

St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology

A.Y. 2024-2025
Class: TE-ITA/B, Semester: VI

Subject: **Business Intelligence Lab**

Experiment – 4: To implement a classifier- Decision tree using open source tool WEKA and ORANGE

1. **Aim:** To Implement any one of the classifiers using WEKA (Decision Tree, Naïve Bayes, Random Forest)
2. **Objectives:** After study of this experiment, the students will be able to Understand and knew about all the three classifiers.
3. **Outcomes:** After study of this experiment, the students will be able to
CO4: Design and Implement the appropriate data mining methods like classification, clustering or Frequent Pattern mining on large data sets.
CO5: Define and apply metrics to measure the performance of various data mining algorithms
4. **Prerequisite:** Introduction to all the three classifiers through algorithms & Problem solving approach.
5. **Requirements:** Personal Computer, Windows XP operating system/Windows 7, Internet Connection, Microsoft Word, WEKA tool, Java/R/Python.
6. **Theory:**
 - a. What is Classification Data Mining?
 - b. Difference between supervised and unsupervised learning
7. **Laboratory Exercise:** Implementation of Classification Algorithm in WEKA and Orange. Take printout of related snapshots.
8. **Post-Experiments Exercise**
 - a. **Extended Theory:**
 - i. Explain about Decision Tree algorithm
 - ii. Solve numerical on decision tree
9. **Exercise:**
 - Simple CLI execution of classification algorithm in WEKA
 - **For training:** `java weka.classifiers.trees.J48 -C 0.25 -M 2 -t directory-path\bank.arff -d directory-path \bank.model`
 - **For Testing:** `java weka.classifiers.trees.J48 -p 9 -l directory-path\bank.model -T directory-path \bank-new.arff`
10. **Conclusion:**
 - a. Summary of Experiment
 - b. Importance of Experiment
 - c. Application of Experiment
11. **Reference:** Data Mining: Concept & Techniques, 3rd Edition, Jiawei Han, Micheline Kamber, Jian Pei, Elsevier.

8. Post-Experiments Exercise

a. Extended Theory:

Q1. What is Classification Data Mining?

Classification is a supervised learning technique in data mining used to predict the categorical class or label of an item based on its features. The model is trained using labeled data, where both input features and their corresponding class labels are known. Once trained, the model can classify new, unseen data into predefined categories.

For instance, in medical data mining, classification can be applied to predict whether a patient has a particular disease based on symptoms and medical records. The model learns from past data where the disease status (e.g., "disease" or "no disease") is labeled and then predicts the class for new patient records.

Common Classification Algorithms:

- Decision Trees (e.g., J48, C4.5)
- Naïve Bayes
- Random Forest

Q2. Difference between supervised and unsupervised learning

Supervised learning	Unsupervised learning
The model is trained on labeled data, where the output (class label) is known.	The model is trained on unlabeled data, and no output labels are provided.
Predict the output for new, unseen data based on past labeled data.	Identify patterns, structures, or groups in the data.
Labeled data (input-output pairs)	Unlabeled data (only inputs, no output labels).
Accuracy, Precision, Recall, F1-Score, ROC-AUC, etc.	Cluster purity, Silhouette score, etc.
Eg: Spam detection, Disease prediction, Stock price forecasting	Eg: Customer segmentation, Market basket analysis, Anomaly detection.

LABORATORY EXERCISE

Training dataset

```
bank - Notepad
File Edit Format View Help
@relation bank@attribute age numeric@attribute sex {MALE,FEMALE}@attribute region {INNER_CITY,RURAL,TOWN,SUBURBAN}@attribute
40,MALE,TOWN,30085.1,YES,YES,YES,YES,NO
51,FEMALE,INNER_CITY,16575.4,YES,NO,YES,NO,NO
23,FEMALE,TOWN,20375.4,YES,YES,NO,NO,NO
57,FEMALE,RURAL,50576.3,YES,NO,NO,NO,NO
57,FEMALE,TOWN,37869.6,YES,YES,NO,NO,YESS
22,MALE,RURAL,8877.07,NO,NO,NO,NO,YES
58,MALE,TOWN,24946.6,YES,NO,YES,NO,NO
37,FEMALE,SUBURBAN,25304.3,YES,YES,YES,NO,NO
54,MALE,TOWN,24212.1,YES,YES,YES,NO,NO
66,FEMALE,TOWN,59803.9,YES,NO,NO,NO,NO
52,FEMALE,INNER_CITY,26658.8,NO,NO,YES,YES,NO
44,FEMALE,TOWN,15735.8,YES,NO,YES,YES
66,FEMALE,TOWN,55204.7,YES,YES,YES,YES
36,MALE,RURAL,19474.6,YES,NO,NO,YES,NO
38,FEMALE,INNER_CITY,22342.1,YES,NO,YES,YES,NO
37,FEMALE,TOWN,17729.8,YES,YES,NO,YES,NO
46,FEMALE,SUBURBAN,41016,YES,NO,NO,YES,NO
62,FEMALE,INNER_CITY,26909.2,YES,NO,NO,NO,YES
31,MALE,TOWN,22522.8,YES,NO,YES,NO,NO
61,MALE,INNER_CITY,57880.7,YES,YES,NO,NO,YES
50,MALE,TOWN,16497.3,YES,YES,NO,NO,NO
54,MALE,INNER_CITY,38446.6,YES,NO,NO,NO,NO
27,FEMALE,TOWN,15538.8,NO,NO,YES,YES,NO
```

