

**Ex no: 1**

## **EXPLORING DATASET USING WEKA**

**Date:**

### **AIM:**

To explore Iris dataset at initial step to uncover patterns, characteristics and point of interests using Weka Tool.

### **ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Apply filters if necessary.

**Step 6:** Select the attributes to explore the dataset.

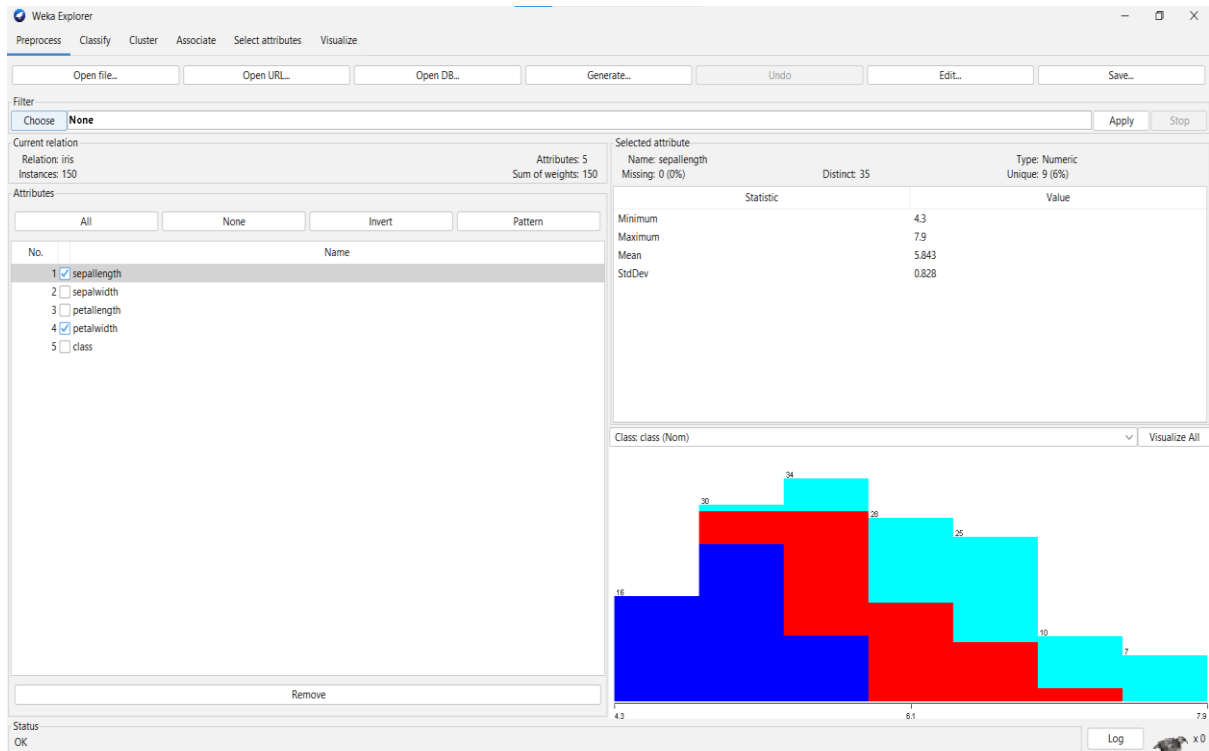
**Step 7:** Analyze over the statistical measures such as Maximum, Minimum, Mean and Standard Deviation.

**Step 8:** Visualize the dataset to uncover patterns and characteristics.

**Step 9:** Display the output.

**Step 10:** Stop the process.

## OUTPUT:



## RESULT:

Hence the above program has been executed successfully.

**Ex no: 2**

## **DATA PREPROCESSING USING WEKA**

**Date:**

### **AIM:**

To implement data preprocessing for all operations to build the final dataset using Weka Tool.

### **ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Filter the dataset using “NumericTransform”.

**Step 6:** Select the filter and apply the changes.

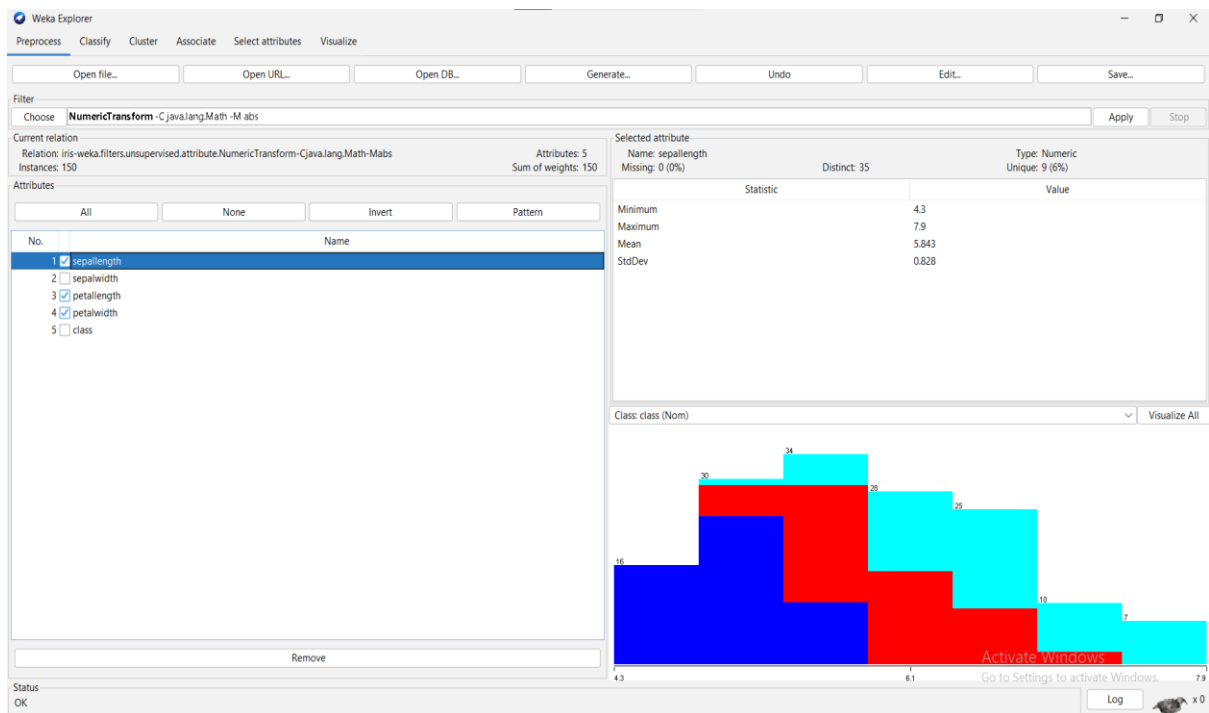
**Step 7:** Select the attributes to visualize the outcome.

