

## EXERCISE -5

- (i) Implement the Classification using Decision Tree algorithm on 'Weather' dataset. Draw the confusion matrix and report the model with accuracy.
- (ii) Implement Bayesian Classification and analyze the results on 'iris' Dataset.

### 5.1.1 Problem Statement:

Implement the Classification using Decision Tree algorithm on 'Weather' dataset. Draw the confusion matrix and report the model with accuracy.

### 5.1.2 Description:

#### About Dataset used

The weather data is a small [open data set](#) with only 14 examples.

In RapidMiner it is named [Golf Dataset](#), whereas Weka has two data set: weather.nominal.arff and weather.numeric.arff

The dataset contains data about weather conditions are suitable for playing a game of [golf](#). the original dataset that only has 5 variables:

- 1.outlook
- 2.temperature
- 3.humidity
- 4.windy
- 5.play

#### About Arff:

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

ARFF files have two distinct sections. The first section is the **Header** information, which is followed the **Data** information.

The **Header** of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types.

Lines that begin with a % are comments. The **@RELATION**, **@ATTRIBUTE** and **@DATA** declarations are case insensitive.

#### About CSV:

Files with .csv (Comma Separated Values) extension represent plain text files that contain records of data with comma separated values. Each line in a CSV file is a new record from the set of records contained in the file. Such files are generated when data transfer is intended from one storage system to another. Since all applications can recognize records separated by comma, import of such data files to database is done very conveniently. Almost all spreadsheet applications such as Microsoft Excel or OpenOffice Calc can import CSV without much effort. Data imported from such files is arranged in cells of a spreadsheet for representation to user.

## Datatypes that are supported by Weka:

- numeric
- integer is treated as numeric
- real is treated as numeric
- [nominal-specification]
- string
- date [date-format]
- relational for multi-instance data (for future use)

where [nominal-specification] and [date-format] are defined below. The keywords **numeric**, **real**, **integer**, **string** and **date** are case insensitive.

### Numeric attributes

Numeric attributes can be real or integer numbers.

### Nominal attributes

Nominal values are defined by providing an [nominal-specification] listing the possible values: {[nominal-name1], [nominal-name2], [nominal-name3], ...}

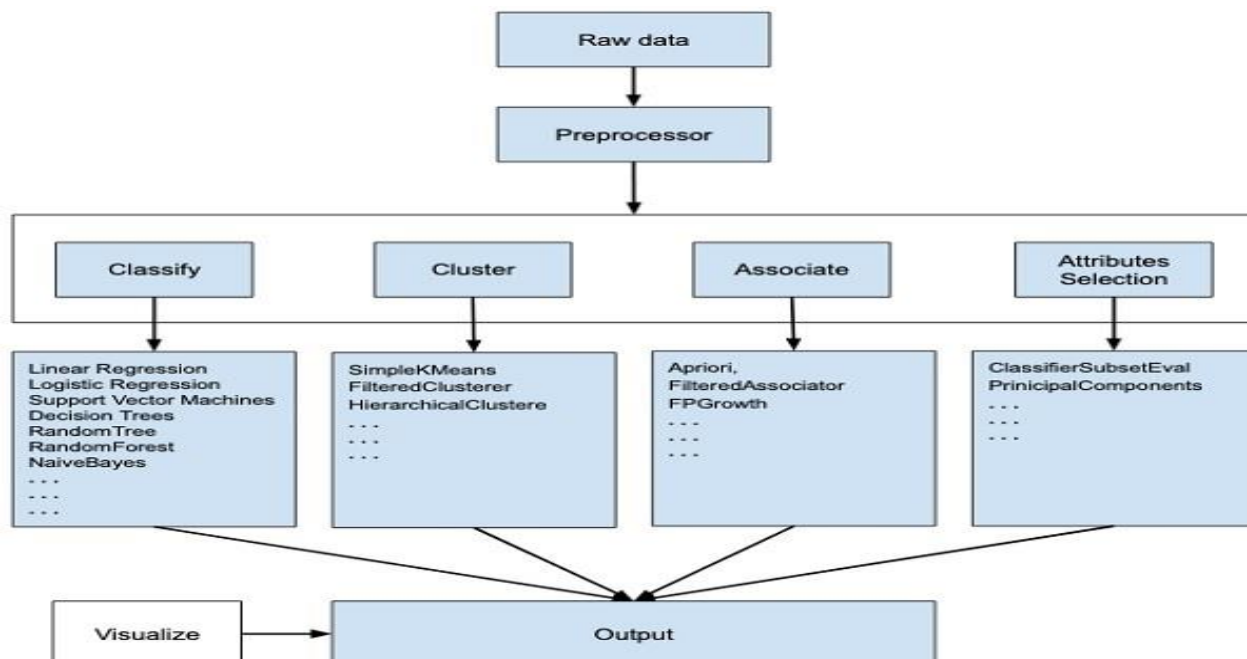
For example, the class value of the Iris dataset can be defined as follows:

```
@ATTRIBUTE class    {Iris-setosa,Iris-versicolor,Iris-virginica}
```

Values that contain spaces must be quoted.

## About WEKA Software:

WEKA - an opensource software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems. What WEKA offers is summarized in the following diagram –



If you observe the beginning of the flow of the image, you will understand that there are many stages in dealing with Big Data to make it suitable for machine learning

First, you will start with the raw data collected from the field. This data may contain several null values and irrelevant fields. You use the data preprocessing tools provided in WEKA to cleanse the data.

Then, you would save the preprocessed data in your local storage for applying ML algorithms.

Next, depending on the kind of ML model that you are trying to develop you would select one of the options such as **Classify**, **Cluster**, or **Associate**. The **Attributes Selection** allows the automatic selection of features to create a reduced dataset.

Note that under each category, WEKA provides the implementation of several algorithms. You would select an algorithm of your choice, set the desired parameters and run it on the dataset.

Then, WEKA would give you the statistical output of the model processing. It provides you a visualization tool to inspect the data.

The various models can be applied on the same dataset. You can then compare the outputs of different models and select the best that meets your purpose.

Thus, the use of WEKA results in a quicker development of machine learning models on the whole.

Now that we have seen what WEKA is and what it does, in the next chapter let us learn how to install WEKA on your local computer

### 5.1.3 Steps to implement the classification:

1.To classify the Weather database using Decision Tree algorithm.

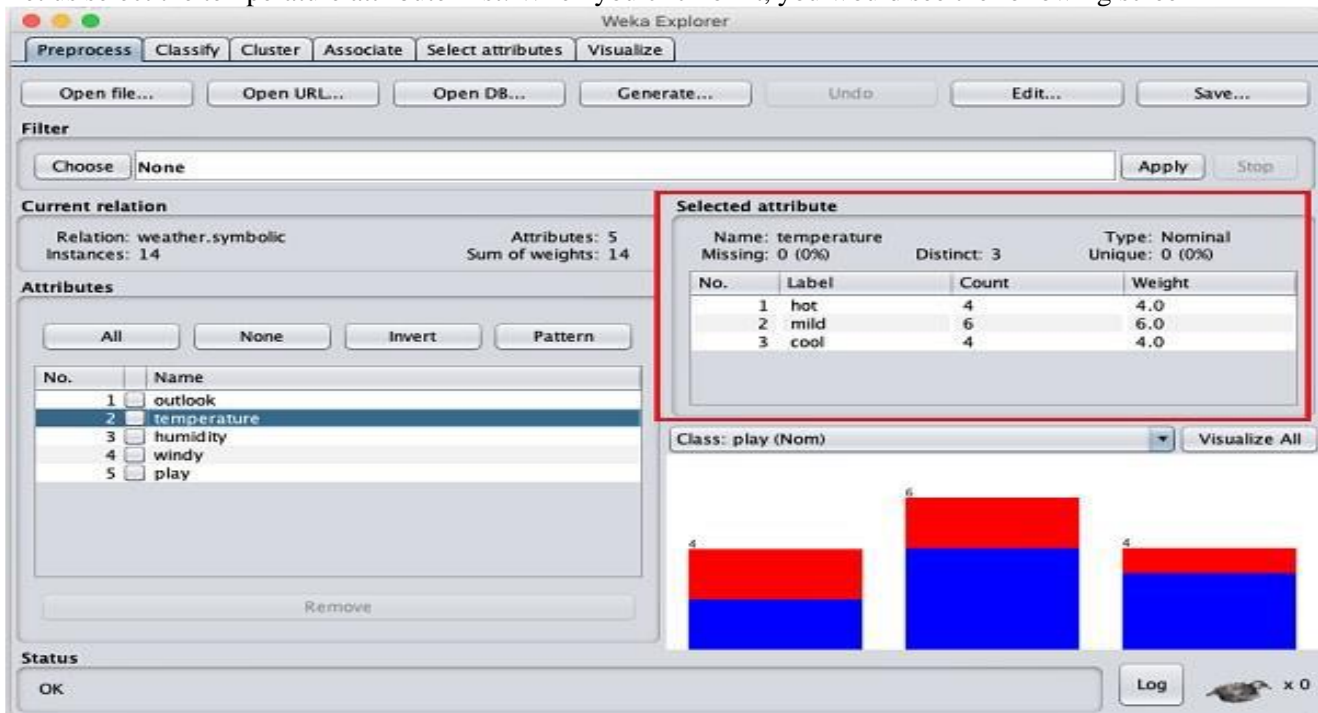
So first open the file by using the **Open file ...** option and select the **weather-numeric.arff** file.

The screenshot displays the WEKA 'Select attributes' window. The 'Current relation' is 'weather' with 14 instances and 5 attributes. The 'Attributes' list includes 'outlook', 'temperature', 'humidity', 'windy', and 'play'. The 'Selected attribute' section shows 'outlook' with 3 distinct values: 'sunny' (5), 'overcast' (4), and 'rainy' (5). The 'Class: play (Nom)' is selected, and a bar chart visualization shows the distribution of 'play' values for each 'outlook' category.

No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

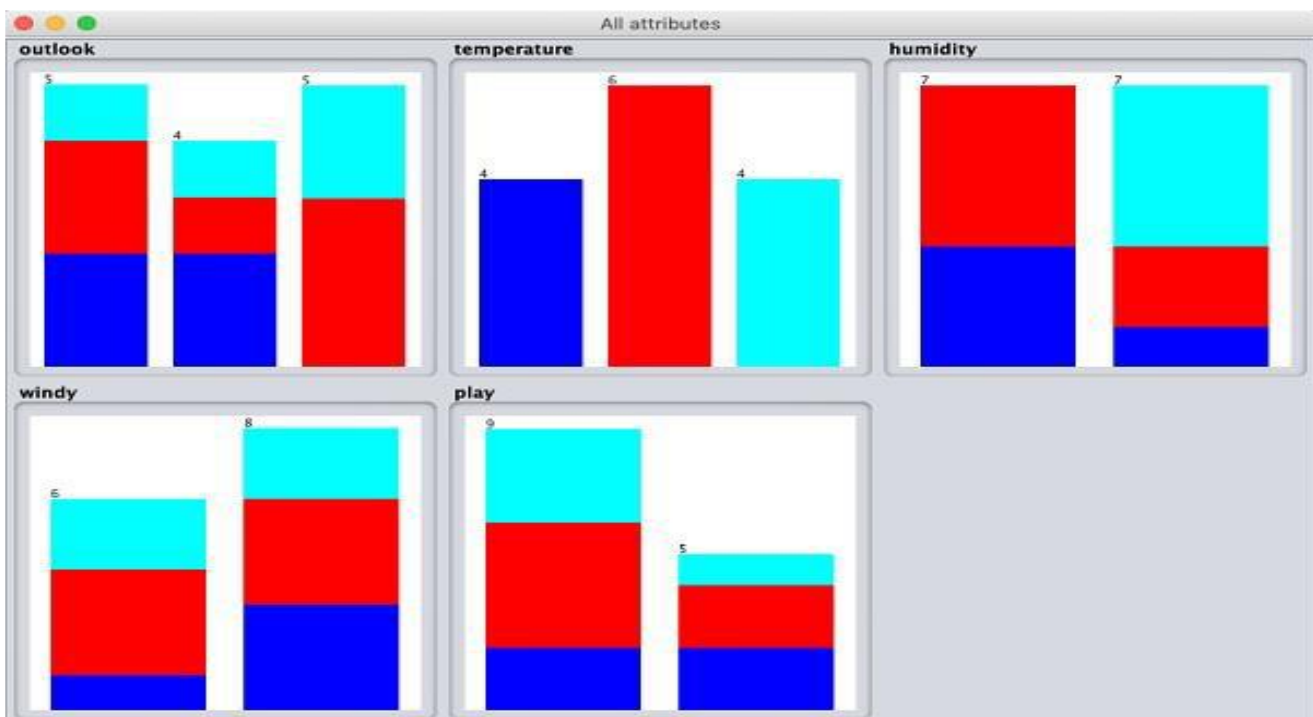
2. The **weather** database contains five fields - outlook, temperature, humidity, windy and play. When you select an attribute from this list by clicking on it, further details on the attribute itself are displayed on the right hand side.

