percentage split and repeat running the Linear Regression Model. Observe the results and derive meaningful results.

**Procedure for cross-validation:**

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Linear Regression algorithm & click on start option with cross validation option with 10 folds.
4. Then we will get regression model & its result as shown below.

# Procedure for percentage split:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Linear Regression algorithm & click on start option with percentage split option with 66% split.
4. Then we will get regression model & its result as shown below.



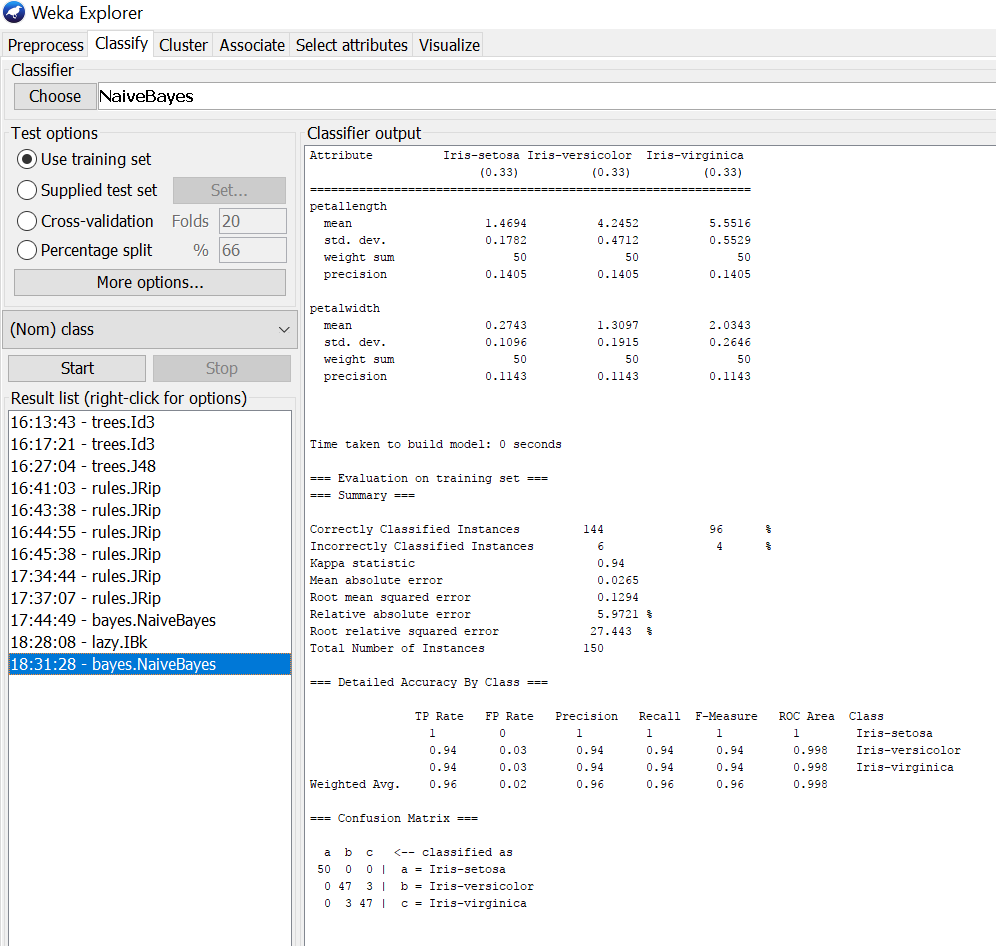
# :**Sample Programs using German Credit Data**