Loading dataset in WEKA

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter: Choose **None** Apply Stop

Current relation: bank
Instances: 300
Attributes: 9
Sum of weights: 300

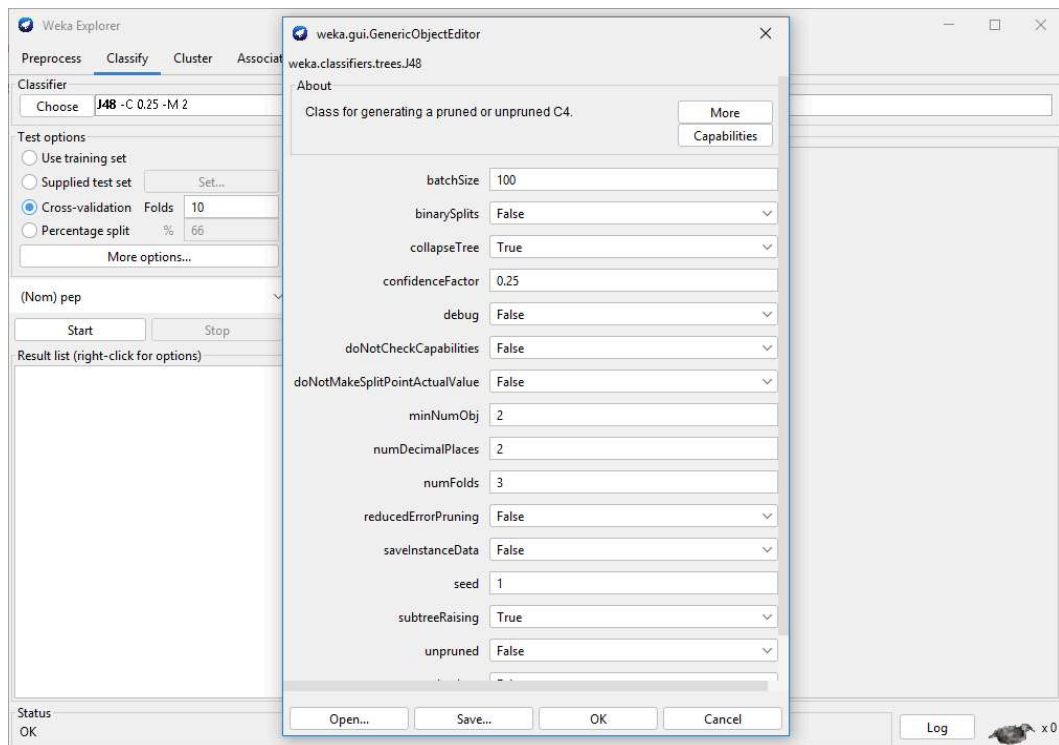
Selected attribute: Name: age
Missing: 0 (0%)
Distinct: 50
Type: Numeric
Unique: 0 (0%)