Let us select the temperature attribute first. When you click on it, you would see the following screen –

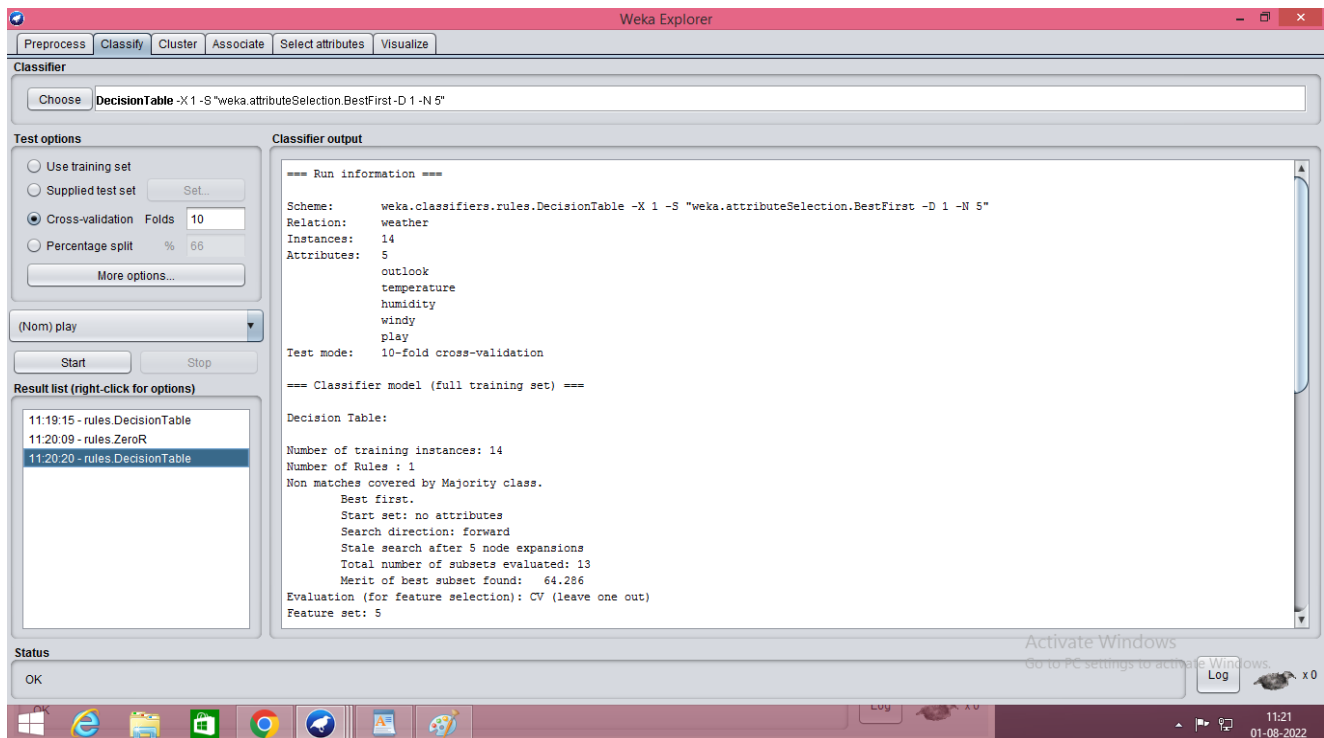


At the bottom of the window, you see the visual representation of the **class** values.

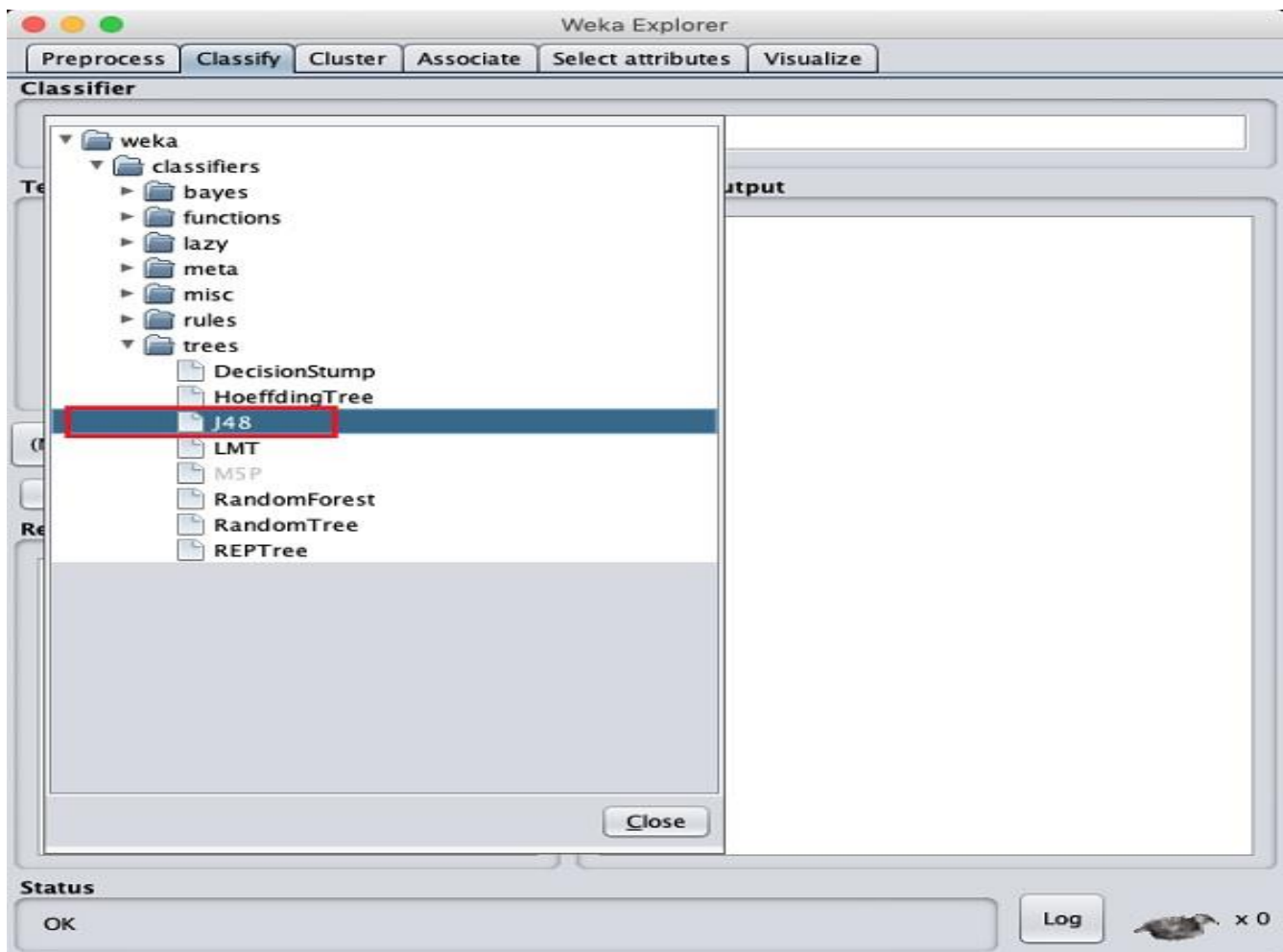
If you click on the **Visualize All** button, you will be able to see all features in one single window as shown here



3. Now to classify the **weather** database select and open classify option

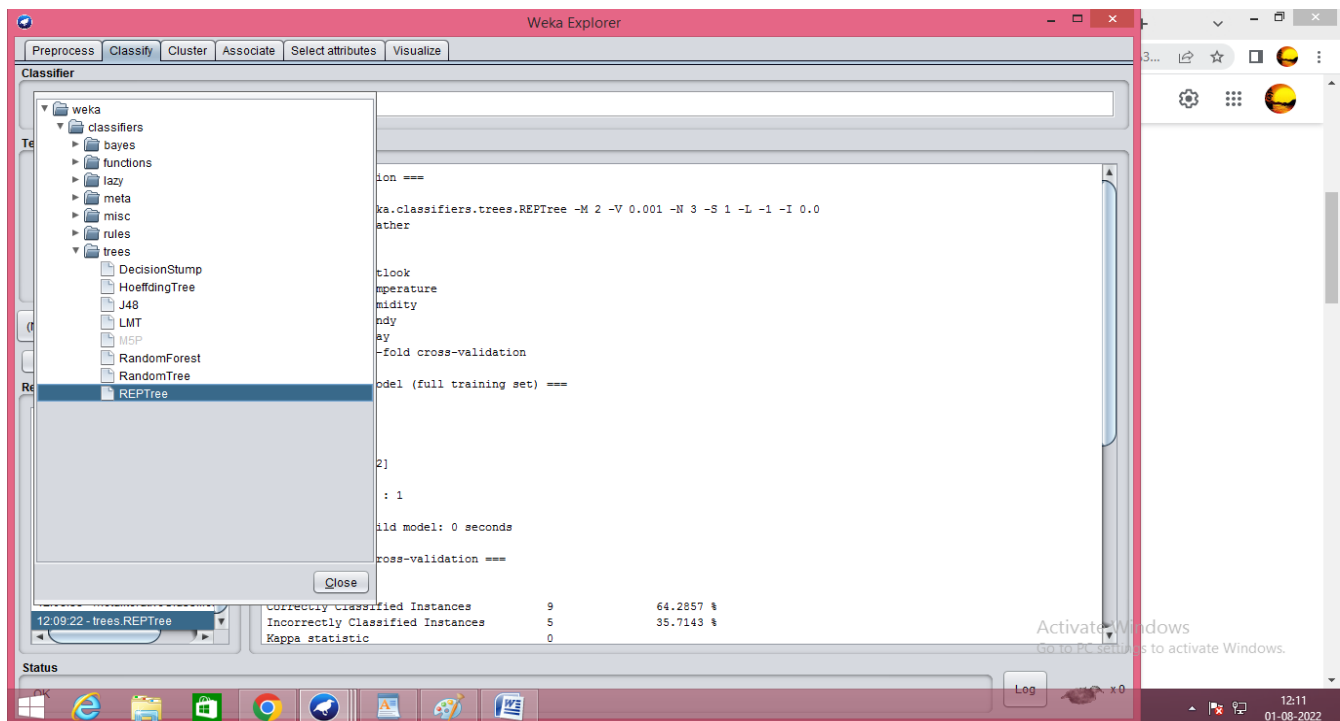


There are many classification methods available in weka software some of them are bayes, meta, trees, lazy, rules ect..



4. According to the problem we are implementing and classifying weather dataset using Decision tree algorithm.

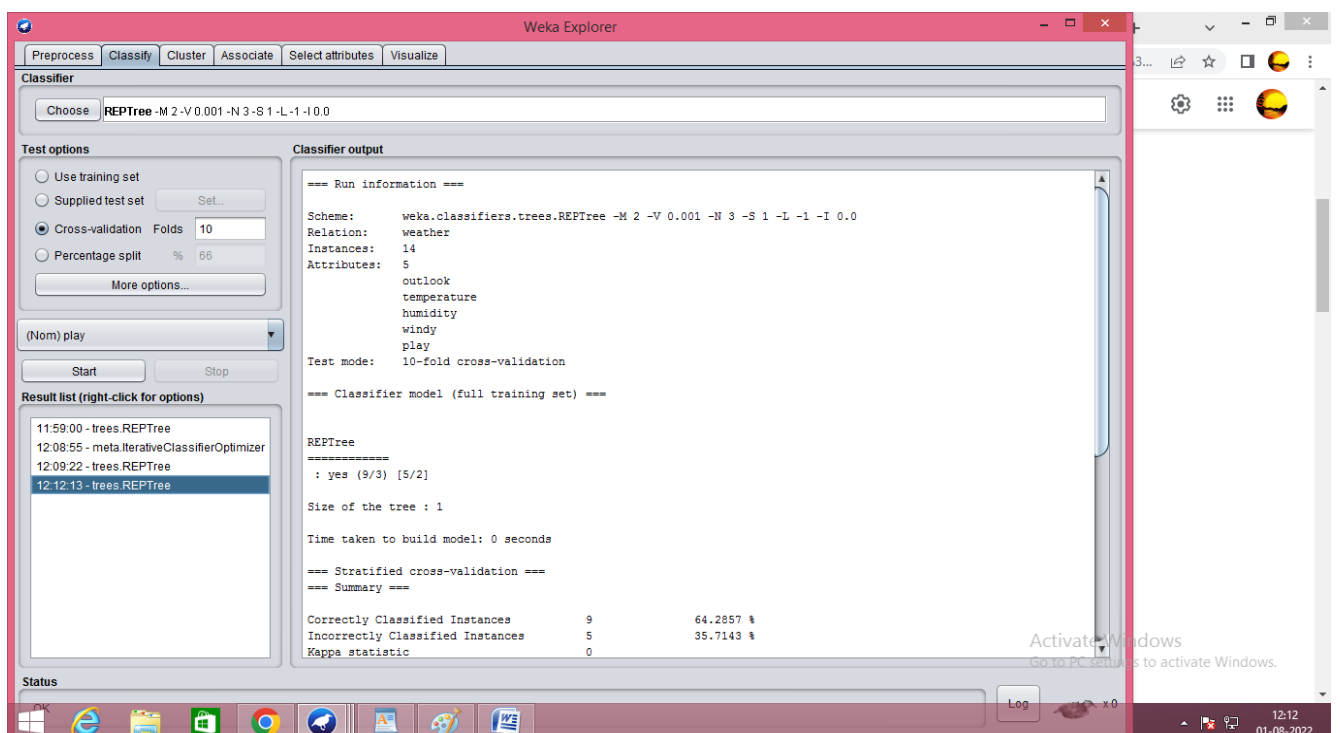
So, select choose and select any one of the classification methods in weka.



5. click on start to apply selected classification method to dataset.