**Step 8:** Analyze over the statistical measures such as Maximum, Minimum, Mean and Standard Deviation.

**Step 9:** Display the output.

**Step 10:** Stop the process.

## OUTPUT:



## RESULT:

Hence the above program has been executed successfully.

**Ex no: 3**

## **DATA VISUALIZATION USING WEKA**

**Date:**

### **AIM:**

To apply visualization techniques by encoding it as visual objects (points, lines) contained in graphics using Weka Tool.

### **ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Apply filters if necessary.

**Step 6:** Click on Visualize tab to visualize the data as visual objects.

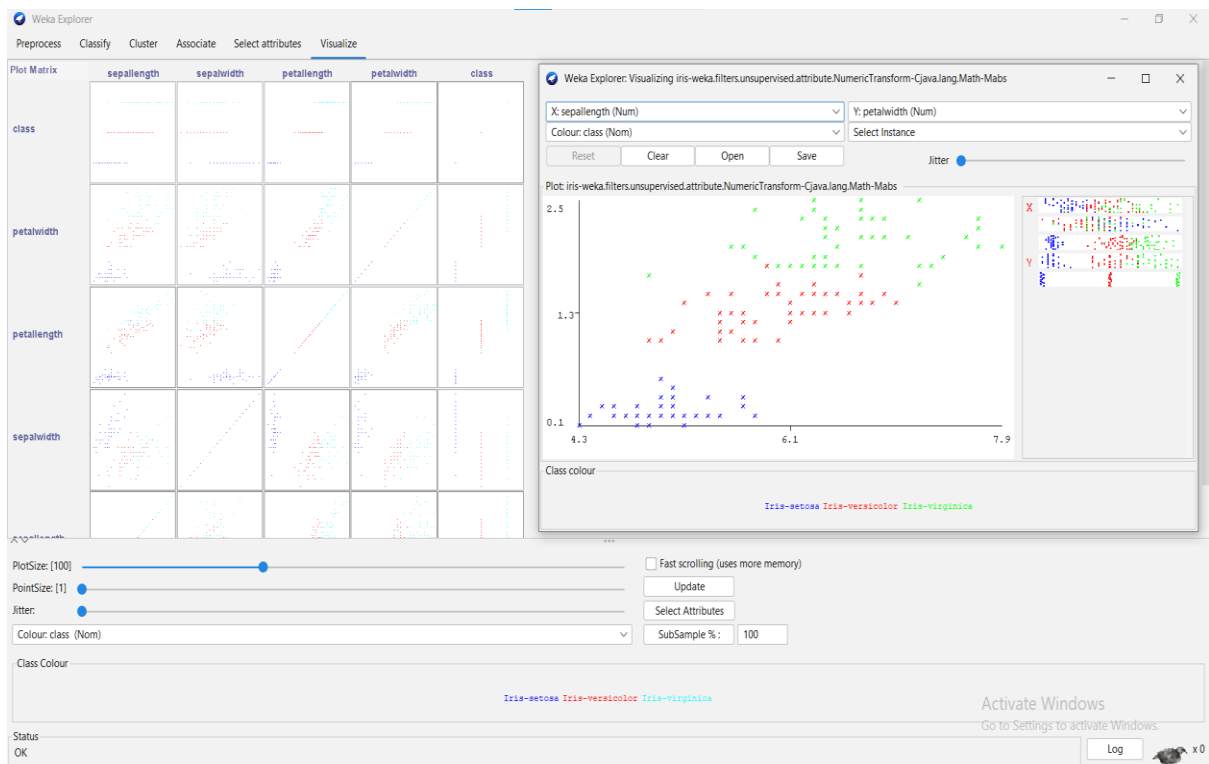
**Step 7:** Click on the visual graphics and the apply the visual encodings such as PlotSize and PointSize.

**Step 8:** Select and update the attributes as per the requirements.

**Step 9:** Display the output.

**Step 10:** Stop the process.

## OUTPUT:



## RESULT:

Hence the above program has been executed successfully.

**Ex no: 4**

## **CROSS VALIDATION USING WEKA**

**Date:**

### **AIM:**

To perform cross validation techniques with trained model complementary subset using Weka Tool.

### **ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Apply filters if necessary.

**Step 6:** Choose the J48 classifier by selecting Classify ---> Choose ---> Trees ---> J48.

**Step 7:** Click on Cross Validation and initialize the folds.

**Step 8:** Click Start the note the Classifier output.

**Step 9:** Display the output.

**Step 10:** Stop the process.

## OUTPUT:

**Weka Explorer**  
Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose J48 -C 0.25 -M 2

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):  
110722 - trees.J48  
110827 - trees.J48

Classifier output:

```
| | | pres > 61
| | | age <= 30: tested_negative (40.0/13.0)
| | | age > 30: tested_positive (60.0/17.0)
| | | plas > 157: tested_positive (52.0/12.0)
```

Number of Leaves : 20  
Size of the tree : 39  
Time taken to build model: 0.03 seconds

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

Correctly Classified Instances	567	73.8281 %
Incorrectly Classified Instances	201	26.1719 %
Kappa statistic	0.4164	
Mean absolute error	0.3158	
Root mean squared error	0.4463	
Relative absolute error	65.4841 %	
Root relative squared error	93.6293 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.814	0.403	0.790	0.814	0.802	0.417	0.751	0.811	tested_negative
	0.597	0.186	0.632	0.597	0.614	0.417	0.751	0.572	tested_positive
Weighted Avg.	0.738	0.327	0.735	0.738	0.736	0.417	0.751	0.727	