**SET9:**

# Write a program to implement Bayes classification technique(contact lenses)

**Procedure for Naïve-Bayes:**

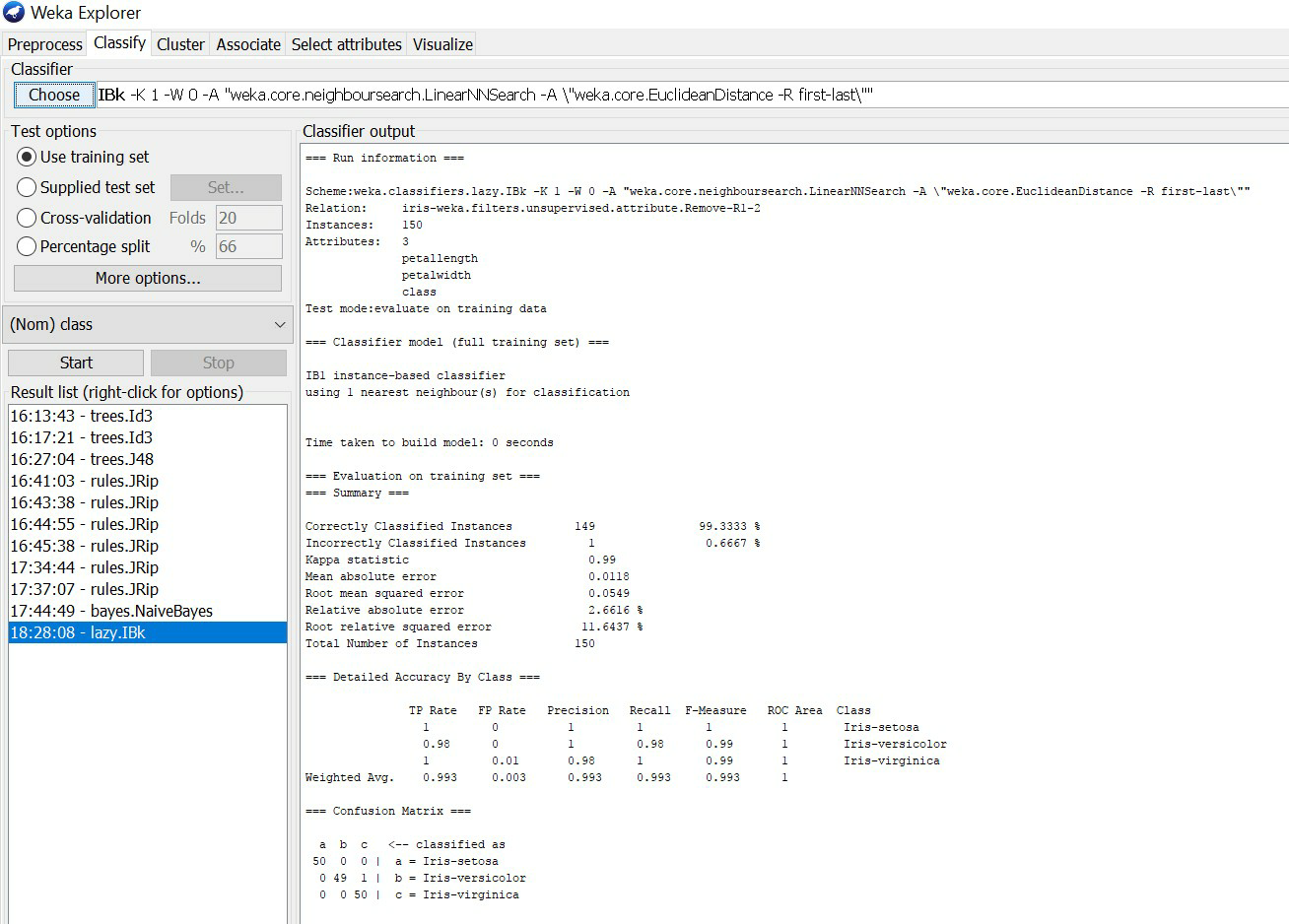
1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under bayes section.
3. In which we selected Naïve-Bayes algorithm & click on start option with ―use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



1. Write KNN classification algorithm with an example

# Procedure for K-Nearest Neighbour (IBK):

1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under lazy section.
3. In which we selected K-Nearest Neighbour (IBK) algorithm & click on start option with ―use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



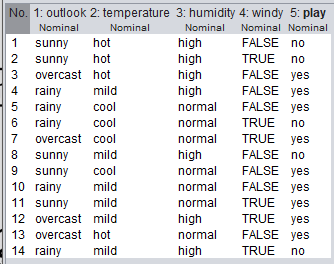
**SET10:**

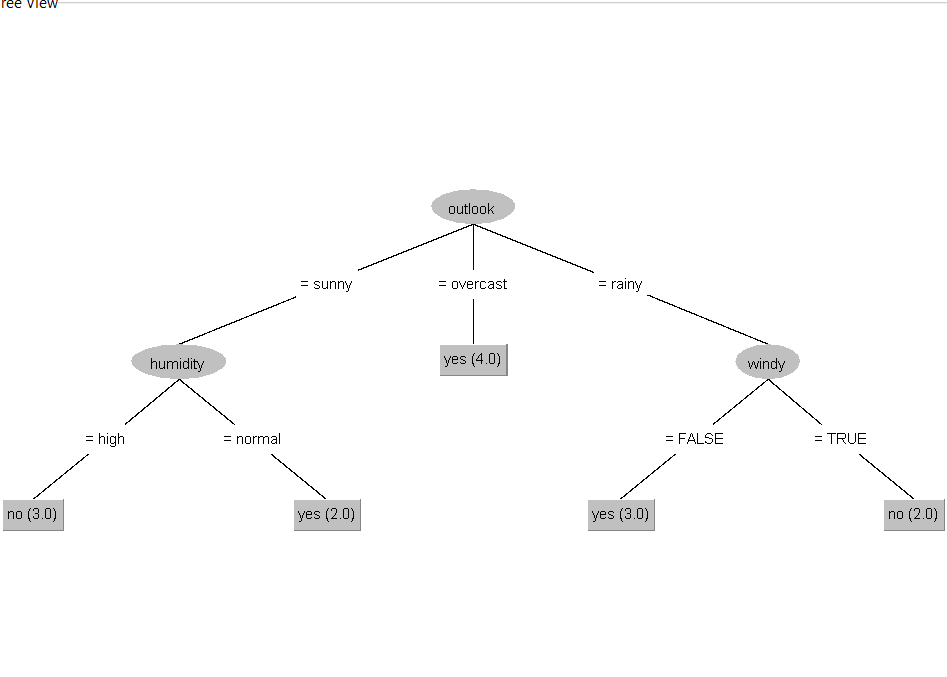
# To predict with the smallest total error using rule based on OneR attribute

# Demonstrate of clustering rule process by using simple K-mean algorithm

**SET11:**

# TO PREDICT THE RESULT FROM A GIVEN DATASET USING DECISION TREE4 J48 ALGORITHM Construct a Decision tree using J48 for blow data set





# To Predict the most probable class “contact –lenses” by computer on each instance based on training set using Baye’s theorem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 |  |  |  |  |
| 2 | 9 | 0 |  |  |  |
| 3 | 3 | 7 | 0 |  |  |
| 4 | 6 | 5 | 9 | 0 |  |
| 5 | 11 | 10 | 2 | 8 | 0 |

Explain agglomerative methods and construct a dendrogram for below distance matrix using single linkage