Statistic	Value
Minimum	18
Maximum	67
Mean	42.57
StdDev	14.22

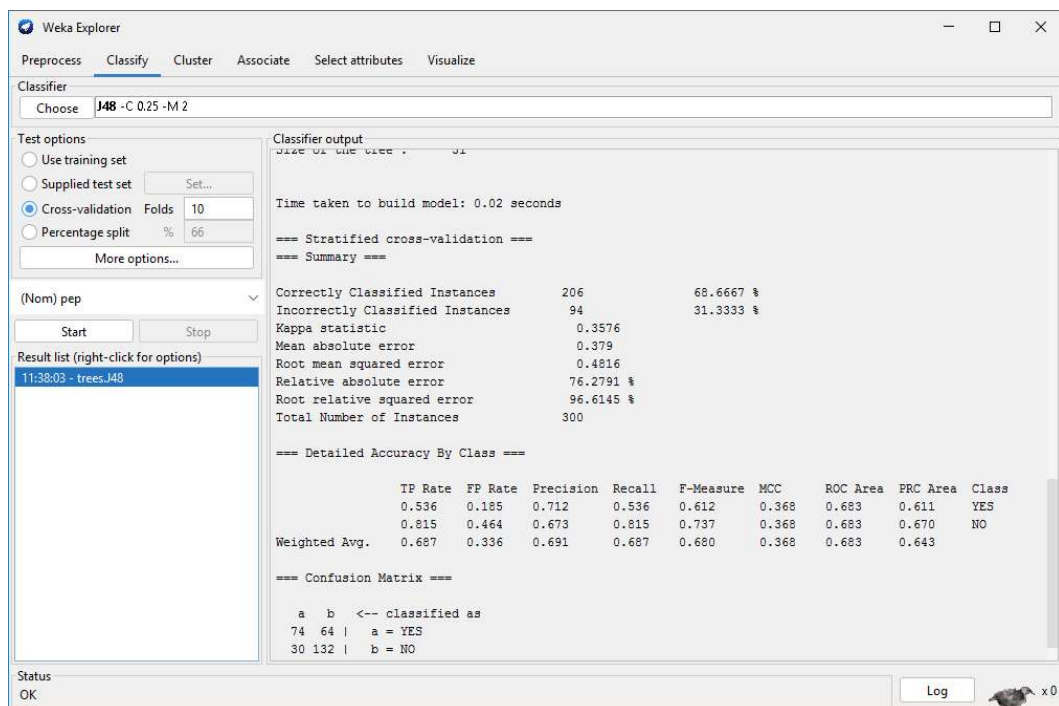
Class: pep (Nom) Visualize All

Status: OK Log x 0

Applying J48 on training dataset



Generation of model after applying Decision tree algorithm



Viewing output in second window

```
11:38:03 - trees.J48
=== Run information ===

Scheme:      weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:    bank
Instances:    300
Attributes:   9
              age
              sex
              region
              income
              married
              children
              car
              mortgage
              pep
Test mode:    10-fold cross-validation

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

J48 pruned tree
-----

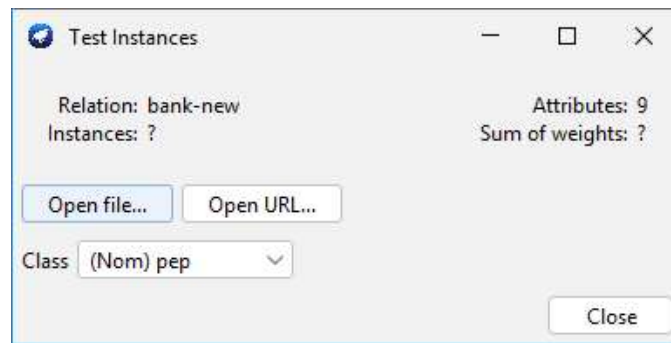
children = YES
|  income <= 30099.3
|  |  car = YES: NO (50.0/15.0)
|  |  car = NO
|  |  |  married = YES
|  |  |  |  income <= 13106.6: NO (9.0/2.0)
|  |  |  |  income > 13106.6
|  |  |  |  |  mortgage = YES: YES (12.0/3.0)
|  |  |  |  |  mortgage = NO
|  |  |  |  |  income <= 18923: YES (9.0/3.0)
|  |  |  |  |  income > 18923: NO (10.0/3.0)
```

Test dataset

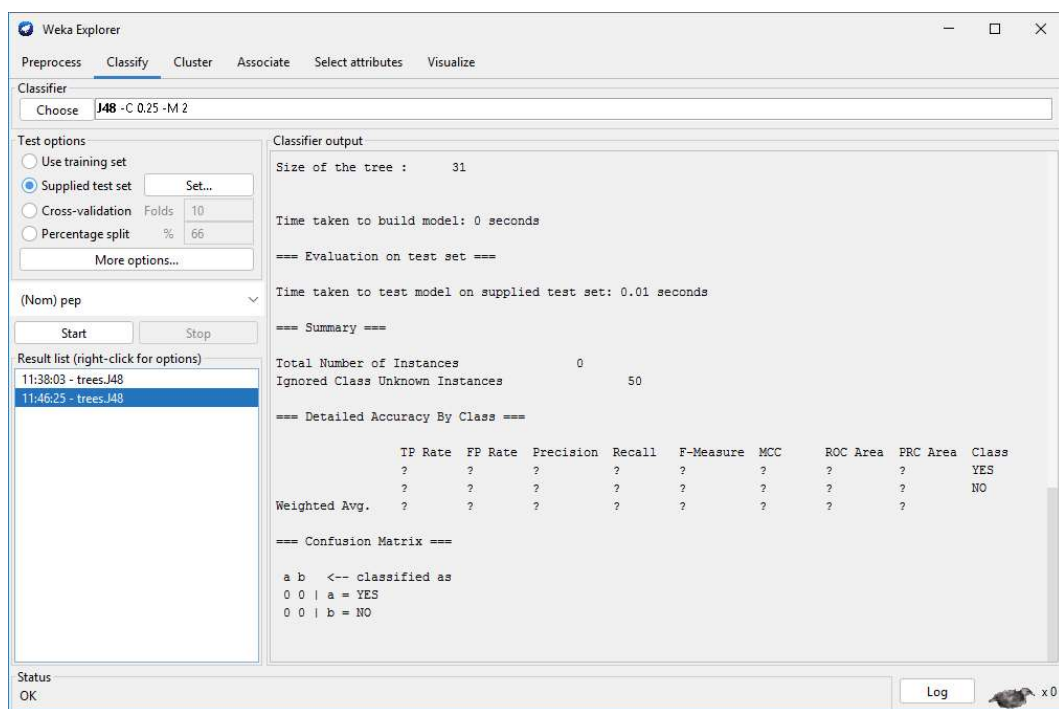
```
@relation bank-new
|
@attribute age numeric
@attribute sex {MALE,FEMALE}
@attribute region {INNER_CITY,RURAL,TOWN,SUBURBAN}
@attribute income numeric
@attribute married {YES,NO}
@attribute children {YES,NO}
@attribute car {YES,NO}
@attribute mortgage {YES,NO}
@attribute pep {YES,NO}

@data
23,MALE,INNER_CITY,18766.9,YES,NO,YES,YES,?
30,MALE,RURAL,9915.67,NO,YES,NO,YES,?
45,FEMALE,RURAL,21881.6,NO,NO,YES,NO,?
50,MALE,TOWN,46794.4,YES,YES,NO,YES,?
41,FEMALE,INNER_CITY,20721.1,YES,NO,YES,NO,?
```