Then it will display all the details like about Attributes, Instances, size of tree Accuracy by class, correlation, confusion Matrix etc..

i) RETTree Classifier



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose REPTree -M 2 -V 0.001 -N 3 -S 1 -L 1 -I 0.0

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

11:59:00 - trees.REPTree

12:08:55 - meta.IterativeClassifierOptimizer

12:09:22 - trees.REPTree

12:12:13 - trees.REPTree

Classifier output

Size of the tree : 1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Metric	Value
Correctly Classified Instances	9 64.2857 %
Incorrectly Classified Instances	5 35.7143 %
Kappa statistic	0
Mean absolute error	0.4725
Root mean squared error	0.4958
Relative absolute error	99.2308 %
Root relative squared error	100.4896 %
Total Number of Instances	14

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
yes	1.000	1.000	0.643	1.000	0.783	?	0.178	0.555	yes
no	0.000	0.000	?	0.000	?	?	0.178	0.318	no
Weighted Avg.	0.643	0.643	?	0.643	?	?	0.178	0.470	

=== Confusion Matrix ===

a b <-- classified as

9 0 | a = yes

5 0 | b = no

Status

Log

12:12 01-08-2022

## ii) Decision Tree Classifier:

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose DecisionTable -X 1 -S "weka.attributeSelection.BestFirst-D 1 -N 5"

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

11:19:15 - rules.DecisionTable

11:20:09 - rules.ZeroR

11:20:20 - rules.DecisionTable

Classifier output

=== Run information ===

Scheme: weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst-D 1 -N 5"

Relation: weather

Instances: 14

Attributes: 5

outlook

temperature

humidity

windy

play

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Decision Table:

Number of training instances: 14

Number of Rules : 1

Non matches covered by Majority class.

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 13

Merit of best subset found: 64.286

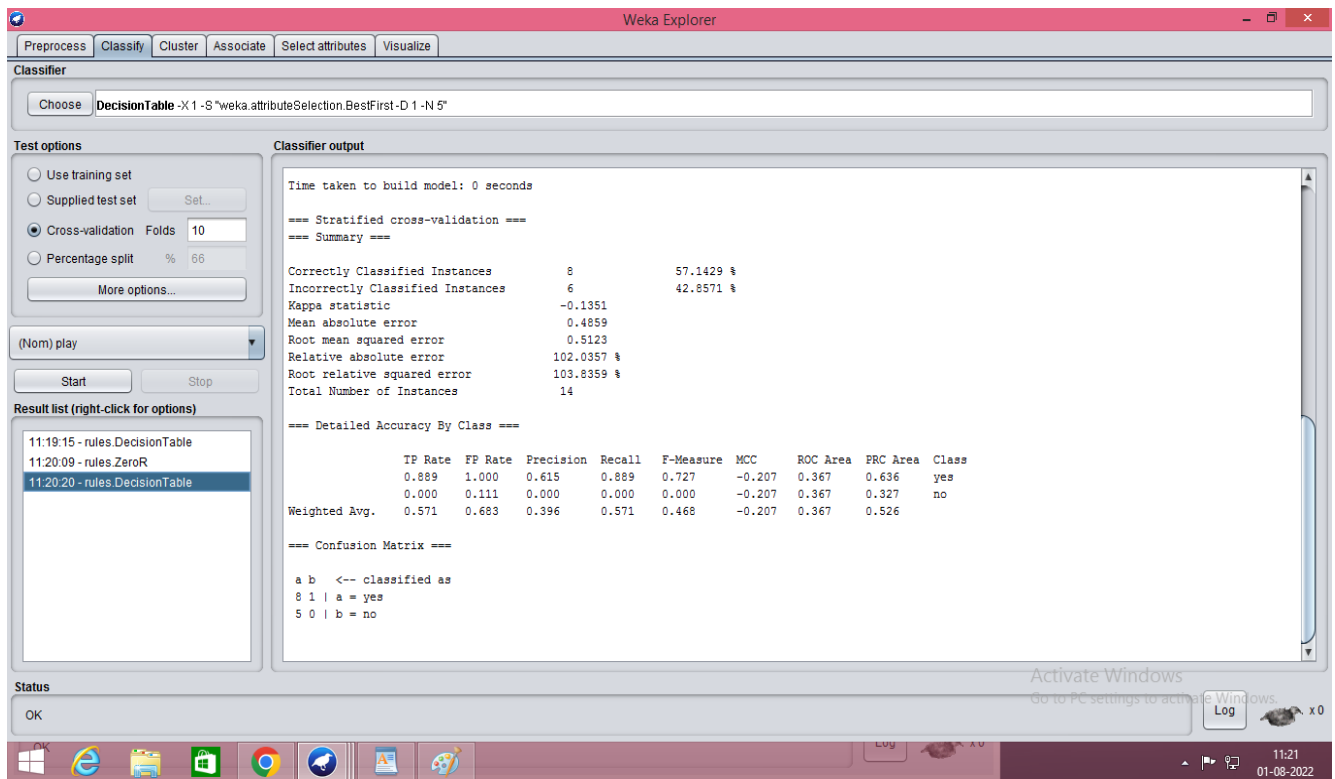
Evaluation (for feature selection): CV (leave one out)

Feature set: 5

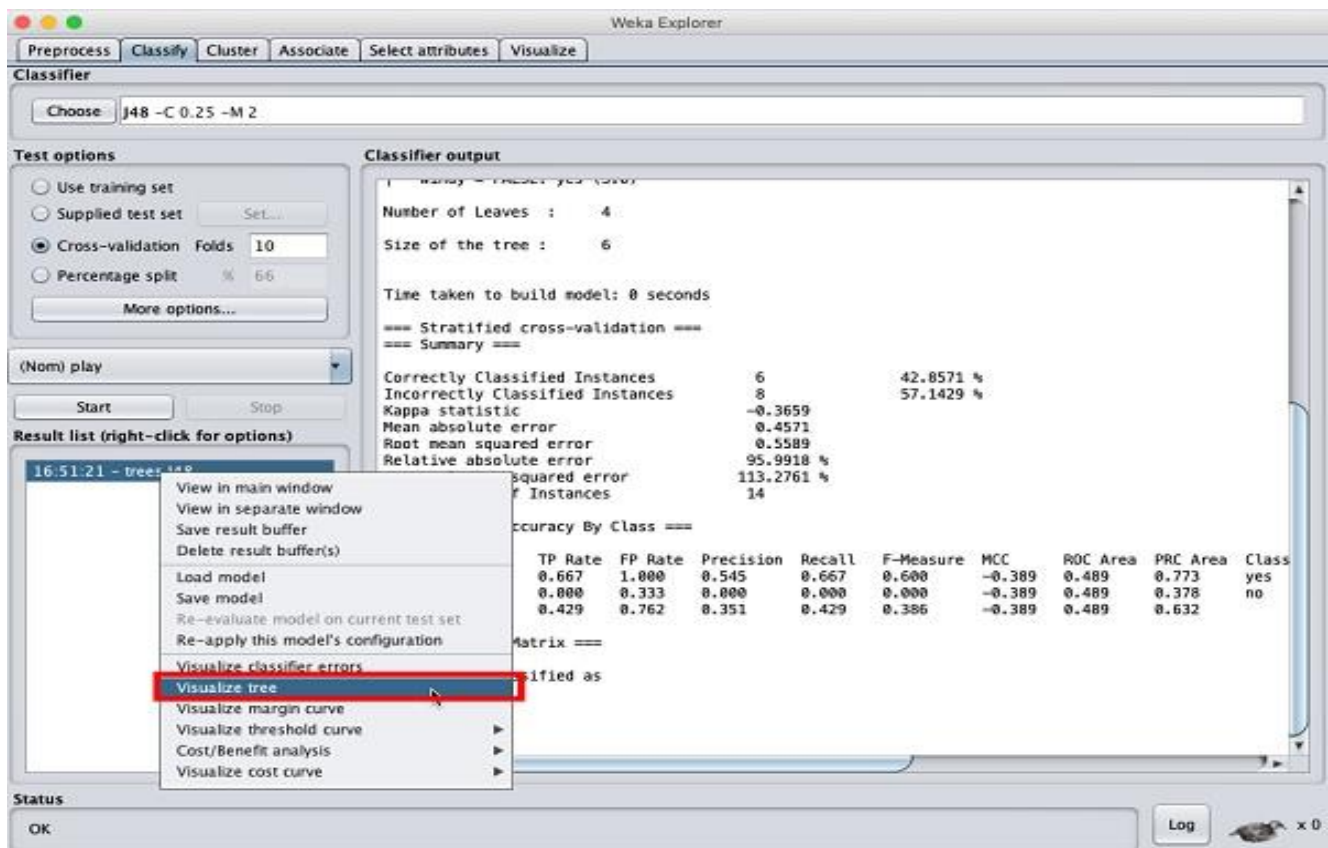
Status

Log

11:21 01-08-2022

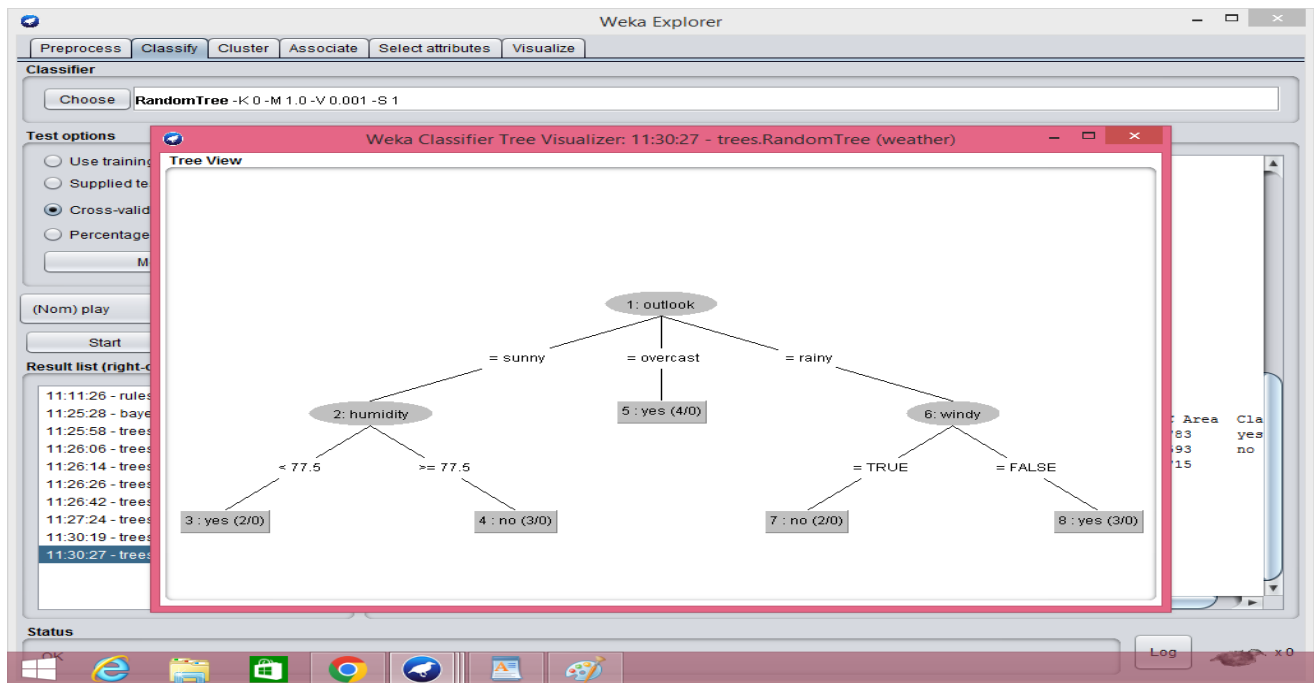


6. If you want to Visualize the tree under start button we have result list in that you can see all the list of classifications you done.



7. Now Right click on one classify method their it will display many options in that select Visualize tree Automatically the tree will display.





8. In weka software Meta Classifier will give best results with high accuracy

Let us classify the weather dataset by using some of the models of meta classifier.

### iii) Iterative classifier Optimizer

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The 'Classifier' dropdown is set to 'IterativeClassifierOptimizer -W weka.classifiers.meta.LogitBoost -L 50 -P 1 -E 1 -I 1 -F 10 -R 1 -percentage 0.0 -metric RMSE -S 1 -- -P 100 -L -1.7976931348623157E308 -H 1.0 -Z 3.0 -O 1'. The 'Test options' section has 'Cross-validation' selected with 'Folds' set to 10. The 'Result list (right-click for options)' shows several classifiers, with 'meta.IterativeClassifierOptimizer' selected. The 'Classifier output' window is open, displaying the following information:

```

=== Run information ===

Scheme:      weka.classifiers.meta.IterativeClassifierOptimizer -W weka.classifiers.meta.LogitBoost -L 50 -P 1 -E 1
Relation:    weather
Instances:    14
Attributes:   5
  outlook
  temperature
  humidity
  windity
  play
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Best value found: 0.5
Best number of iterations found: 0

LogitBoost: No model built yet.

Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      8      57.1429 %
Incorrectly Classified Instances    6      42.8571 %
  
```

The status bar at the bottom shows 'Log' and 'x0'. A watermark 'Activate Windows Go to PC settings to activate Windows.' is visible in the bottom right corner.

The screenshot shows the Weka Explorer interface with the **IterativeClassifierOptimizer** classifier selected. The **Test options** are set to **Cross-validation** with **Folds** set to 10. The **Classifier output** pane displays the following results:

Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===

=== Summary ===

Metric	Value	Percentage
Correctly Classified Instances	8	57.1429 %
Incorrectly Classified Instances	6	42.8571 %
Kappa statistic	-0.1351	
Mean absolute error	0.5534	
Root mean squared error	0.5797	
Relative absolute error	116.2083 %	
Root relative squared error	117.4947 %	
Total Number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.889	1.000	0.615	0.889	0.727	-0.207	0.322	0.606	yes	
0.000	0.111	0.000	0.000	0.000	-0.207	0.322	0.302	no	
Weighted Avg.	0.571	0.683	0.396	0.571	0.468	-0.207	0.322	0.497	

=== Confusion Matrix ===

```

a b  <-- classified as
8 1 | a = yes
5 0 | b = no

```

The **Result list** on the left shows a timeline of classifier evaluations, with **12:14:15 - meta.IterativeClassifierOptimizer** selected.

