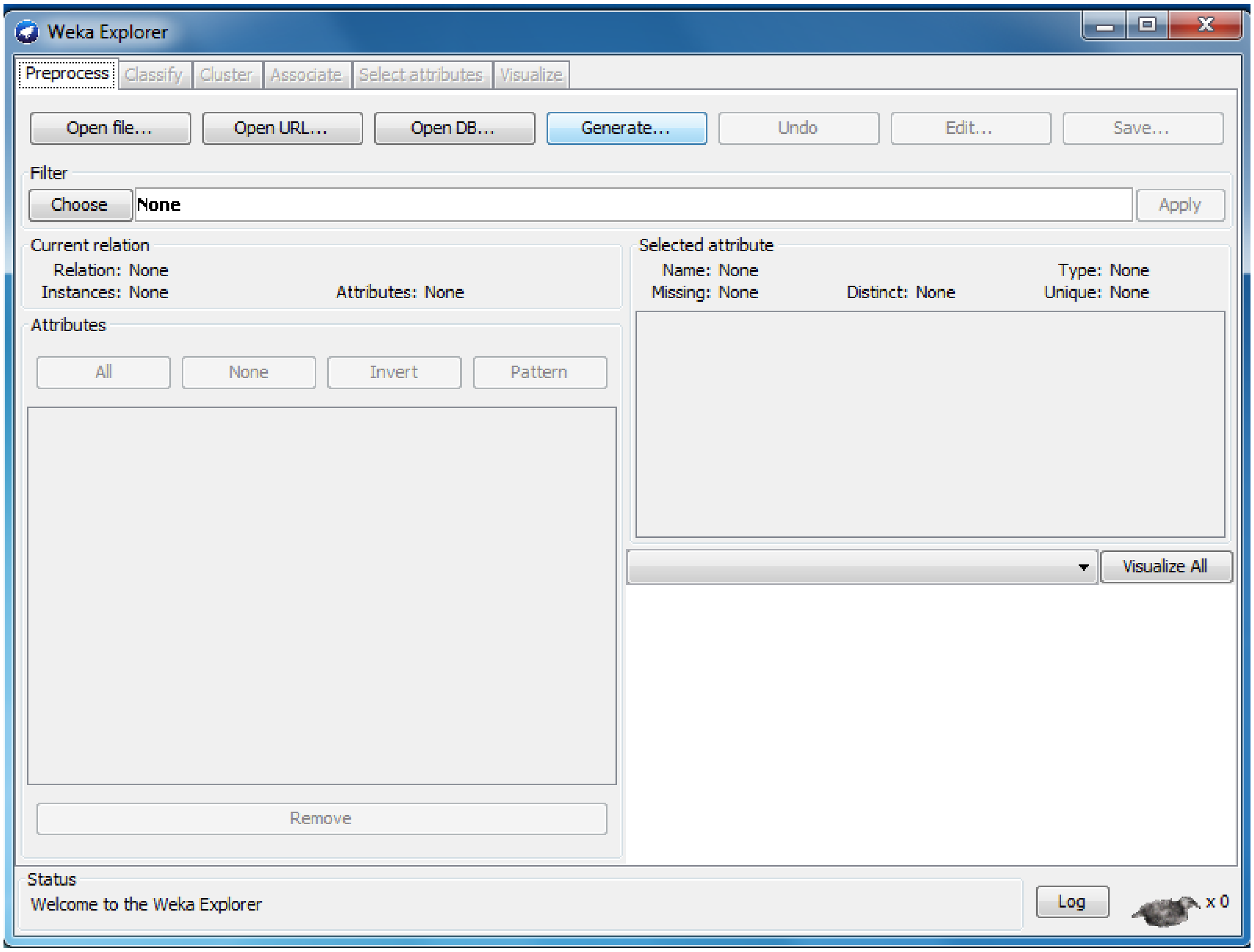
Lab10. WEKA Machine Learning Software – experiments on Diabetes data set

WEKA software is already installed on the lab computers. To launch the software, go to **'C:\Program Files\Weka-3-6\'**, double click on Weka 3.6 . To install it at home, go to this website:

http://www.cs.waikato.ac.nz/ml/weka/.