=== Confusion Matrix ===

a	b	<-- classified as	
407	93	a = tested_negative	
108	160	b = tested_positive	

Status: OK

Activate Windows  
Go to Settings to activate Windows.  
Log

## Before Validation

**Weka Explorer**  
Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose J48 -C 0.25 -M 2

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):  
110722 - trees.J48

Classifier output:

```
| | | plas > 157: tested_positive (52.0/12.0)
```

Number of Leaves : 20  
Size of the tree : 39  
Time taken to build model: 0.05 seconds

=== Evaluation on training set ===  
Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances	646	84.1146 %
Incorrectly Classified Instances	122	15.8854 %
Kappa statistic	0.6319	
Mean absolute error	0.2383	
Root mean squared error	0.3452	
Relative absolute error	52.4339 %	
Root relative squared error	72.4207 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.936	0.336	0.839	0.936	0.885	0.642	0.888	0.915	tested_negative
	0.664	0.064	0.948	0.664	0.745	0.642	0.888	0.808	tested_positive
Weighted Avg.	0.841	0.241	0.842	0.841	0.836	0.642	0.888	0.878	

=== Confusion Matrix ===

a	b	<-- classified as	
468	32	a = tested_negative	
90	178	b = tested_positive	

Status: OK

Activate Windows  
Go to Settings to activate Windows.  
Log

## After Validation

## RESULT:

Hence the above program has been executed successfully.



**Ex no: 5**

## **BASELINE ACCURACY FOR CLASSIFICATION**

**Date:**

### **AIM:**

To implement baseline accuracy for classification problems with baseline score using Weka Tool.

### **ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Apply filters if necessary.

**Step 6:** Choose J48, Naive Bayes, IBk and PART classifiers under Classify tab.

**Step 7:** Note down the classifier output for each classifier.

**Step 8:** Analyze the change of baseline accuracy for the various classification models.

**Step 9:** Display the output.

**Step 10:** Stop the process.

## OUTPUT:

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'J48 - C 0.25 - M 2'. The 'Test options' section shows 'Use training set' selected. The 'Result list' on the left shows '06:01:10 - trees.J48' selected. The 'Classifier output' pane displays the following information:

Classifier output  
| | plas > 157: tested\_positive (92.0/12.0)

Number of Leaves : 20  
Size of the tree : 39

Time taken to build model: 0.08 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.03 seconds

=== Summary ===

Correctly Classified Instances	646	84.1146 %
Incorrectly Classified Instances	122	15.8854 %
Kappa statistic	0.6319	
Mean absolute error	0.2383	
Root mean squared error	0.3452	
Relative absolute error	50.4359 %	
Root relative squared error	72.4207 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.936	0.336	0.839	0.936	0.885	0.642	0.888	0.515	tested_negative
	0.664	0.064	0.848	0.664	0.745	0.642	0.888	0.808	tested_positive

=== Confusion Matrix ===

	a	b	<-- classified as
468 32	a = tested_negative		
90 178	b = tested_positive		

Status: OK

J48

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'NaiveBayes'. The 'Test options' section shows 'Use training set' selected. The 'Result list' on the left shows '06:02:35 - bayes.NaiveBayes' selected. The 'Classifier output' pane displays the following information:

Classifier output

age  
mean 31.2494 37.0808  
std. dev. 11.4059 10.9146  
weight sum 500 268  
precision 1.1765 1.1765

Time taken to build model: 0.02 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.03 seconds

=== Summary ===

Correctly Classified Instances	586	76.3021 %
Incorrectly Classified Instances	182	23.6979 %
Kappa statistic	0.4674	
Mean absolute error	0.2811	
Root mean squared error	0.4133	
Relative absolute error	61.8486 %	
Root relative squared error	86.7082 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

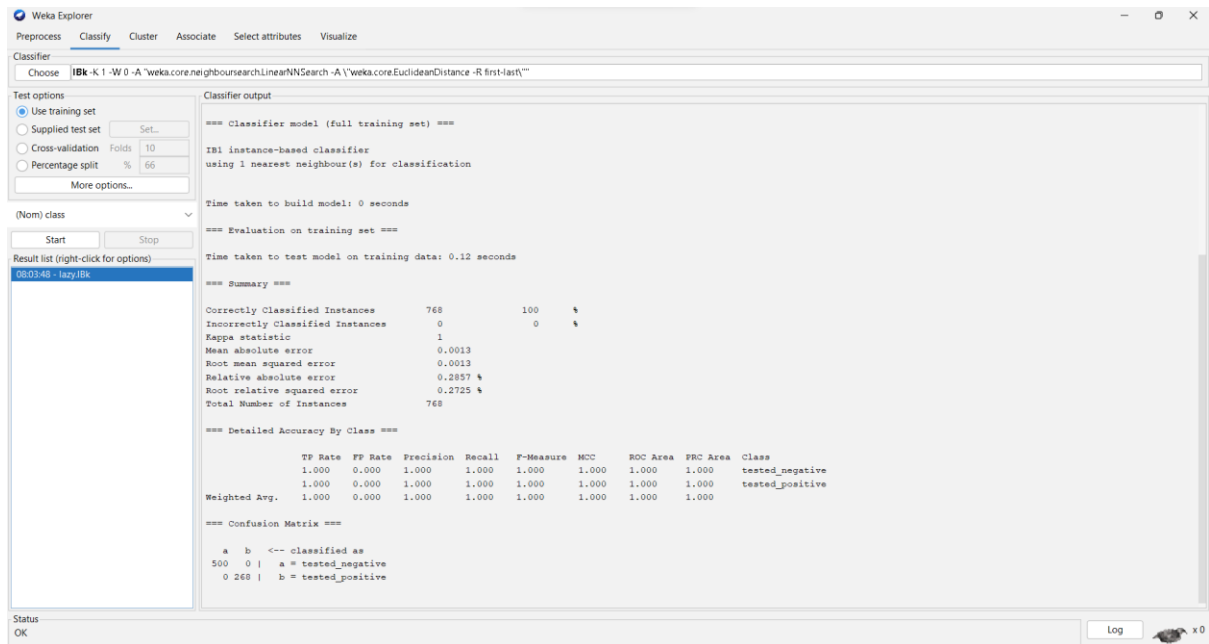
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.842	0.384	0.803	0.842	0.822	0.469	0.825	0.502	tested_negative
	0.616	0.158	0.676	0.616	0.645	0.469	0.825	0.684	tested_positive