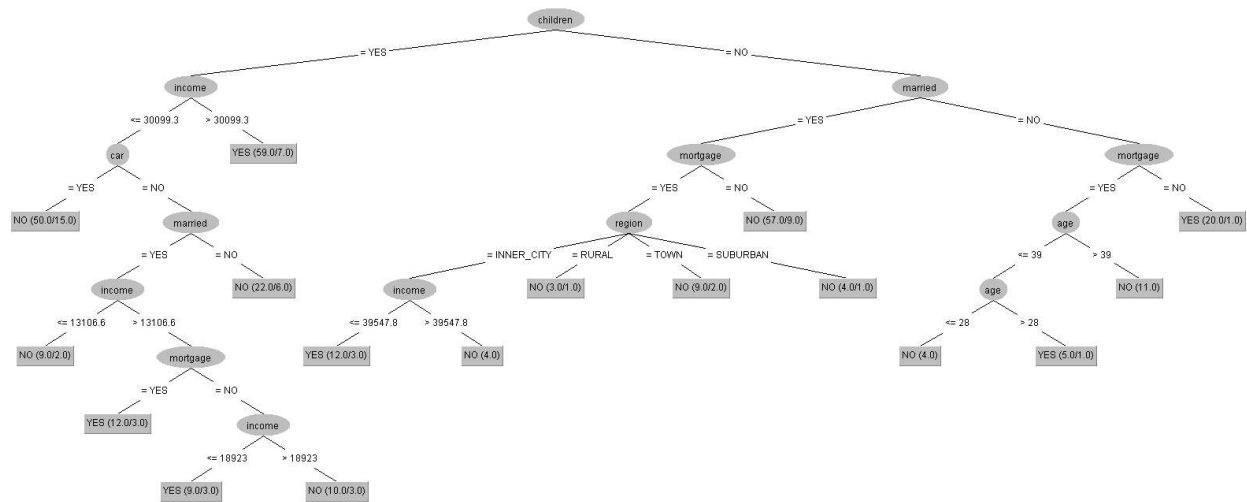
Loading test dataset in WEKA



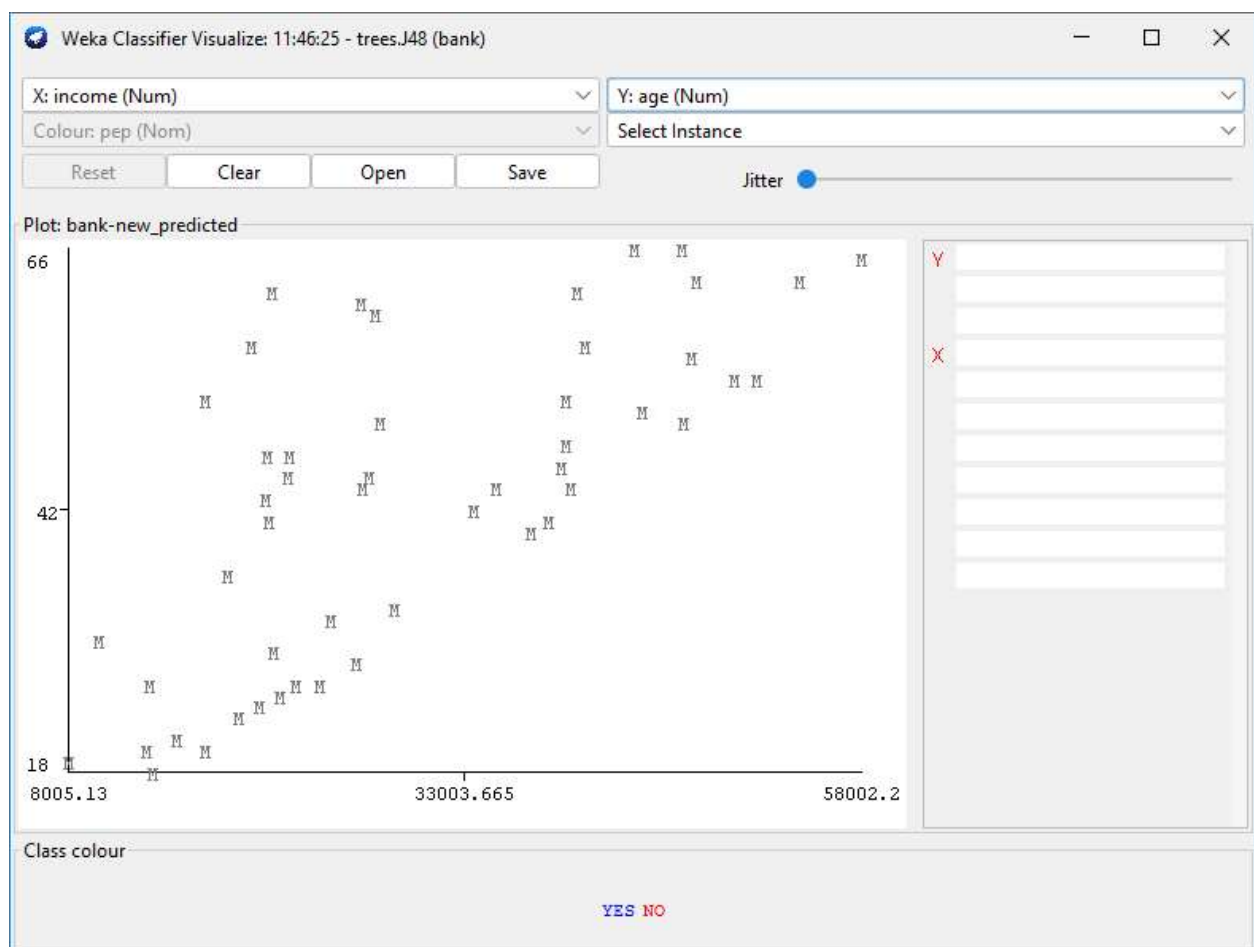
Output of the Test Dataset



Visualization of Tree



Visualization of classifiers



Predicated class values for the given test dataset

```
@relation bank-new_predicted

@attribute age numeric
@attribute sex {MALE,FEMALE}
@attribute region {INNER_CITY,RURAL,TOWN,SUBURBAN}
@attribute income numeric
@attribute married {YES,NO}
@attribute children {YES,NO}
@attribute car {YES,NO}
@attribute mortgage {YES,NO}
@attribute 'prediction margin' numeric
@attribute 'predicted pep' {YES,NO}
@attribute pep {YES,NO}

@data
23,MALE,INNER_CITY,18766.9,YES,NO,YES,YES,0.5,YES,?
30,MALE,RURAL,9915.67,NO,YES,NO,YES,-0.454545,NO,?
45,FEMALE,RURAL,21881.6,NO,NO,YES,NO,0.9,YES,?
50,MALE,TOWN,46794.4,YES,YES,NO,YES,0.762712,YES,?
```


Post experiment Exercise:

For training: `java weka.classifiers.trees.J48 -C 0.25 -M 2 -t directory-path\bank.arff -d directory-path \bank.model`

Training Output in WEKA Simple CLI

```
SimpleCLI
> java weka.classifiers.trees.J48 -C 0.25 -M 2 -t C:\ShubhamMalekar\Exp4\bank.arff -d
C:\ShubhamMalekar\Exp4\bank.model

Options: -C 0.25 -M 2