#### iv) MultiClassClassifier

The screenshot shows the Weka Explorer interface with the **MultiClassClassifier** selected. The **Test options** are set to **Cross-validation** with **Folds** set to 10. The **Classifier output** pane displays the following results:

=== Run information ===

Scheme: weka.classifiers.meta.MultiClassClassifier -M 0 -R 2.0 -S 1 -W weka.classifiers.functions.Logistic -- -R 1.0E-8 -M -1 -num-decimal-places 4

Relation: weather

Instances: 14

Attributes: 5

- outlook
- temperature
- humidity
- windy
- play

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

MultiClassClassifier

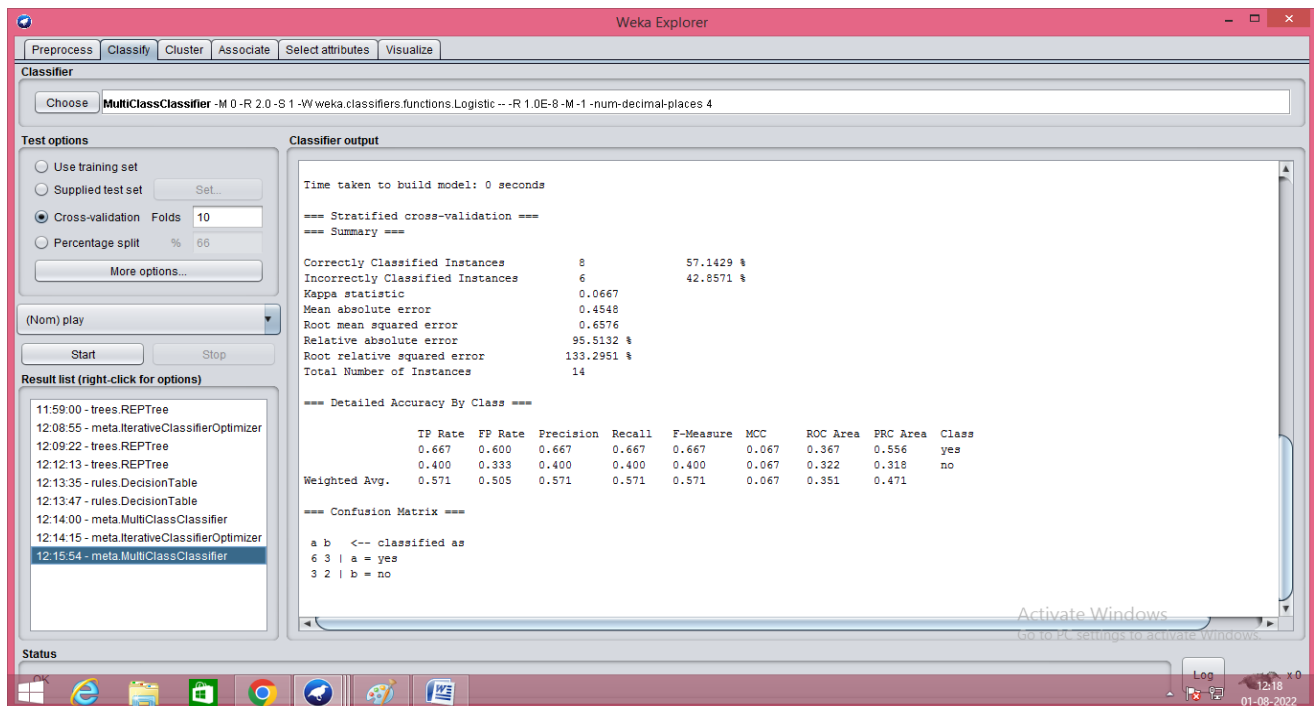
Classifier 1

Logistic Regression with ridge parameter of 1.0E-8

Coefficients...

Variable	Class	Value
outlook=sunny	yes	-6.4257
outlook=overcast	yes	13.5922
outlook=rainy	yes	-5.6562
temperature	yes	-0.0776
humidity	yes	-0.1556
windy=FALSE	yes	3.7317

The **Result list** on the left shows a timeline of classifier evaluations, with **12:15:54 - meta.MultiClassClassifier** selected.

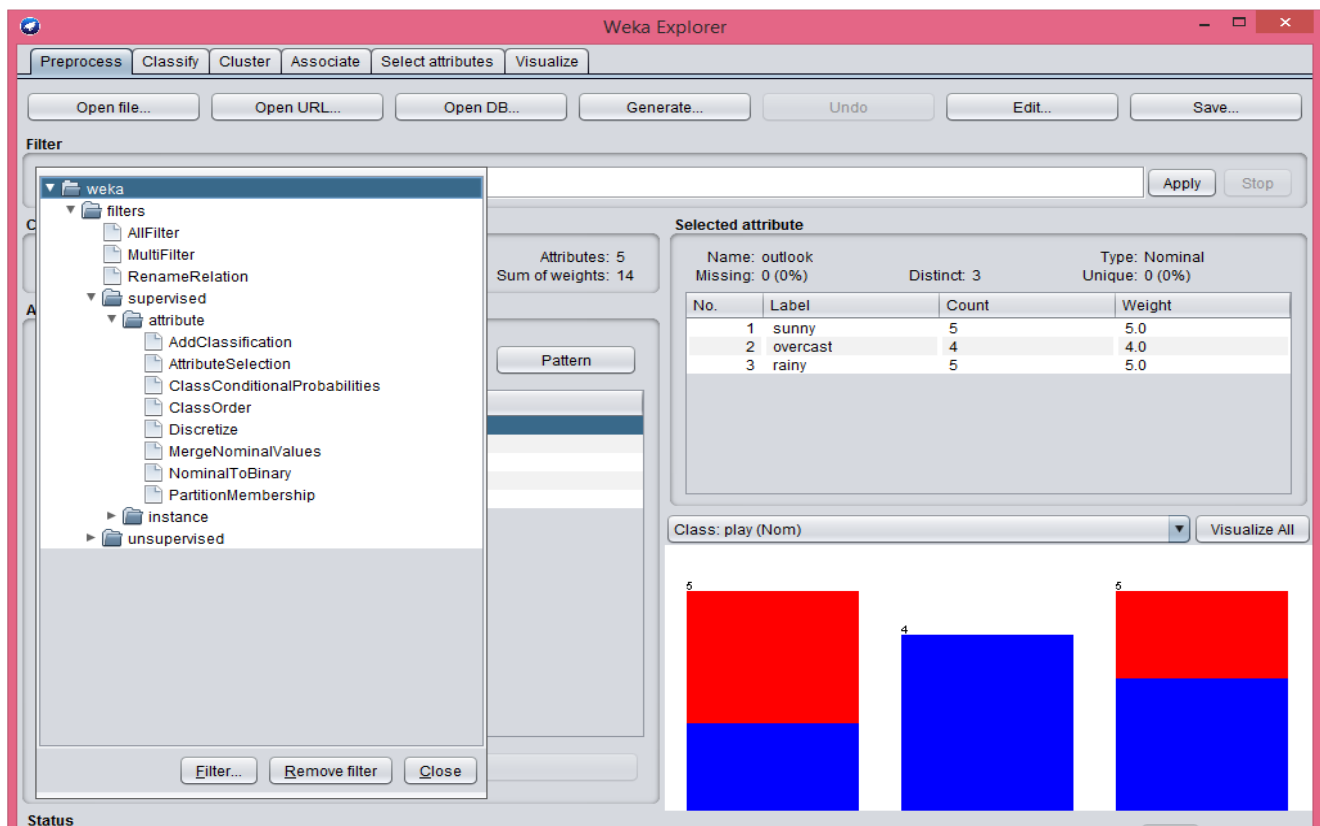


9. All the above classification is done without preprocessing.

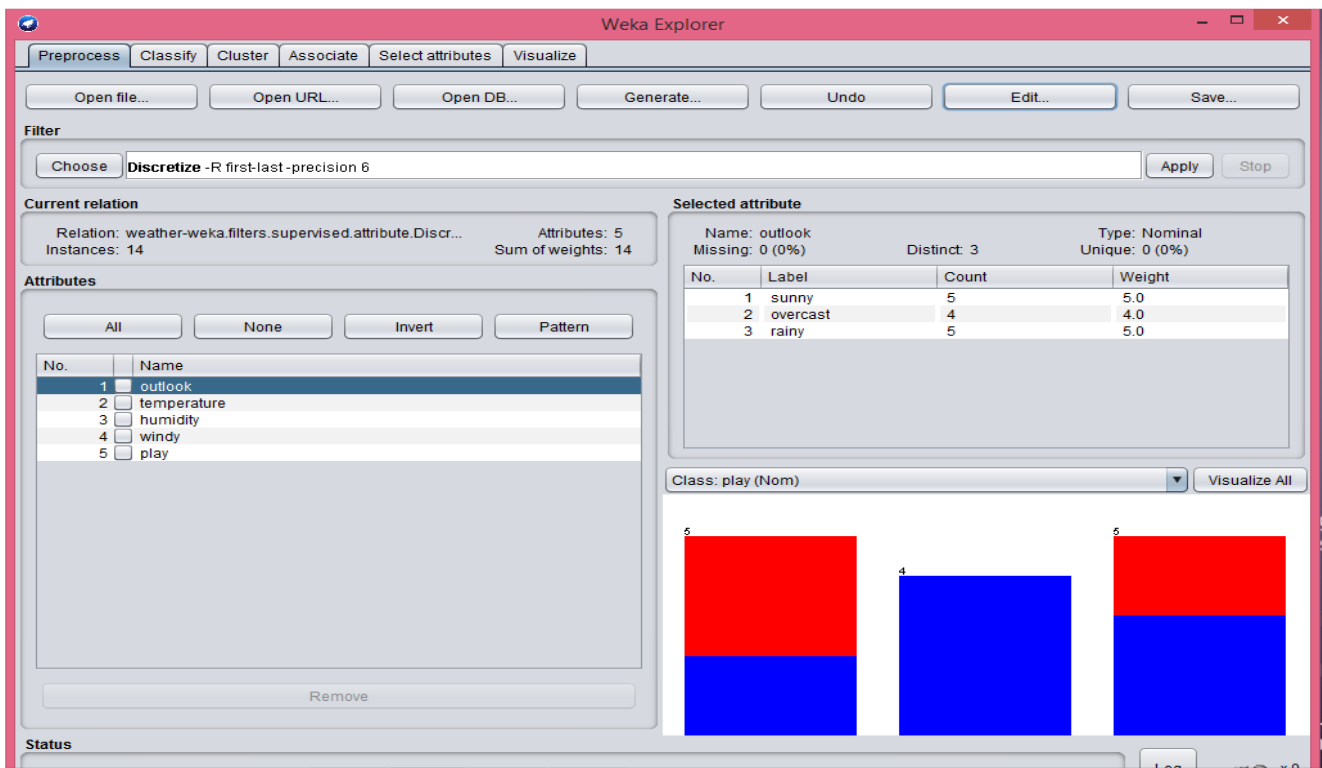
Now we will perform preprocessing on Weather Dataset.

10. Now to perform Discretize preprocessing filter, go to choose option now select the filter option under filter we have different modes of filters to perform preprocessing.

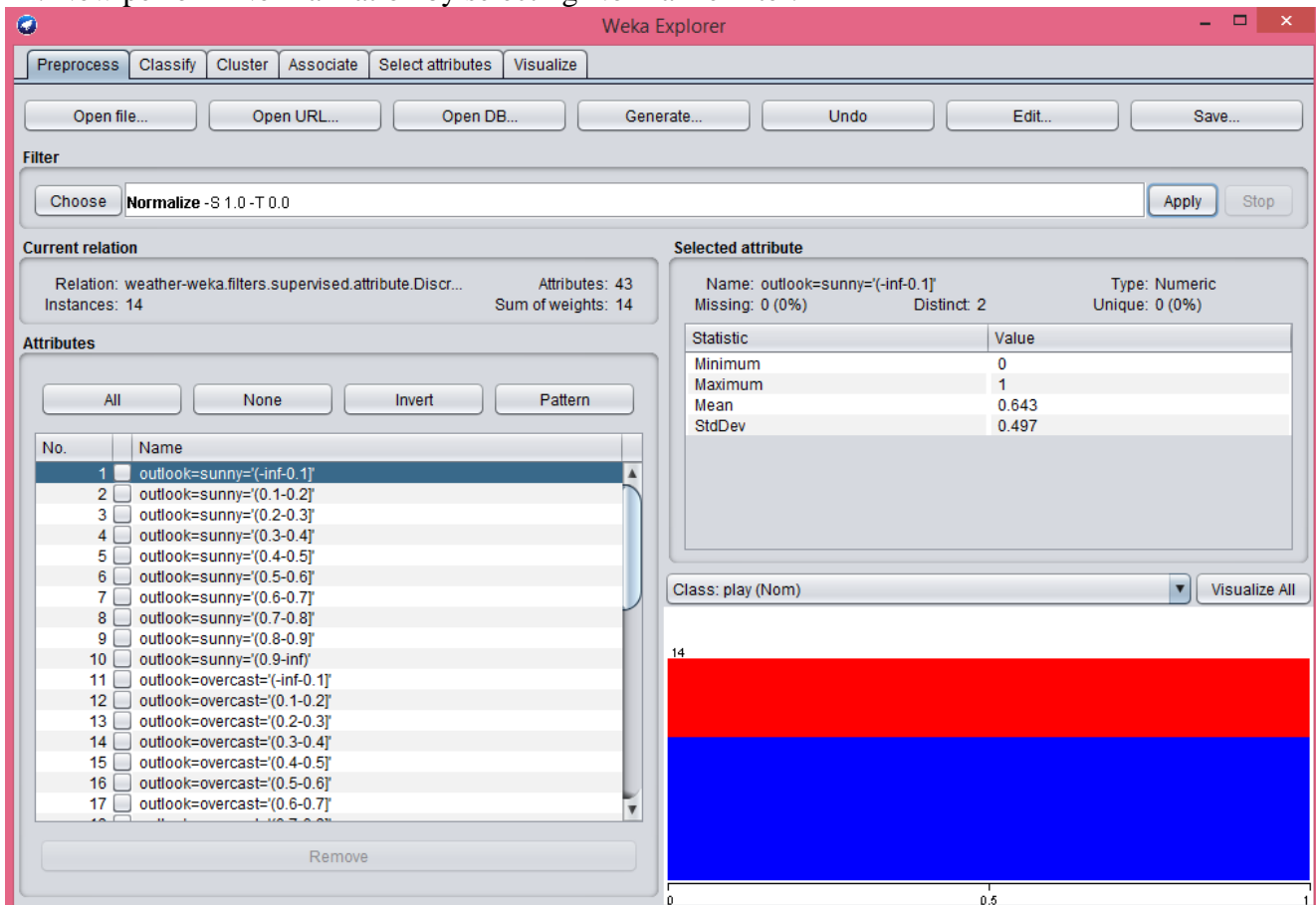
Select the attribute under supervised there we have a Discretize option choose that.