=== Confusion Matrix ===

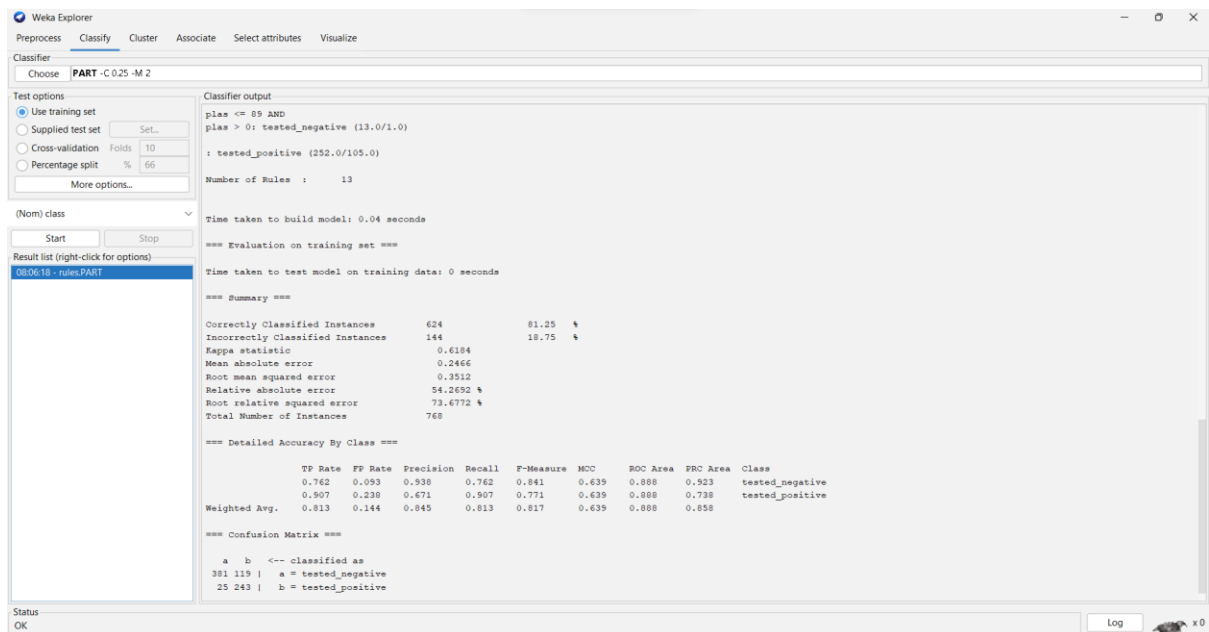
	a	b	<-- classified as
421 79	a = tested_negative		
103 165	b = tested_positive		

Status: OK

NaiveBayes



**IBk**



**PART**

**RESULT:**

Hence the above program has been executed successfully.

**Ex no: 6**

## **NEAREST NEIGHBOUR CLASSIFICATION**

**Date:**

### **AIM:**

To implement nearest neighbour algorithm to classify the given data based on similarity score using Weka Tool.

### **ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Apply filters if necessary.

**Step 6:** Choose IBk classifier under Classify tab and train the model.

**Step 7:** Open the properties by right clicking the classifier.

**Step 8:** Change the value of KNN and notice the changes.

**Step 9:** Display the output.

**Step 10:** Stop the process.

## OUTPUT:

**Classifier**  
Choose `libk-k 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A "weka.core.EuclideanDistance -R first-last"`

**Test options**  
☒ Use training set  
☐ Supplied test set  
☐ Cross-validation Folds 1  
☐ Percentage split % 66

**Classifier output**  
==== Classifier model (full training set) ====  
libk instance-based classifier  
using 1 nearest neighbour(s) for classification  
Time taken to build model: 0 seconds  
==== Evaluation on training set ====  
Time taken to test model on training data: 0 seconds  
==== Summary ====  
Correctly Classified Instances 14 100 %  
Incorrectly Classified Instances 0 0 %  
Kappa statistic 1  
Mean absolute error 0.0425  
Root mean squared error 0.0625  
Relative absolute error 13.4615 %  
Root relative squared error 13.0347 %  
Total Number of Instances 14  
==== Detailed Accuracy By Class ====  

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	yes
0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	no
Weighted Avg.	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	

  
==== Confusion Matrix ====  
a b <-- classified as  
9 0 | a = yes  
0 5 | b = no

**weka.gui.GenericObjectEditor**  
weka.classifiers.lazy.libk  
About  
K-nearest neighbours classifier.  
KNN 1  
batchSize 100  
crossValidate False  
debug False  
distanceWeighting No distance weighting  
doNotCheckCapabilities False  
meanSquared False  
nearestNeighbourSearchAlgorithm Choose LinearNNSearch -A "weka.core  
numDecimalPlaces 2  
windowSize 0  
Open... Save... OK Cancel

When KNN = 1

**Classifier**  
Choose `libk-k 15 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A "weka.core.EuclideanDistance -R first-last"`