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

J48 pruned tree
-----

children = YES
|   income <= 30099.3
|   |   car = YES: NO (50.0/15.0)
|   |   car = NO
|   |   |   married = YES
|   |   |   |   income <= 13106.6: NO (9.0/2.0)
|   |   |   |   income > 13106.6
|   |   |   |   |   mortgage = YES: YES (12.0/3.0)
|   |   |   |   |   mortgage = NO
|   |   |   |   |   |   income <= 18923: YES (9.0/3.0)
|   |   |   |   |   |   income > 18923: NO (10.0/3.0)
|   |   |   |   |   married = NO: NO (22.0/6.0)
|   |   |   |   income > 30099.3: YES (59.0/7.0)
children = NO
|   married = YES
|   |   mortgage = YES
|   |   |   region = INNER_CITY
|   |   |   |   income <= 39547.8: YES (12.0/3.0)
|   |   |   |   income > 39547.8: NO (4.0)
|   |   |   |   region = RURAL: NO (3.0/1.0)
|   |   |   |   region = TOWN: NO (9.0/2.0)
|   |   |   |   region = SUBURBAN: NO (4.0/1.0)
|   |   |   mortgage = NO: NO (57.0/9.0)
|   |   married = NO
|   |   |   mortgage = YES
|   |   |   |   age <= 39
|   |   |   |   |   age <= 28: NO (4.0)
|   |   |   |   |   age > 28: YES (5.0/1.0)
|   |   |   |   |   age > 39: NO (11.0)
|   |   |   |   mortgage = NO: YES (20.0/1.0)

Number of Leaves : 17
```