11. After choosing Discretize option click on Apply option to apply the filter to the dataset.



12. Now perform Normalization by selecting Normalize filter.



13. We performed preprocessing on weather dataset, now again classify the dataset by using different classification models to get accurate results.

i) REPTree Classifier

The screenshot shows the Weka Explorer interface with the REPTree classifier selected. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' section displays the following results:

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Metric	Value
Correctly Classified Instances	8
Incorrectly Classified Instances	6
Kappa statistic	-0.1351
Mean absolute error	0.5082
Root mean squared error	0.5352
Relative absolute error	106.7308 %
Root relative squared error	108.4859 %
Total Number of Instances	14

=== Detailed Accuracy By Class ===

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
yes	0.889	1.000	0.615	0.889	0.727	-0.207	0.100	0.538
no	0.000	0.111	0.000	0.000	0.000	-0.207	0.100	0.283
Weighted Avg.	0.571	0.683	0.396	0.571	0.468	-0.207	0.100	0.447

=== Confusion Matrix ===

```

a b  <-- classified as
8 1  | a = yes
5 0  | b = no

```

The 'Result list' on the left shows a list of classifiers, with 'trees.REPTree' selected at the bottom.

ii) Iterative Classifier Optimizer

The screenshot shows the Weka Explorer interface with the IterativeClassifierOptimizer selected. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' section displays the following results:

=== Run information ===

Scheme: weka.classifiers.meta.IterativeClassifierOptimizer -W weka.classifiers.meta.LogitBoost -L 50 -P 1 -E 1 -I 1 -F 10 -R 1 -percentage 0.0 -metric RMSE -S 1 -- -P 100 -L -1.7976931348623157E308 -H 1.0 -Z 3.0 -O 1 -E 1 -S 1 -I 10 -W weka.classifiers

Relation: weather-weka.filters.supervised.attribute.Discretize-Rfirst-last-precision6-weka.filters.supervised.attribute.Discretize-Rfirst-last-precision6

Instances: 14

Attributes: 5

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Best value found: 0.5

Best number of iterations found: 0

LogitBoost: No model built yet.

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===

=== Summary ===

Metric	Value
Correctly Classified Instances	6
Incorrectly Classified Instances	8
Kappa statistic	-0.1351
Mean absolute error	0.5082
Root mean squared error	0.5352
Relative absolute error	106.7308 %
Root relative squared error	108.4859 %
Total Number of Instances	14

The 'Result list' on the left shows a list of classifiers, with 'meta.IterativeClassifierOptimizer' selected at the bottom.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose **IterativeClassifierOptimizer** -W weka.classifiers.meta.LogitBoost-L 50 -P 1 -E 1 -I 1 -F 10 -R 1 -percentage 0.0 -metric RMSE -S 1 -- -P 100 -L 1.7976931348623157E308 -H 1.0 -Z 3.0 -O 1 -E 1 -S 1 -I 10 -W weka.classifiers.tree

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

- 11:59:00 - trees.REPTree
- 12:08:55 - meta.IterativeClassifierOptimizer
- 12:09:22 - trees.REPTree
- 12:12:13 - trees.REPTree
- 12:13:35 - rules.DecisionTable
- 12:13:47 - rules.DecisionTable
- 12:14:00 - meta.MultiClassClassifier
- 12:14:15 - meta.IterativeClassifierOptimizer
- 12:15:54 - meta.MultiClassClassifier
- 12:19:37 - meta.MultiClassClassifier
- 12:19:51 - meta.MultiClassClassifier
- 12:21:59 - trees.REPTree
- 12:31:10 - meta.IterativeClassifierOptimizer

Classifier output

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	6	42.8571 %
Incorrectly Classified Instances	8	57.1429 %
Kappa statistic	-0.3659	
Mean absolute error	0.5466	
Root mean squared error	0.609	
Relative absolute error	114.7863 %	
Root relative squared error	123.4378 %	
Total Number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.667	1.000	0.545	0.667	0.600	-0.389	0.311	0.669	yes
	0.000	0.333	0.000	0.000	0.000	-0.389	0.311	0.305	no
Weighted Avg.	0.429	0.762	0.351	0.429	0.386	-0.389	0.311	0.539	

=== Confusion Matrix ===

a b <-- classified as

6 3 | a = yes

5 0 | b = no

Status

Log 12:31 01-08-2022

### iii) MultiClass Classifier:

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose **MultiClassClassifier** -M 0 -R 2.0 -S 1 -W weka.classifiers.functions.Logistic -- -R 1.0E-8 -M -1 -num-decimal-places 4

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

- 11:59:00 - trees.REPTree
- 12:08:55 - meta.IterativeClassifierOptimizer
- 12:09:22 - trees.REPTree
- 12:12:13 - trees.REPTree
- 12:13:35 - rules.DecisionTable
- 12:13:47 - rules.DecisionTable
- 12:14:00 - meta.MultiClassClassifier
- 12:14:15 - meta.IterativeClassifierOptimizer
- 12:15:54 - meta.MultiClassClassifier
- 12:19:37 - meta.MultiClassClassifier
- 12:19:51 - meta.MultiClassClassifier

Classifier output

=== Run information ===

Scheme: weka.classifiers.meta.MultiClassClassifier -M 0 -R 2.0 -S 1 -W weka.classifiers.functions.Logistic -- -R 1.0E-8 -M -1 -num-decimal-places 4

Relation: weather-weka.filters.supervised.attribute.Discretize-Rfirst-last-precision6-weka.filters.supervised.attribute.Discretize-Rfirst-last

Instances: 14

Attributes: 5

outlook

temperature

humidity

windy

play

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

MultiClassClassifier

Classifier 1

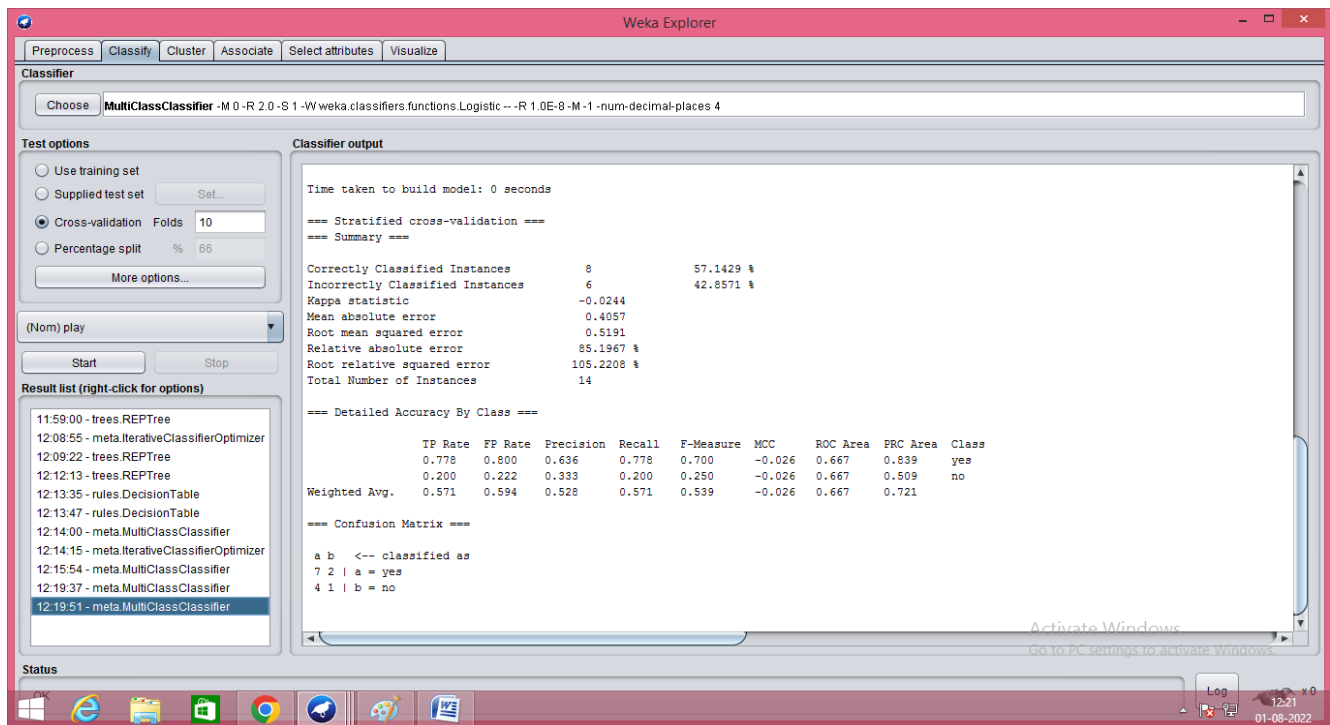
Logistic Regression with ridge parameter of 1.0E-8

Coefficients...

Variable	Class	yes
outlook=sunny		-6.0889
outlook=overcast		12.5872
outlook=rainy		-5.0997
windy=FALSE		1.8945
Intercept		4.4343

Status

Log 12:21 01-08-2022



### 5.1.4 Results and Discussion:

Implementing the Classification using Decision Tree algorithm on 'Weather' dataset is successfully completed by drawing the confusion matrix and report the model with accuracy.

We observed that when we classify the dataset by using Decision tree and other different classifier methods we get best and high accuracy in Meta classification models which is best classifier in weka software, Again we perform preprocessing on weather dataset and perform classification, by this we conclude that on preprocessed dataset when we perform classification it produce accurate and high results.

### 5.2.1 Problem Statement:

Implement Bayesian Classification and analyze the results on 'iris' Dataset.

### 5.2.2 Description:

#### About Dataset used

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
6. Number of samples of each species of iris flowers:
7. Predicted attribute: class of iris plant.
8. Missing Attribute Values: None

The Iris Dataset contains information of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

Data Set Characteristics: Multivariate

Area: Life Sciences

Number of samples (or instances) in the dataset: 150

Number of attributes (or features): 05 Attribute Information:

- Iris Setosa
- Iris Versicolour
- Iris Virginica

Class Distribution: 33.3% for each of 3 classes.

#### About Arff:

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

ARFF files have two distinct sections. The first section is the **Header** information, which is followed the **Data** information.

The **Header** of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types.

Lines that begin with a % are comments. The **@RELATION**, **@ATTRIBUTE** and **@DATA** declarations are case insensitive.

#### About CSV:

Files with .csv (Comma Separated Values) extension represent plain text files that contain records of data with comma separated values. Each line in a CSV file is a new record from the set of records contained in the file. Such files are generated when data transfer is intended from one storage system to another. Since all applications can recognize records separated by comma, import of such data files to database is done very conveniently.



Almost all spreadsheet applications such as Microsoft Excel or OpenOffice Calc can import CSV without much effort. Data imported from such files is arranged in cells of a spreadsheet for representation to user.

### Datatypes that are supported by Weka:

- numeric
- integer is treated as numeric
- real is treated as numeric
- [nominal-specification]
- string
- date [date-format]
- relational for multi-instance data (for future use)

where [nominal-specification] and [date-format] are defined below. The keywords **numeric**, **real**, **integer**, **string** and **date** are case insensitive.

Numeric attributes

Numeric attributes can be real or integer numbers.

Nominal attributes

Nominal values are defined by providing an [nominal-specification] listing the possible values: {[nominal-name1], [nominal-name2], [nominal-name3], ...}

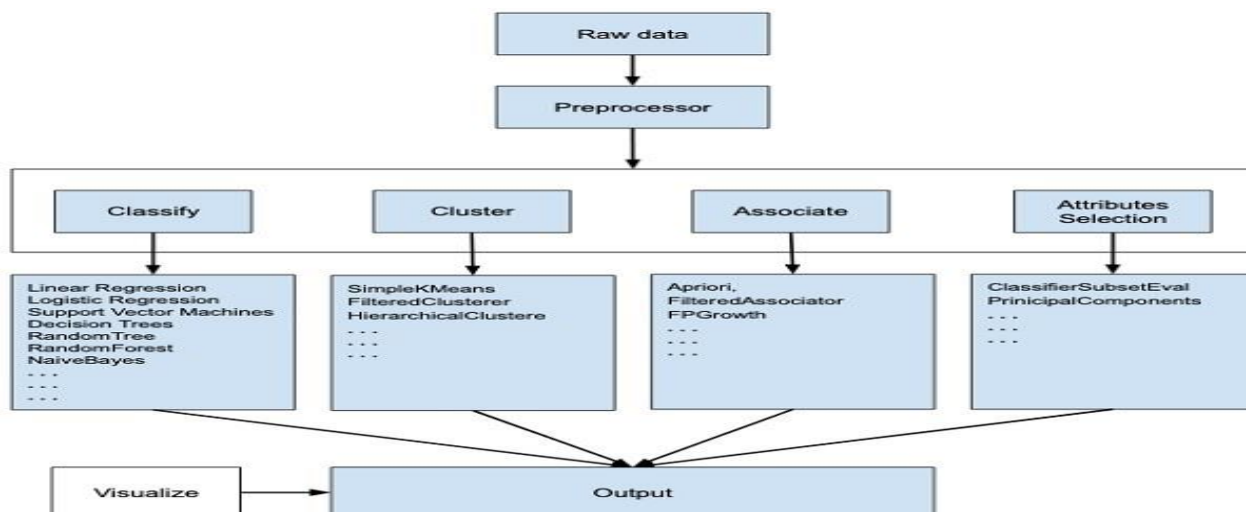
For example, the class value of the Iris dataset can be defined as follows:

```
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
```

Values that contain spaces must be quoted.