**Test options**  
☒ Use training set  
☐ Supplied test set  
☐ Cross-validation Folds 1  
☐ Percentage split % 66

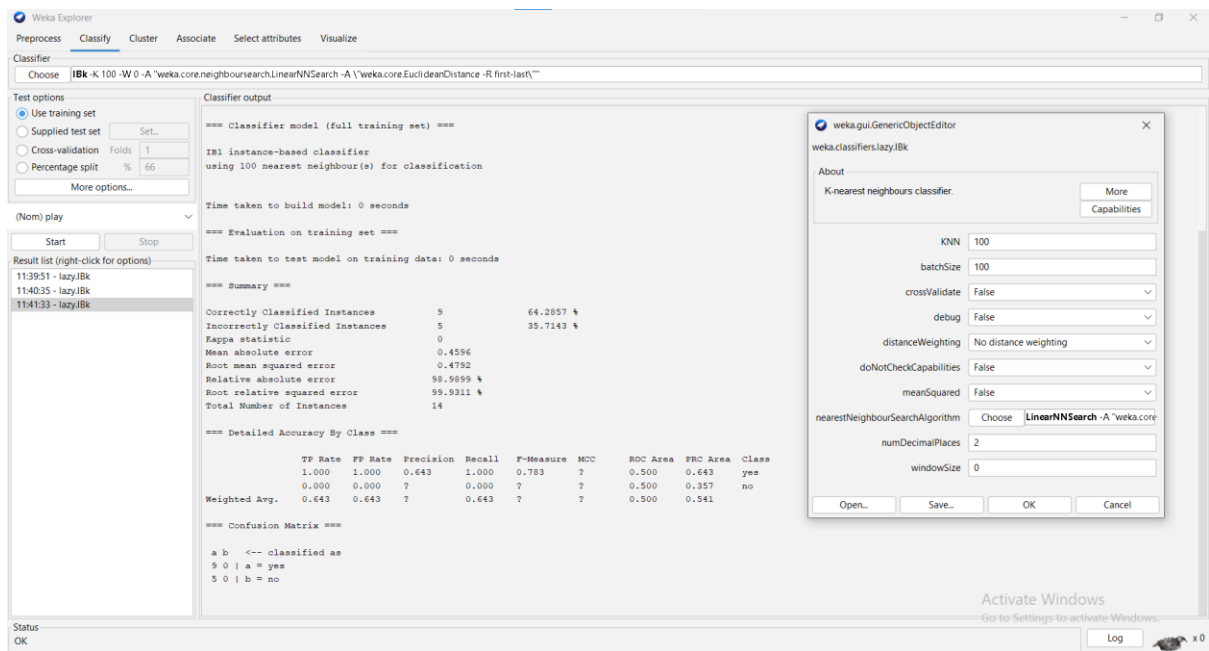
**Classifier output**  
==== Classifier model (full training set) ====  
libk instance-based classifier  
using 15 nearest neighbour(s) for classification  
Time taken to build model: 0 seconds  
==== Evaluation on training set ====  
Time taken to test model on training data: 0 seconds  
==== Summary ====  
Correctly Classified Instances 9 64.2857 %  
Incorrectly Classified Instances 5 35.7143 %  
Kappa statistic 0  
Mean absolute error 0.4596  
Root mean squared error 0.4792  
Relative absolute error 98.9899 %  
Root relative squared error 99.9311 %  
Total Number of Instances 14  
==== Detailed Accuracy By Class ====  

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	1.000	0.643	1.000	0.783	?	?	0.500	0.643	yes
0.000	0.000	?	0.000	?	?	?	0.500	0.357	no
Weighted Avg.	0.643	0.643	?	0.643	?	?	0.500	0.541	

  
==== Confusion Matrix ====  
a b <-- classified as  
9 0 | a = yes  
5 0 | b = no

**weka.gui.GenericObjectEditor**  
weka.classifiers.lazy.libk  
About  
K-nearest neighbours classifier.  
KNN 15  
batchSize 100  
crossValidate False  
debug False  
distanceWeighting No distance weighting  
doNotCheckCapabilities False  
meanSquared False  
nearestNeighbourSearchAlgorithm Choose LinearNNSearch -A "weka.core  
numDecimalPlaces 2  
windowSize 0  
Open... Save... OK Cancel

When KNN = 15



**When KNN = 100**

**RESULT:**

Hence the above program has been executed successfully.\

**Ex no: 7**

## **ASSOCIATION RULE MINING**

**Date:**

**AIM:**

To perform association rule mining to discover the dataset correlation using Weka Tool.

**ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Apply filters if necessary.

**Step 6:** Click on Associate tab and choose Apriori associator and train the model.

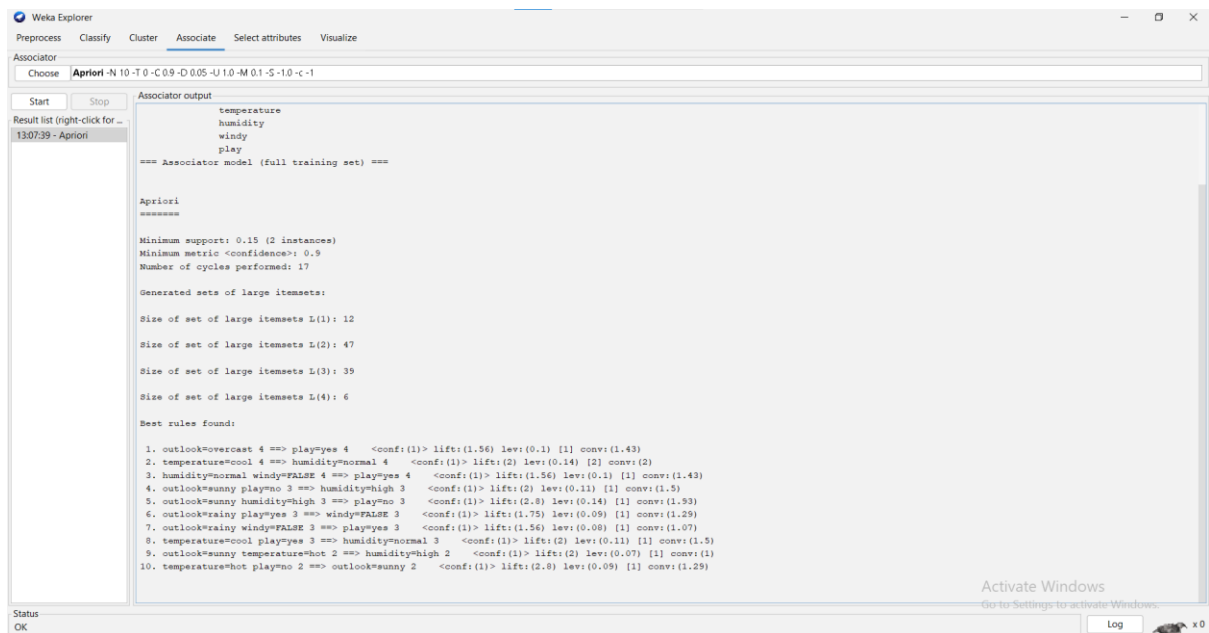
**Step 7:** Notice the changes in the associator output.