Applying test dataset on the model file to get the following results

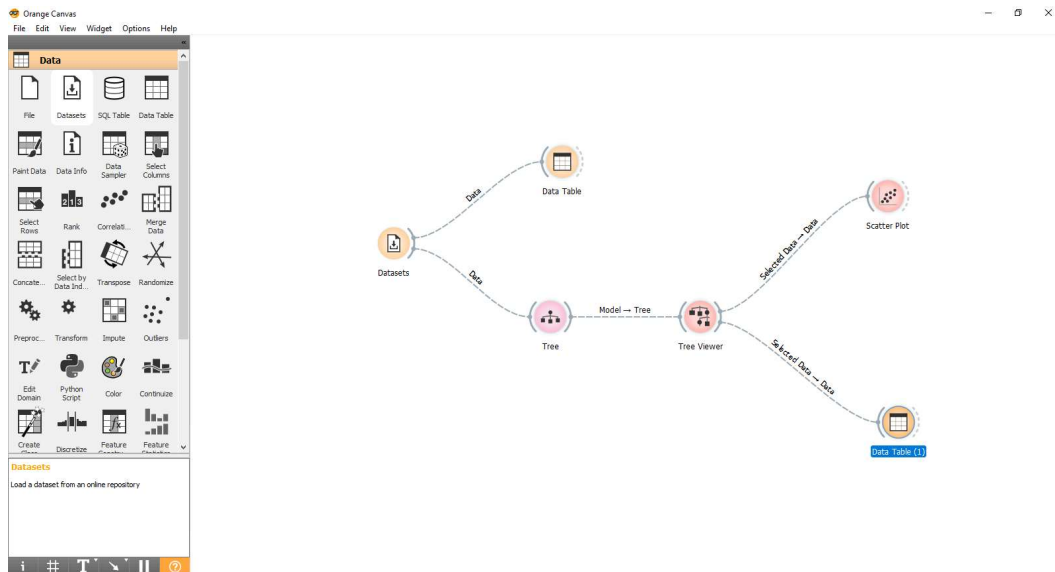
```
SimpleCLI
> java weka.classifiers.trees.J48 -p 9 -l C:\ShubhamMalekar\Exp4\bank.model
-T C:\ShubhamMalekar\Exp4\bank-new.arff

=== Predictions on test data ===

inst#    actual    predicted error prediction ()
1        1:??      1:YES      0.75
2        1:??      2:NO       0.727
3        1:??      1:YES      0.95
4        1:??      1:YES      0.881
5        1:??      2:NO       0.842
6        1:??      2:NO       0.727
7        1:??      2:NO       0.667
8        1:??      2:NO       0.7
9        1:??      1:YES      0.881
10       1:??      2:NO       0.7
11       1:??      2:NO       1
12       1:??      2:NO       1
13       1:??      2:NO       0.842
14       1:??      2:NO       0.842
15       1:??      2:NO       0.778
16       1:??      1:YES      0.881
17       1:??      1:YES      0.75
18       1:??      1:YES      0.881
19       1:??      1:YES      0.667
20       1:??      2:NO       0.7
21       1:??      1:YES      0.881
22       1:??      2:NO       0.7
23       1:??      1:YES      0.881
24       1:??      2:NO       0.667
25       1:??      2:NO       0.7
26       1:??      2:NO       0.727
27       1:??      2:NO       0.842
28       1:??      2:NO       0.667
29       1:??      2:NO       0.842
30       1:??      2:NO       0.842
31       1:??      1:YES      0.881
32       1:??      1:YES      0.881
33       1:??      2:NO       0.842
34       1:??      1:YES      0.881
35       1:??      2:NO       0.842
```

Orange

Applying tree window in orange



Applying Iris dataset

The screenshot shows the Orange Datasets window, which displays a list of available datasets. The **Iris** dataset is selected and highlighted.

Title	Size	Instances	Variables	Target	Tags
Iris	4.5 KB	150	5	5	category, biology
Adult Census Income	5.3 MB	48842	15	15	category, economy, fai...
COMPAS Analysis	2.7 MB	7214	52	52	category, criminal justi...
Forest Fires	31.3 KB	517	15	15	numeric, ecology
German Credit Data	177.6 KB	1000	21	21	category, finance, fair...
Hair section	18.2 MB	3250	833	833	none, spectral, hyp...
Liver cirrhosis - spectral image	3.4 MB	1078	546	546	none, spectral, hyp...
CD8+ in chronic viral infection	14.6 MB	9197	22193	22193	none, mouse, expre...

Description

Iris (1936), from [UCI ML Repository](#)

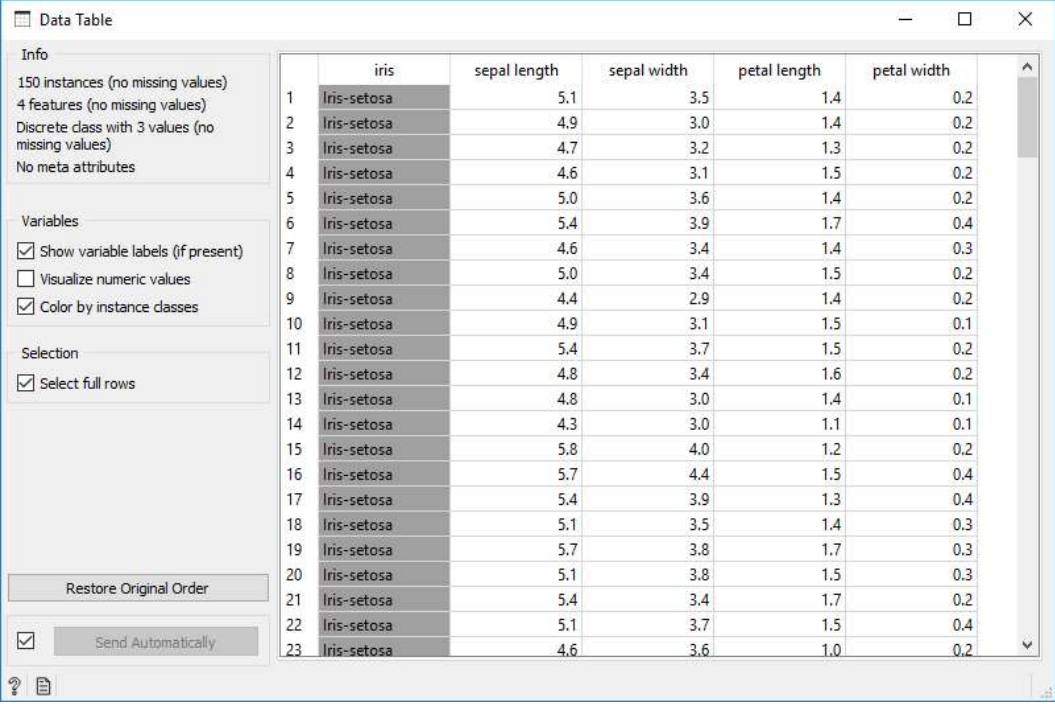
The Iris flower data set or Fisher's Iris data set was introduced by the British statistician and biologist Ronald Fisher in his 1936 paper as an example of linear discriminant analysis. The data on length and width of petal and sepal leaves was actually collected by American botanist Edgar Anderson to quantify the morphologic variation of Iris flowers of three related species.