### About WEKA Software:

WEKA - an opensource software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems. What WEKA offers is summarized in the following diagram –



If you observe the beginning of the flow of the image, you will understand that there are many stages in dealing with Big Data to make it suitable for machine learning –

First, you will start with the raw data collected from the field. This data may contain several null values and irrelevant fields. You use the data preprocessing tools provided in WEKA to cleanse the data.

Then, you would save the preprocessed data in your local storage for applying ML algorithms.

Next, depending on the kind of ML model that you are trying to develop you would select one of the options such as **Classify**, **Cluster**, or **Associate**. The **Attributes Selection** allows the automatic selection of features to create a reduced dataset.

Note that under each category, WEKA provides the implementation of several algorithms. You would select an algorithm of your choice, set the desired parameters and run it on the dataset.

Then, WEKA would give you the statistical output of the model processing. It provides you a visualization tool to inspect the data.

The various models can be applied on the same dataset. You can then compare the outputs of different models and select the best that meets your purpose.

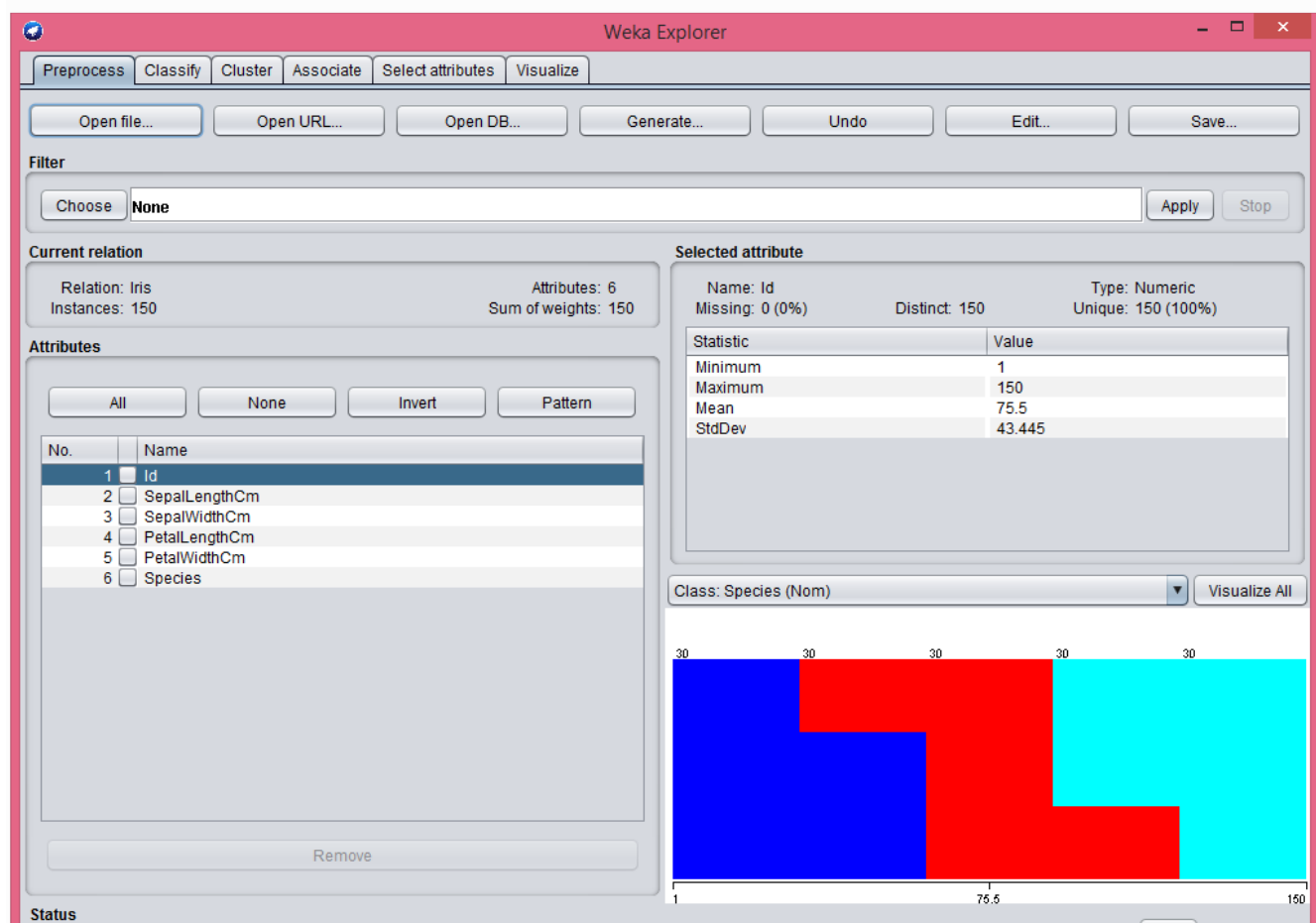
Thus, the use of WEKA results in a quicker development of machine learning models on the whole.

Now that we have seen what WEKA is and what it does, in the next chapter let us learn how to install WEKA on your local computer.

### 5.2.3 Steps to implement the classification:

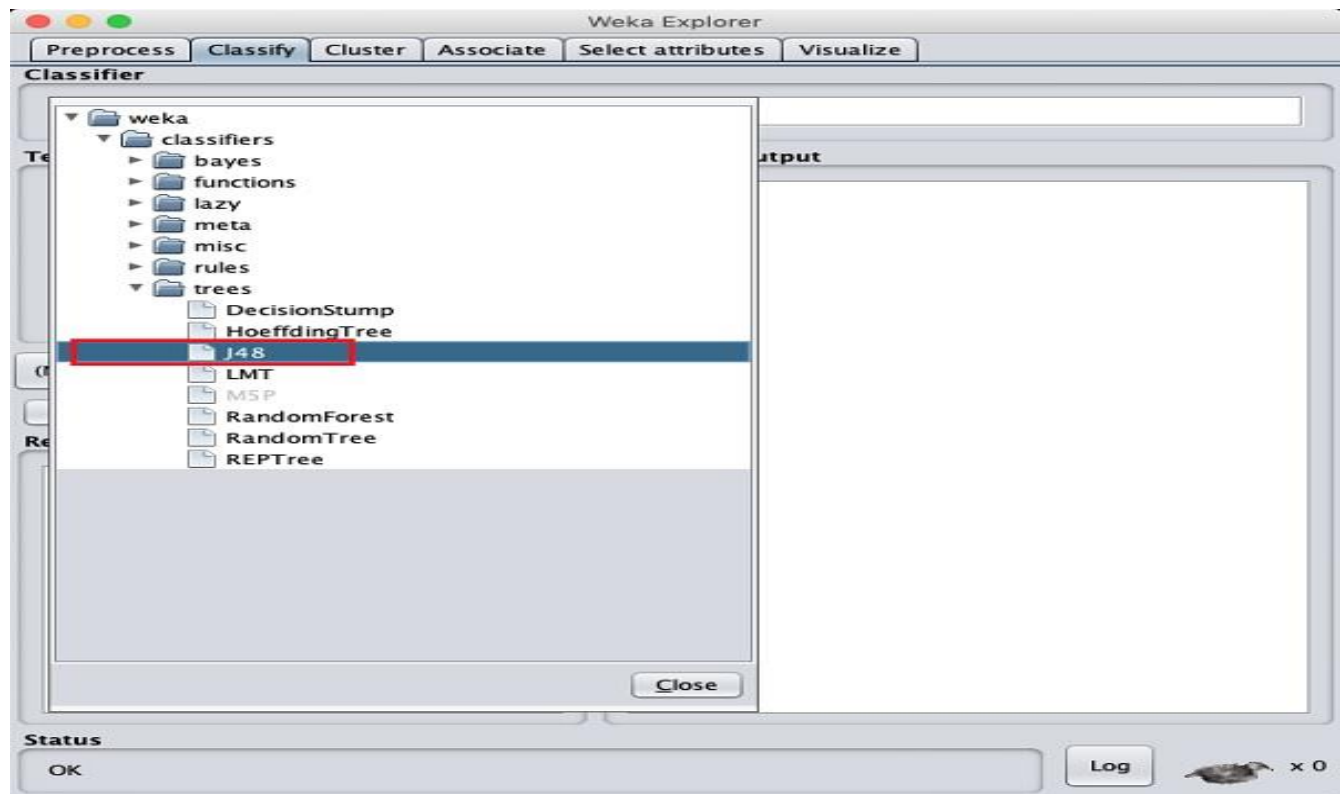
1.To classify the Iris database using Decision Tree algorithm.

So first open the file by using the **Open file ...** option and select the **Iris.arff** file.



2. Now to classify the **Iris** database select and open classify option.

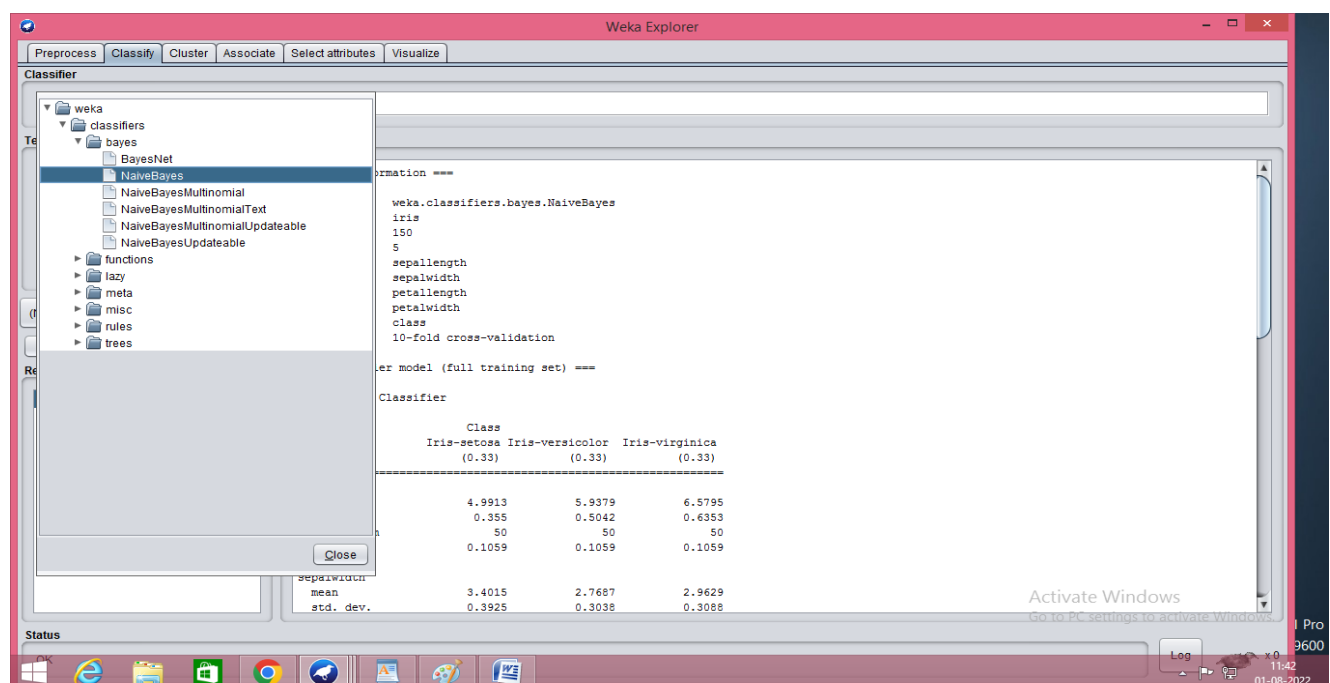
There are many classification methods are available in weka software some of them are bayes, meta, trees, lazy, rules ect..



3. According to the problem we are implementing and classifying Iris dataset using Bayesian Classification

So, select choose and select bayes under bayes select an optimized Bayesian classification model.

- **Naïve Bayes:**



4. To apply the classification to dataset click on start button

Then it will display all the details like about Attributes, Instances, size of tree Accuracy by class, correlation, confusion Matrix etc..

The screenshot shows the Weka Explorer interface with the NaiveBayes classifier selected. The 'Test options' section on the left has 'Cross-validation' selected with 'Folds' set to 10. The 'Classifier output' section on the right displays the following information:

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes  
Relation: iris  
Instances: 150  
Attributes: 5  
sepalength  
sepalwidth  
petallength  
petalwidth  
class

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

Attribute	Iris-setosa (0.33)	Iris-versicolor (0.33)	Iris-virginica (0.33)
sepalength			
mean	4.9913	5.9379	6.5795
std. dev.	0.355	0.5042	0.6353
weight sum	50	50	50
precision	0.1059	0.1059	0.1059
sepalwidth			
mean	3.4015	2.7687	2.9629
std. dev.	0.3925	0.3038	0.3088

The 'Result list' on the left shows two entries: '11:41:28 - bayes.NaiveBayes' and '11:43:38 - bayes.NaiveBayes'. The 'Status' bar at the bottom indicates 'Log' and '11:44 01-08-2022'.