**Step 8:** Click on Visualize tab and see the visual depiction of results.

**Step 9:** Display the output.

**Step 10:** Stop the process.

## OUTPUT:



The screenshot shows the Weka Explorer interface with the 'Associate' tab selected. The 'Apriori' algorithm is chosen, and the 'Start' button has been clicked. The 'Result list' on the left shows '130739 - Apriori'. The main window displays the 'Apriori output' for the full training set. The output includes the list of attributes (temperature, humidity, windy, play), the minimum support (0.15), minimum metric (confidence: 0.9), and the number of cycles performed (17). It also lists the generated sets of large itemsets (L1 to L4) and the 10 best rules found, each with its confidence, lift, and leverage.

```
temperature
humidity
windy
play

=== Apriori model (full training set) ===

Apriori
=====

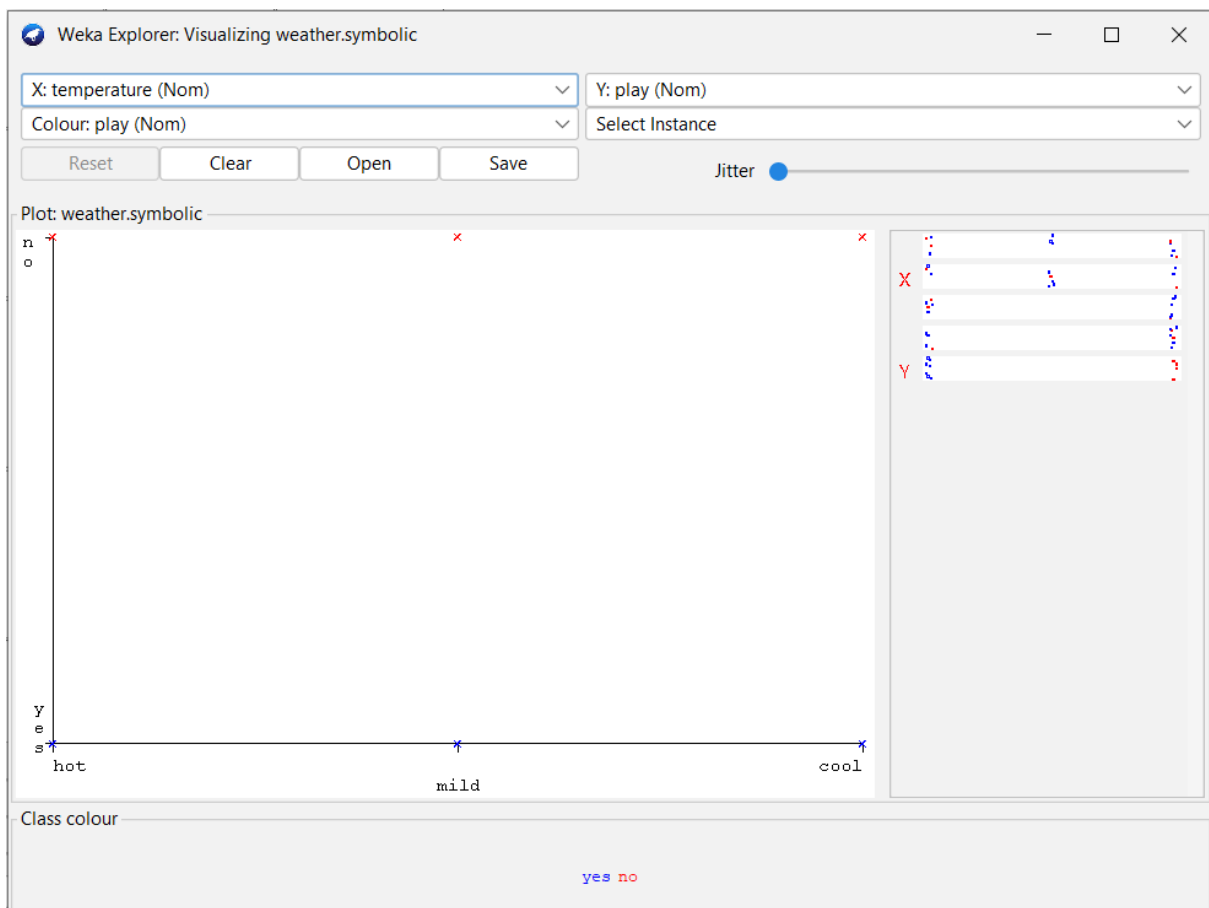
Minimum support: 0.15 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12
Size of set of large itemsets L(2): 47
Size of set of large itemsets L(3): 39
Size of set of large itemsets L(4): 6

Best rules found:

1. outlook=overcast 4 ==> play=yes 4 <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
2. temperature=cool 4 ==> humidity=normal 4 <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
3. humidity=normal windy=FALSE 4 ==> play=yes 4 <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
4. outlook=sunny play=no 3 ==> humidity=high 3 <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
5. outlook=sunny humidity=high 3 ==> play=no 3 <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.53)
6. outlook=rainy play=yes 3 ==> windy=FALSE 3 <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
7. outlook=rainy windy=FALSE 3 ==> play=yes 3 <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
8. temperature=cool play=yes 3 ==> humidity=normal 3 <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
9. outlook=sunny temperature=hot 2 ==> humidity=high 2 <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2 <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)
```



## RESULT:

Hence the above program has been executed successfully.



**Ex no: 8**

## **TEXT CLASSIFICATION**

**Date:**

### **AIM:**

To implement text classification to classify documents into pre-defined categories on sample documents using Weka Tool.