References

R. A. Fisher (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7(2):179-188.

☐ **Send Data**

Viewing Iris dataset in data table



Data Table

Info
150 instances (no missing values)
4 features (no missing values)
Discrete class with 3 values (no missing values)
No meta attributes

Variables
☒ Show variable labels (if present)
☐ Visualize numeric values
☒ Color by instance classes

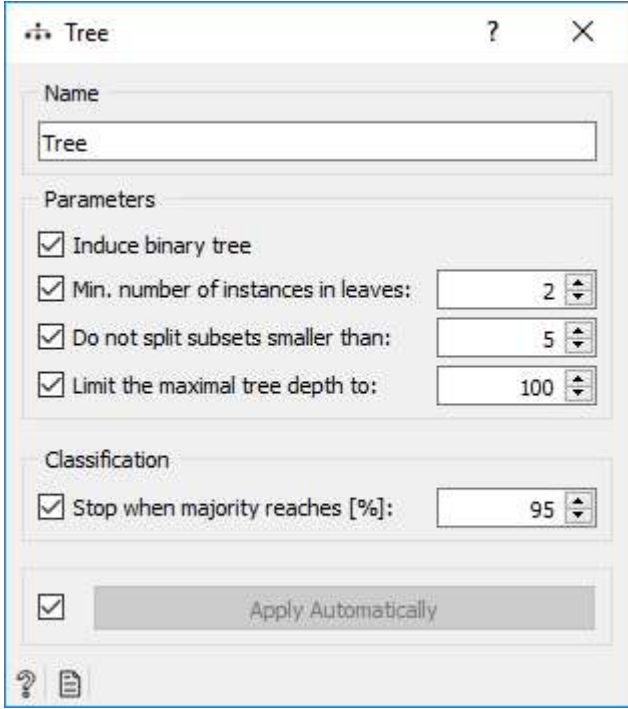
Selection
☒ Select full rows

Restore Original Order

☒ Send Automatically

	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris-setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris-setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.8	4.0	1.2	0.2
16	Iris-setosa	5.7	4.4	1.5	0.4
17	Iris-setosa	5.4	3.9	1.3	0.4
18	Iris-setosa	5.1	3.5	1.4	0.3
19	Iris-setosa	5.7	3.8	1.7	0.3
20	Iris-setosa	5.1	3.8	1.5	0.3
21	Iris-setosa	5.4	3.4	1.7	0.2
22	Iris-setosa	5.1	3.7	1.5	0.4
23	Iris-setosa	4.6	3.6	1.0	0.2