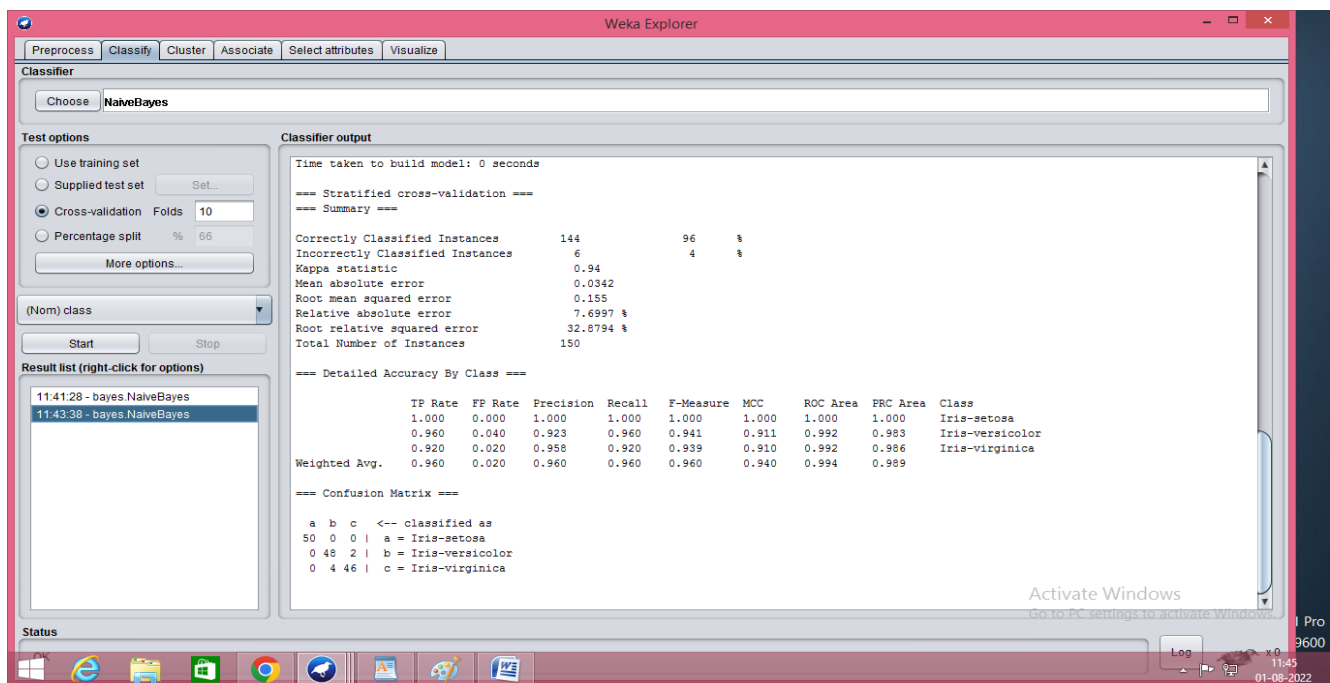
The screenshot shows the Weka Explorer interface with the NaiveBayes classifier selected. The 'Test options' section on the left has 'Cross-validation' selected with 'Folds' set to 10. The 'Classifier output' section on the right displays the following information:

=== Classifier model (full training set) ===

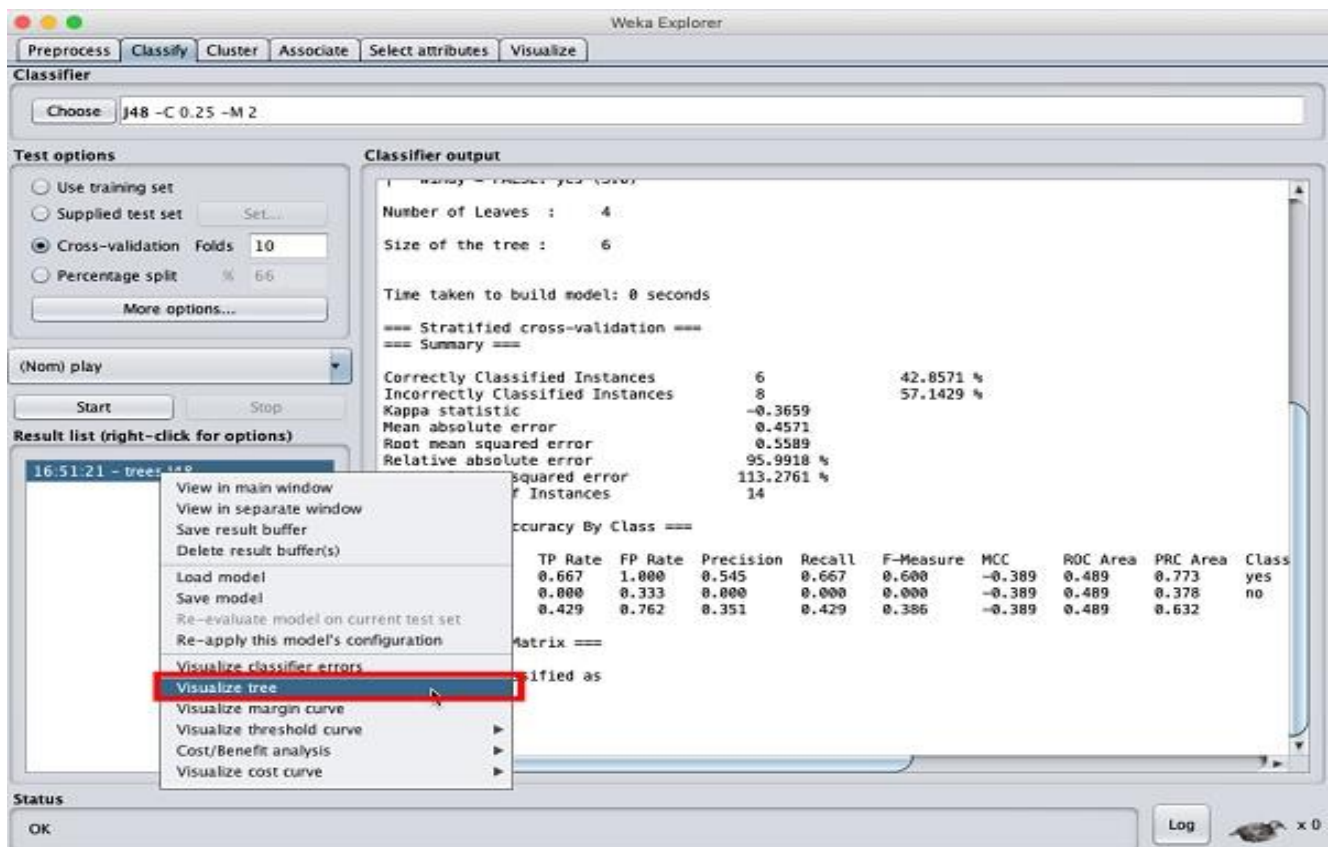
Naive Bayes Classifier

Attribute	Iris-setosa (0.33)	Iris-versicolor (0.33)	Iris-virginica (0.33)
sepalength			
mean	4.9913	5.9379	6.5795
std. dev.	0.355	0.5042	0.6353
weight sum	50	50	50
precision	0.1059	0.1059	0.1059
sepalwidth			
mean	3.4015	2.7687	2.9629
std. dev.	0.3925	0.3038	0.3088
weight sum	50	50	50
precision	0.1091	0.1091	0.1091
petallength			
mean	1.4694	4.2452	5.5516
std. dev.	0.1782	0.4712	0.5529
weight sum	50	50	50
precision	0.1405	0.1405	0.1405
petalwidth			
mean	0.2743	1.3097	2.0343
std. dev.	0.1096	0.1915	0.2646
weight sum	50	50	50
precision	0.1143	0.1143	0.1143

The 'Result list' on the left shows two entries: '11:41:28 - bayes.NaiveBayes' and '11:43:38 - bayes.NaiveBayes'. The 'Status' bar at the bottom indicates 'Log' and '11:44 01-08-2022'.



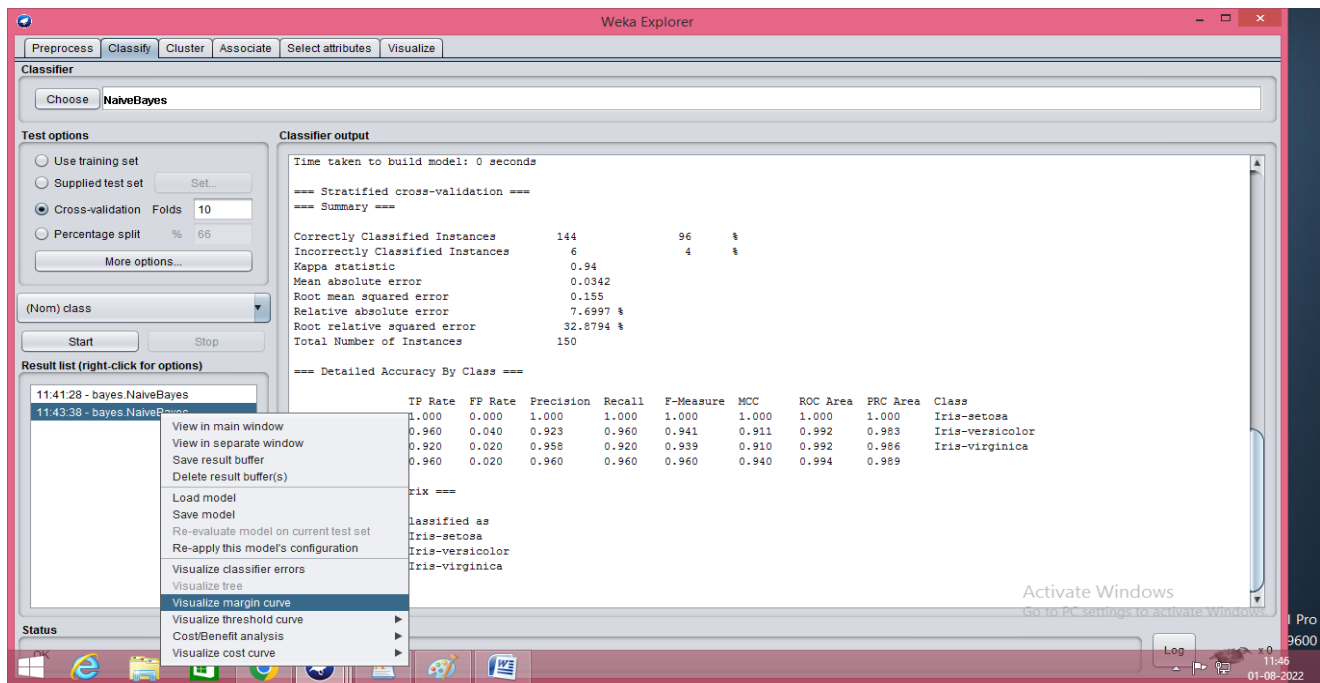
5. If you want to Visualize the classification under start button we have result list in that you can see all the list of classifications you done.



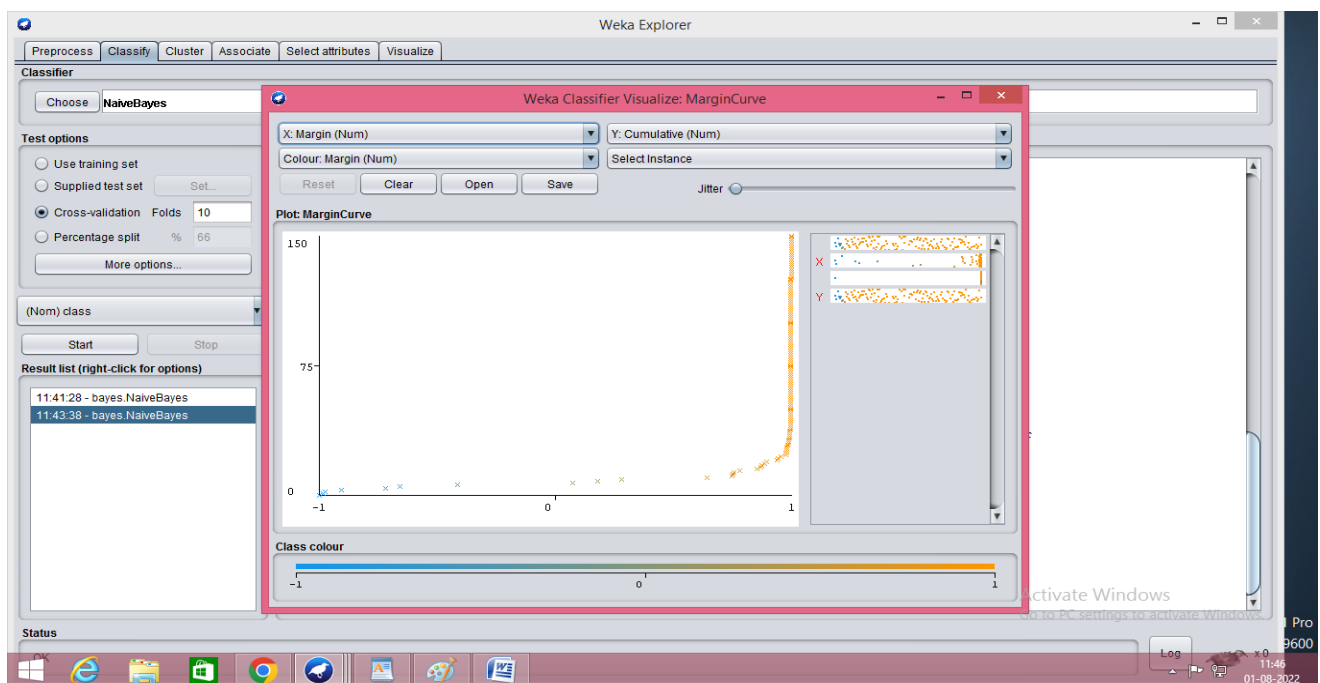
6. Now Right click on one classify model their it will display many options like Visualize tree, visualize margin curve, visualize threshold curve ect..