### **ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Filter the dataset using “StringToWordVector”.

**Step 6:** Click on Classify tab, select Supplied test set and choose the dataset.

**Step 7:** Click More options, choose Plain Text under output prediction and click Ok.

**Step 8:** Train the model and notice the classifier output.

**Step 9:** Display the output.

**Step 10:** Stop the process.

## OUTPUT:

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Filter: Choose **StringToWordVector** -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters '\n\r\t\"'0?'" Apply Stop

Current relation: Relation: Reuters-21578 Corn ModApte Train-weka.filters.unsupervised.attribute.NumericToBinary-weka.filters.unsu... Attributes: 2234 Instances: 1554 Sum of weights: 1554

Attributes:

No.	Name
1	class-att
2	&#3
3	&lt
4	-
5	--
6	0
7	0/92
8	00
9	000
10	1
11	10
12	100
13	11
14	12
15	13
16	14
17	15
18	150
19	16
20	17
21	18
22	19
23	1000

Selected attribute: Name: class-att Missing: 0 (0%) Distinct: 2 Type: Nominal Unique: 0 (0%)

No.	Label	Count	Weight
1	0	1509	1509
2	1	45	45

Class zone (Num) Visualize All

1509

Activate Windows

Status: Problem evaluating classifier Log x 0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **InputMappedClassifier** -I -trim -W weka.classifiers.rules.ZeroR

Test options:

- ☐ Use training set
- ☒ Supplied test set Set...
- ☐ Cross-validation Folds: 10
- ☐ Percentage split %: 66

More options...

(Num) zone: Start Stop

Result list (right-click for options): 130253 - misc.InputMappedClassifier

Classifier output:

```
=== Run information ===
Scheme: weka.classifiers.misc.InputMappedClassifier -I -trim -W weka.classifiers.rules.ZeroR
Relation: Reuters-21578 Corn ModApte Train-weka.filters.unsupervised.attribute.NumericToBinary-weka.filters.unsupervised.instance.RemoveFolds-80-N5-F1-weka.filters.unsuperv
Instances: 1554
Attributes: 2234
Attributes: [list of attributes omitted]
Test mode: user supplied test set: size unknown (reading incrementally)

=== Classifier model (full training set) ===
InputMappedClassifier:
ZeroR predicts class value: 0.002574002574002574
Attribute mappings:

Model attributes Incoming attributes
-----
(nominal) class-att --> 1 (nominal) class-att
(numeric) &#3 --> 2 (numeric) &#3
(numeric) &lt --> 3 (numeric) &lt
(numeric) - --> 4 (numeric) -
(numeric) -- --> 5 (numeric) --
(numeric) 0 --> 6 (numeric) 0
(numeric) 0/92 --> 7 (numeric) 0/92
(numeric) 00 --> 8 (numeric) 00
(numeric) 000 --> 9 (numeric) 000
(numeric) 1 --> 10 (numeric) 1
(numeric) 10 --> 11 (numeric) 10
(numeric) 100 --> 12 (numeric) 100
(numeric) 11 --> 13 (numeric) 11
(numeric) 12 --> 14 (numeric) 12
(numeric) 13 --> 15 (numeric) 13
(numeric) 14 --> 16 (numeric) 14
(numeric) 15 --> 17 (numeric) 15
(numeric) 150 --> 18 (numeric) 150
(numeric) 16 --> 19 (numeric) 16
(numeric) 1000 --> 20 (numeric) 1000
```

Activate Windows

Status: Problem evaluating classifier Log x 0

## RESULT:

Hence the above program has been executed successfully.

**Ex no: 9**

## **REPEATED TRAINING & TESTING DATASETS**

**Date:**

### **AIM:**

To perform repeated training and testing to evaluate the classifier by splitting the datasets using Weka Tool.

### **ALGORITHM:**

**Step 1:** Start the process.

**Step 2:** Open Weka 3.8.6.

**Step 3:** Open Weka Explorer.

**Step 4:** Import the dataset by selecting either Open File, Open URL or Open DB under Preprocess tab.

**Step 5:** Under Classify tab, select Supplied test set and choose the dataset.

**Step 6:** Choose J48 classifier and train the model.

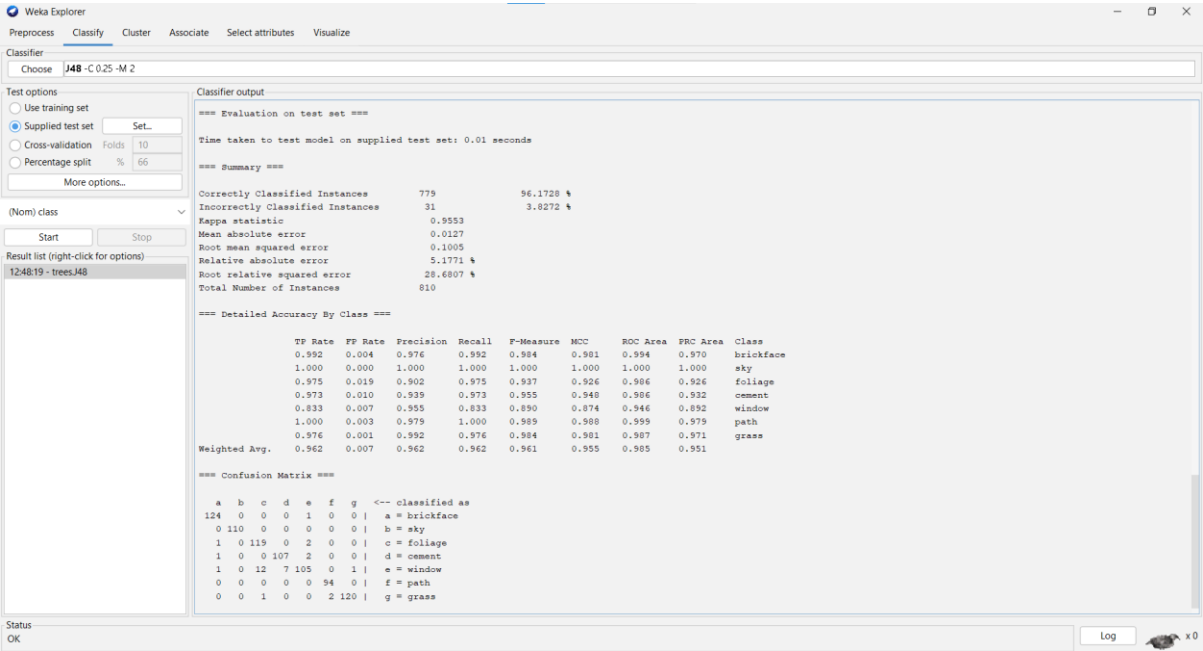
**Step 7:** Train the model using training set and percentage split (66%) as well.