Applying tree Algorithm on Iris dataset



Tree

Name
Tree

Parameters

☒ Induce binary tree

☒ Min. number of instances in leaves: 2

☒ Do not split subsets smaller than: 5

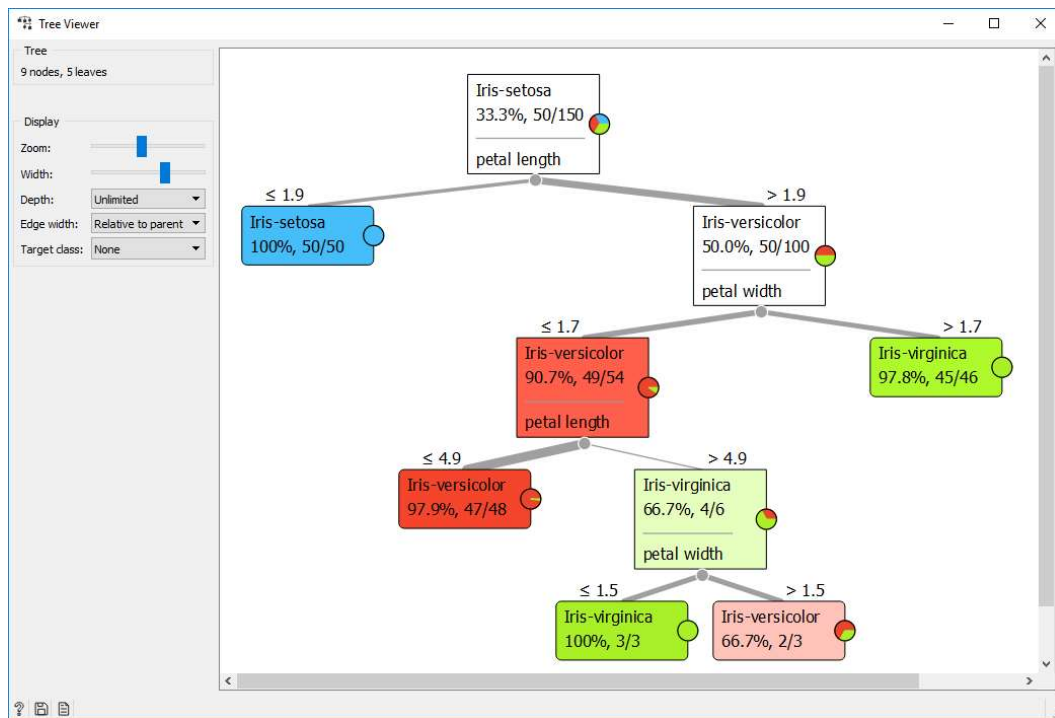
☒ Limit the maximal tree depth to: 100

Classification

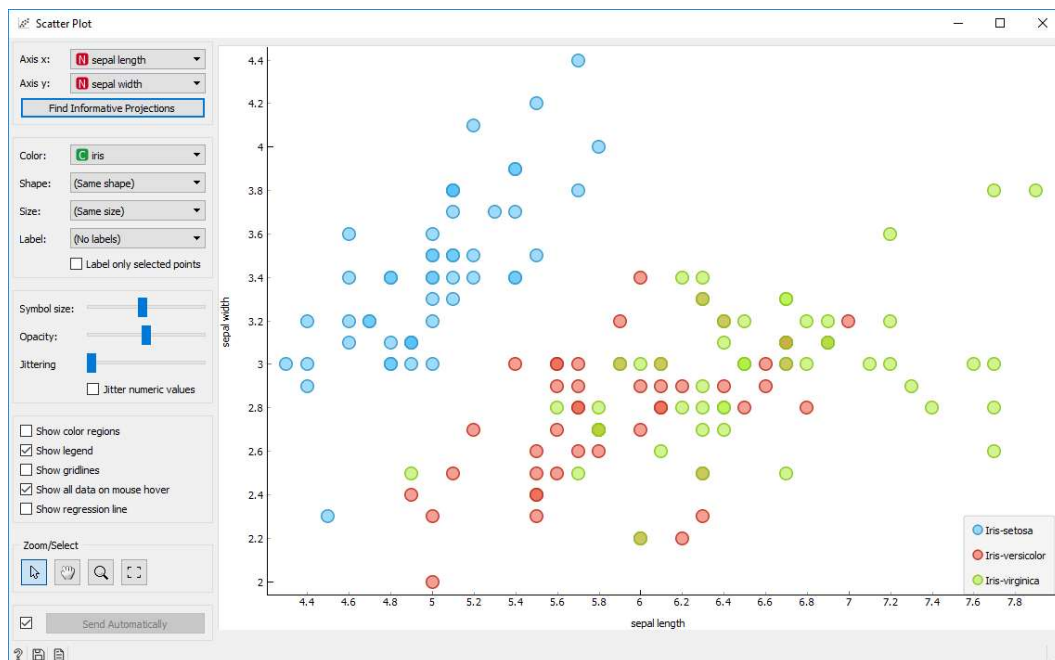
☒ Stop when majority reaches [%]: 95

☒ Apply Automatically

Tree visible in Tree Viewer



Scatter plot based on the Root node



Tuples visible in data Table based on node selected in decision tree

Data Table (1)						
Info						
150 instances (no missing values)						
4 features (no missing values)						
Discrete class with 3 values (no missing values)						
No meta attributes						
Variables						
<input checked="" type="checkbox"/> Show variable labels (if present)						
<input type="checkbox"/> Visualize numeric values						
<input checked="" type="checkbox"/> Color by instance classes						
Selection						
<input checked="" type="checkbox"/> Select full rows						
Restore Original Order						
<input checked="" type="checkbox"/> Send Automatically						
	iris	sepal length	sepal width	petal length	petal width	
1	Iris-setosa	5.1	3.5	1.4	0.2	
2	Iris-setosa	4.9	3.0	1.4	0.2	
3	Iris-setosa	4.7	3.2	1.3	0.2	
4	Iris-setosa	4.6	3.1	1.5	0.2	
5	Iris-setosa	5.0	3.6	1.4	0.2	
6	Iris-setosa	5.4	3.9	1.7	0.4	
7	Iris-setosa	4.6	3.4	1.4	0.3	
8	Iris-setosa	5.0	3.4	1.5	0.2	
9	Iris-setosa	4.4	2.9	1.4	0.2	
10	Iris-setosa	4.9	3.1	1.5	0.1	
11	Iris-setosa	5.4	3.7	1.5	0.2	
12	Iris-setosa	4.8	3.4	1.6	0.2	
13	Iris-setosa	4.8	3.0	1.4	0.1	
14	Iris-setosa	4.3	3.0	1.1	0.1	
15	Iris-setosa	5.8	4.0	1.2	0.2	
16	Iris-setosa	5.7	4.4	1.5	0.4	
17	Iris-setosa	5.4	3.9	1.3	0.4	
18	Iris-setosa	5.1	3.5	1.4	0.3	
19	Iris-setosa	5.7	3.8	1.7	0.3	
20	Iris-setosa	5.1	3.8	1.5	0.3	
21	Iris-setosa	5.4	3.4	1.7	0.2	
22	Iris-setosa	5.1	3.7	1.5	0.4	
23	Iris-setosa	4.6	3.6	1.0	0.2	