7. The classification model we used according to that option are unable.

8. Now select on Visualize Margin curve.



9. Automatically curve will display in another window.

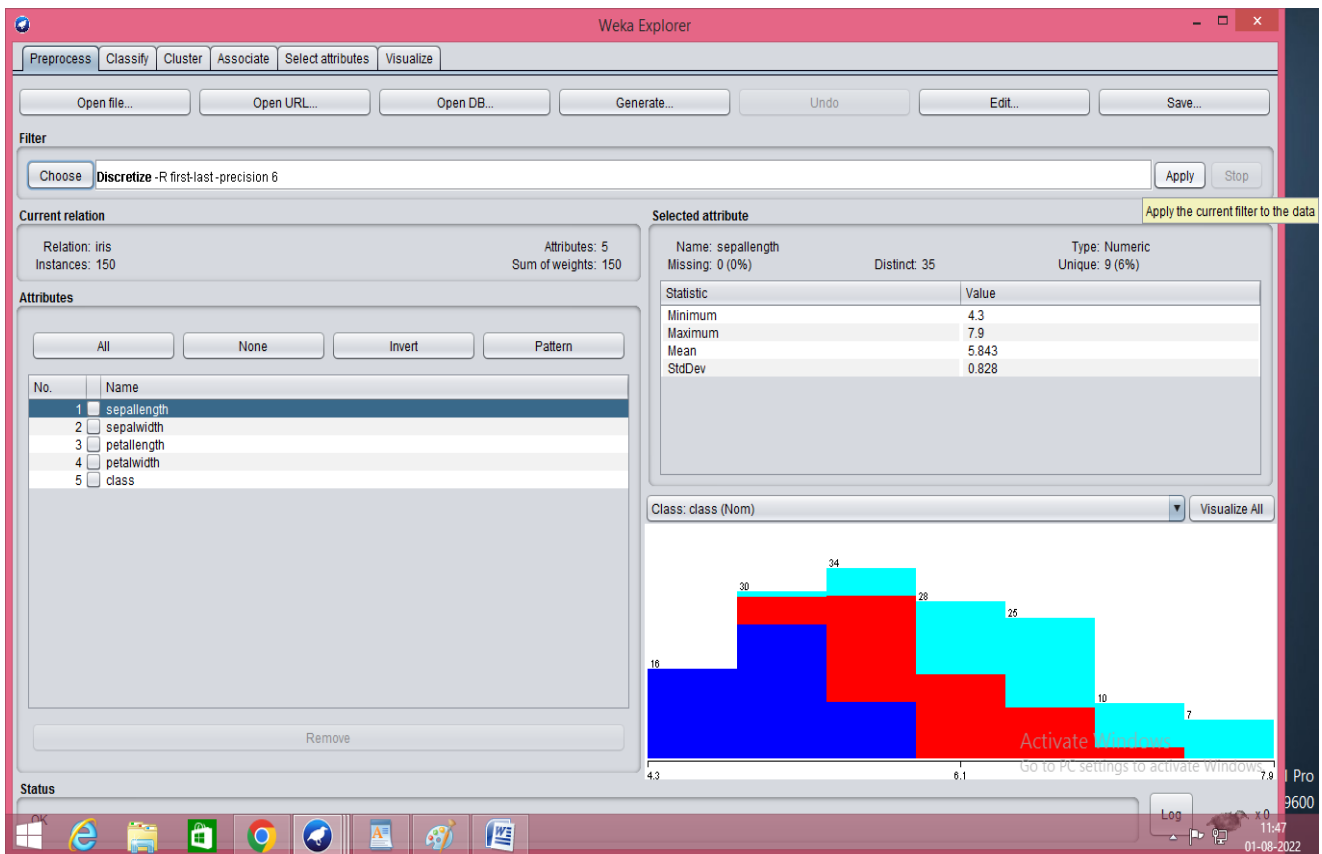


10. All the above classification is done without preprocessing.

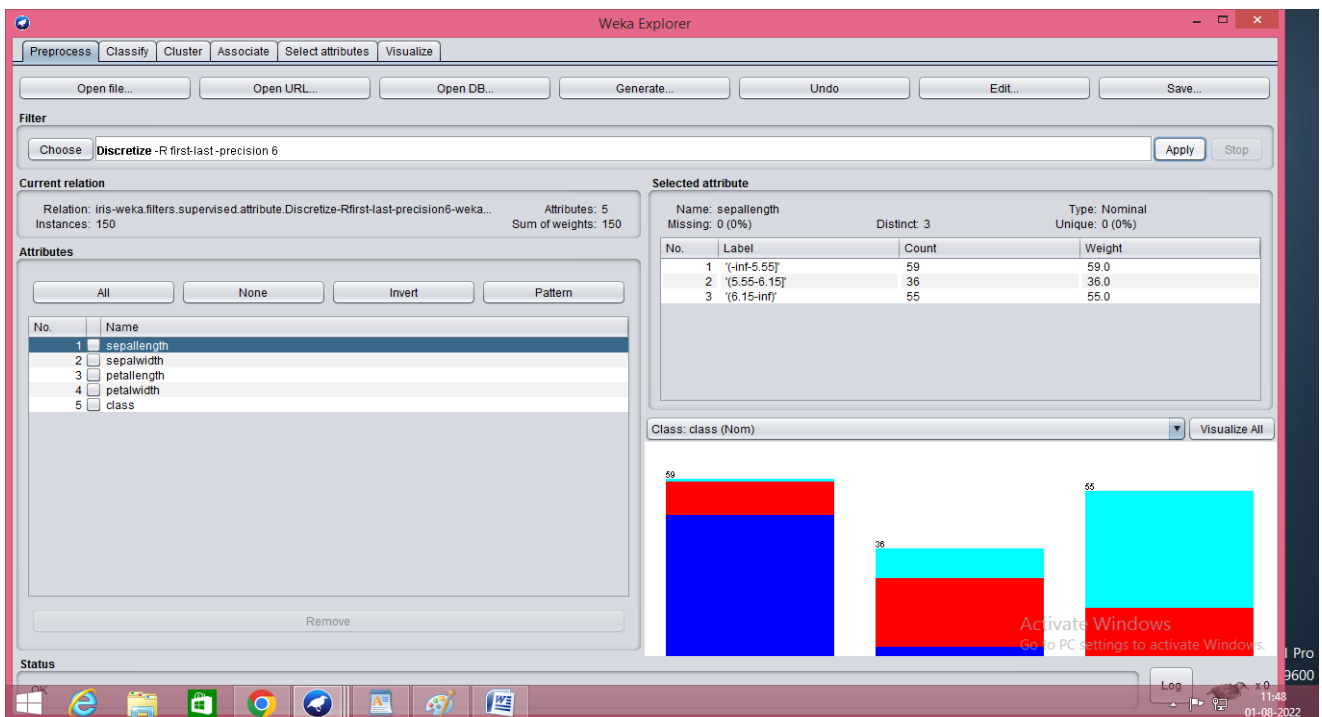
Now we will perform preprocessing on Iris Dataset.

11. Now to perform Discretize preprocessing filter, go to choose option now select the filter option under filter we have different modes of filters to perform preprocessing.

Select the attribute under supervised there we have a Discretize option choose that.



12. At the right side click on that Apply button to apply the current filter to the data.



13. We performed preprocessing on Iris dataset, now again classify the dataset by using bayesian classification models to get accurate results.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose NaiveBayes

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) class

Start Stop

Result list (right-click for options)

11:41:28 - bayes.NaiveBayes

11:43:38 - bayes.NaiveBayes

11:48:44 - bayes.NaiveBayes

Classifier output

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes

Relation: iris-weka.filters.supervised.attribute.Discretize-Rfirst-last-precision6-weka.filters.supervised.attribute.Discretize-Rfirst-last

Instances: 150

Attributes: 5

sepalength

sepalwidth

petallength

petalwidth

class

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

Attribute	Class	Iris-setosa (0.33)	Iris-versicolor (0.33)	Iris-virginica (0.33)
sepalength				
'(-inf-5.55]'		48.0	12.0	2.0
'(5.55-6.15]'		4.0	24.0	11.0
'(6.15-inf)'		1.0	17.0	40.0
[total]		53.0	53.0	53.0
sepalwidth				
'(-inf-2.95]'		3.0	35.0	22.0

Status

Log

11:49 01-08-2022

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose NaiveBayes

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) class

Start Stop

Result list (right-click for options)

11:41:28 - bayes.NaiveBayes

11:43:38 - bayes.NaiveBayes

11:48:44 - bayes.NaiveBayes

Classifier output

Naive Bayes Classifier

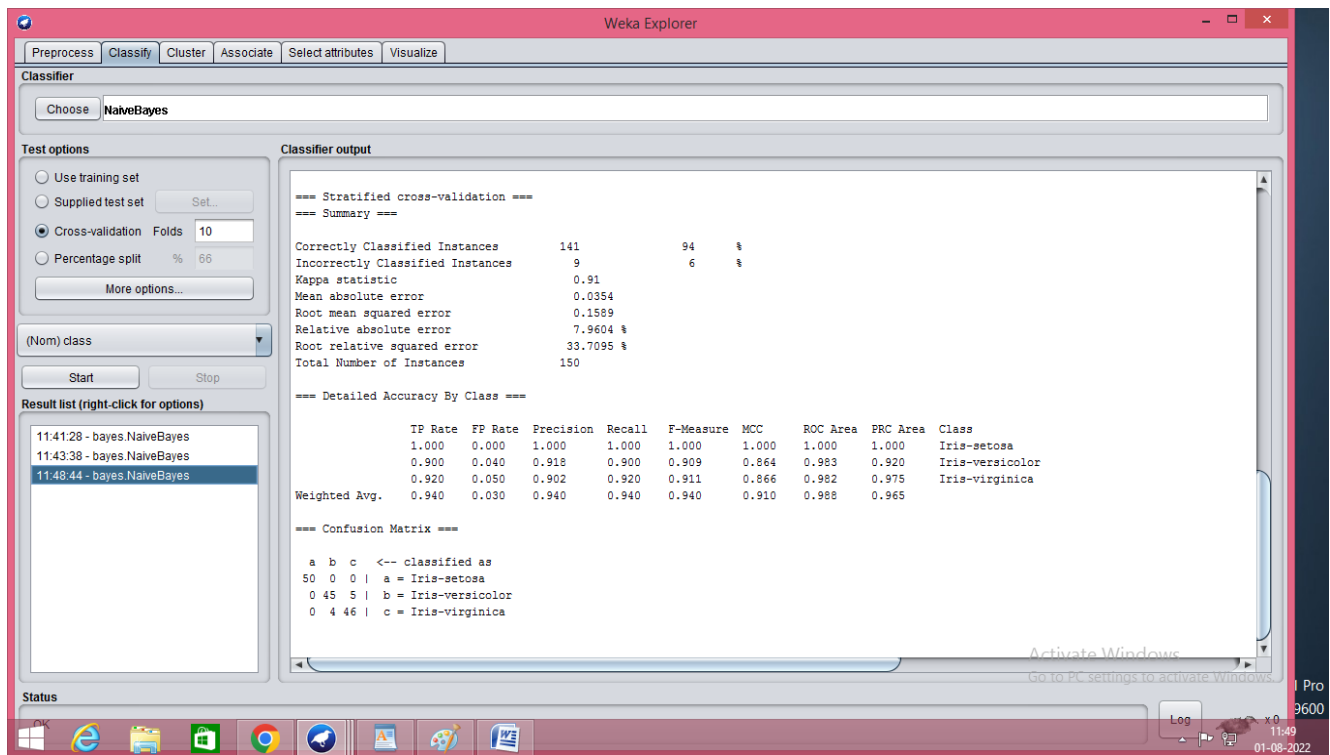
Attribute	Class	Iris-setosa (0.33)	Iris-versicolor (0.33)	Iris-virginica (0.33)
sepalength				
'(-inf-5.55]'		48.0	12.0	2.0
'(5.55-6.15]'		4.0	24.0	11.0
'(6.15-inf)'		1.0	17.0	40.0
[total]		53.0	53.0	53.0
sepalwidth				
'(-inf-2.95]'		3.0	35.0	22.0
'(2.95-3.35]'		19.0	16.0	25.0
'(3.35-inf)'		31.0	2.0	6.0
[total]		53.0	53.0	53.0
petallength				
'(-inf-2.45]'		51.0	1.0	1.0
'(2.45-4.75]'		1.0	45.0	2.0
'(4.75-inf)'		1.0	7.0	50.0
[total]		53.0	53.0	53.0
petalwidth				
'(-inf-0.8]'		51.0	1.0	1.0
'(0.8-1.75]'		1.0	50.0	6.0
'(1.75-inf)'		1.0	2.0	46.0
[total]		53.0	53.0	53.0

Status

Log

11:49 01-08-2022





## 5.2.4 Results and Discussion:

Implementing Bayesian Classification and analyze the results on 'iris' Dataset is successfully completed by drawing the confusion matrix and report the model with accuracy.

We observed that when we classify the dataset by using Bayesian classifier called NaiveBayes without preprocessing, then that we get best and high accuracy of classification only when we preprocessed the Dataset.