**Step 8:** Compare both training and testing classifier outputs and note the changes.

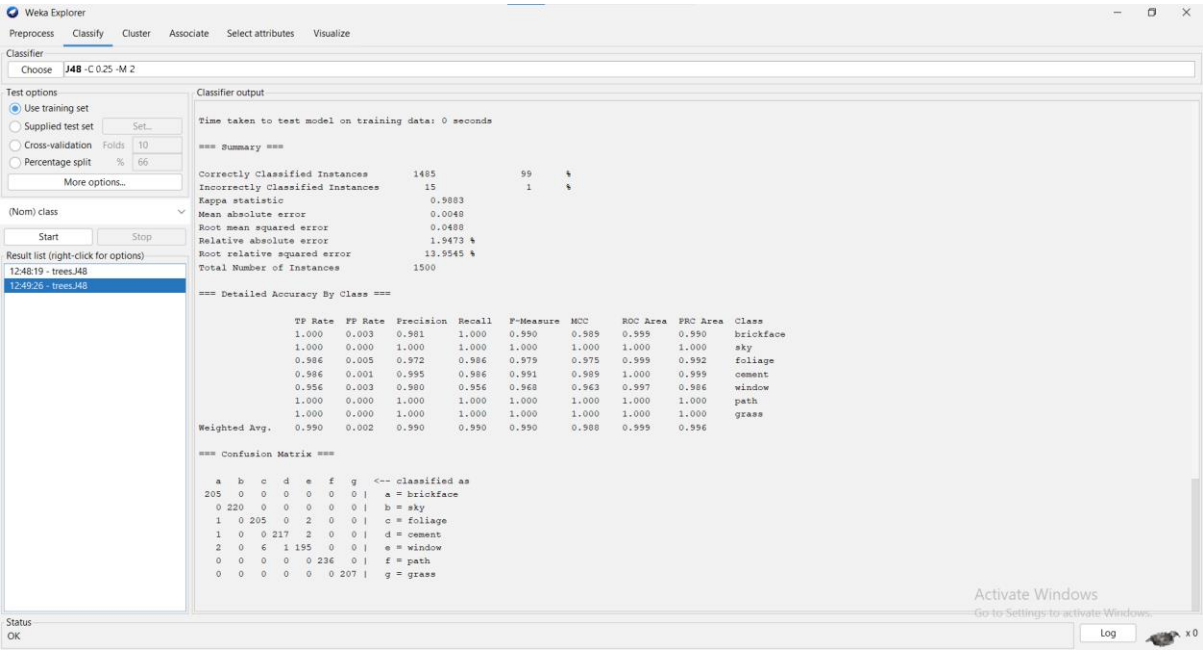
**Step 9:** Display the output.

**Step 10:** Stop the process.

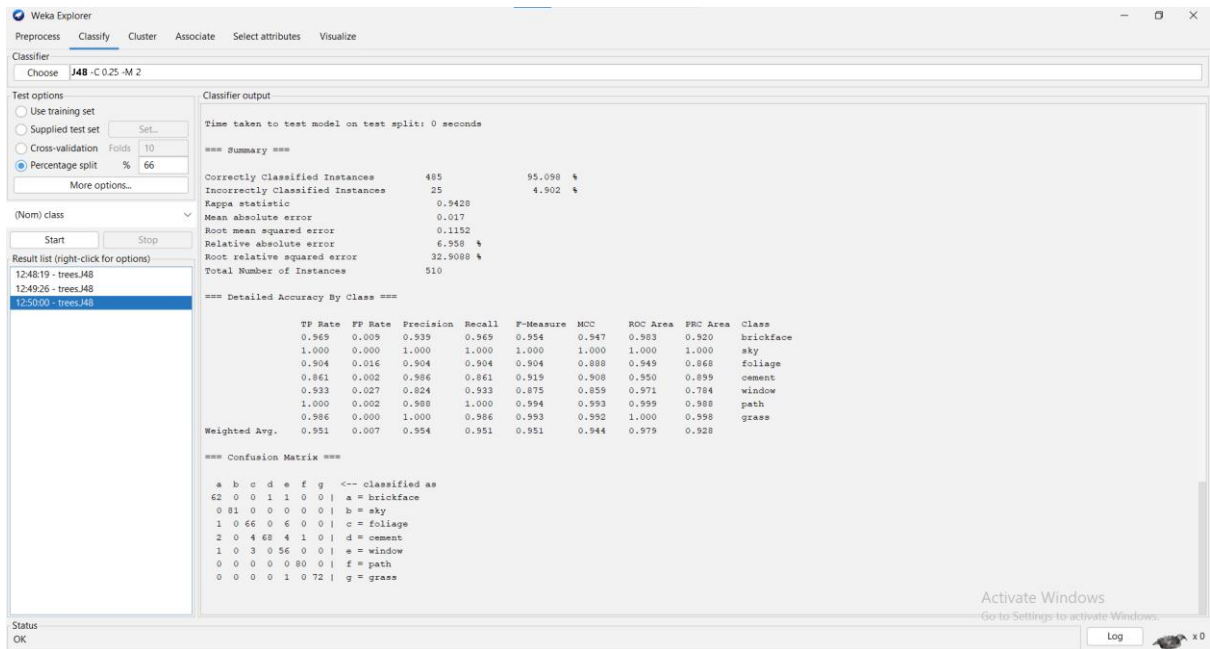
OUTPUT:



Supplied test set



Training set



## Percentage Split

### RESULT:

Hence the above program has been executed successfully.