After you start the software, click on the "Explorer" button, which will open GUI that enables you to load a file, preprocess and visualize it, and run different supervised and unsupervised data mining algorithms on it. Learn how to use "Explorer GUI" – it is very user friendly and should not take long.



Filter: None

Current relation
Relation: None
Instances: None
Attributes: None

Selected attribute
Name: None
Type: None
Missing: None
Distinct: None
Unique: None

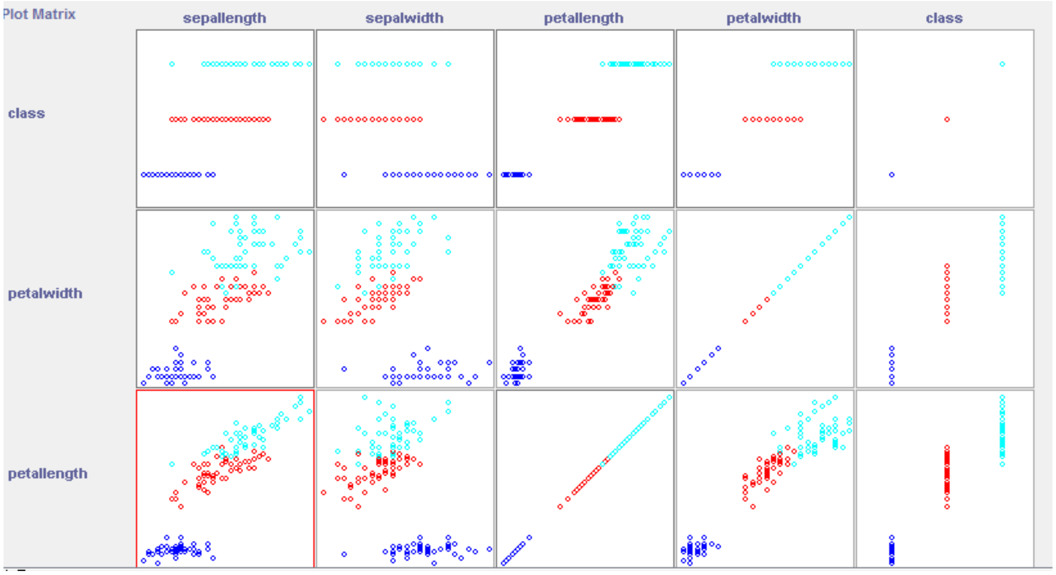
Attributes

Status
Welcome to the Weka Explorer

Task 1. Example with iris.arff data

Task 1.1. Find file **iris.arff** in the "data" folder of WEKA software **'C:\Program Files\Weka-3-6\data\'**. Open the file in text editor (e.g., notepad) and study the content of the file. The description of .arff format can be found at http://www.cs.waikato.ac.nz/ml/weka/arff.html. How many examples (i.e., instances) and how many attributes are there? What is this data set about -- explain in a few sentences.

Task 1.2. Now, open the file in WEKA by clicking on the “Open file…” button (you need to direct it in the "data" folder of WEKA software). Look at the histogram of each attribute (pay attention to blue, cyan and red colors in the histogram). Which attributes seem to be the most discriminative? Go to "Visualize" tab where you can examine the scatterplot (as illustrated as below). Which two attributes do you think are most correlated? Why?



Task 1.3. Click the "Classify" tab, where you will be able to "Choose" a classifier. Start with "ZeroR" classifier under ‘rules’ choice, which simply predicts all examples to be of the majority class. Use the "Percentage split" option. Press "Start". What is the observed accuracy on test data?

Task 1.4. Next, choose J48 algorithm under "trees". Use the default values. Press "Start". What is the accuracy? What is the question at the root of the tree? How deep is the tree?

Task 1.5. Let us select some other option by clicking on the J48 window (the input box besides “Choose” button). Select minNumObj = 10 and rerun the tree learning. What does this option mean? What is the difference in the resulting tree and its accuracy? Go back and put back minNumObj to its default value of 2. Now, select reducedErrorPrrunnig = True. What does this mean? Rerun the tree learning. What is the difference in the resulting tree and its accuracy? We can also select Cross-validation (10 fold) instead of Percentage split. Rerun the tree learning with this option (it will train 10 trees). Report the results.

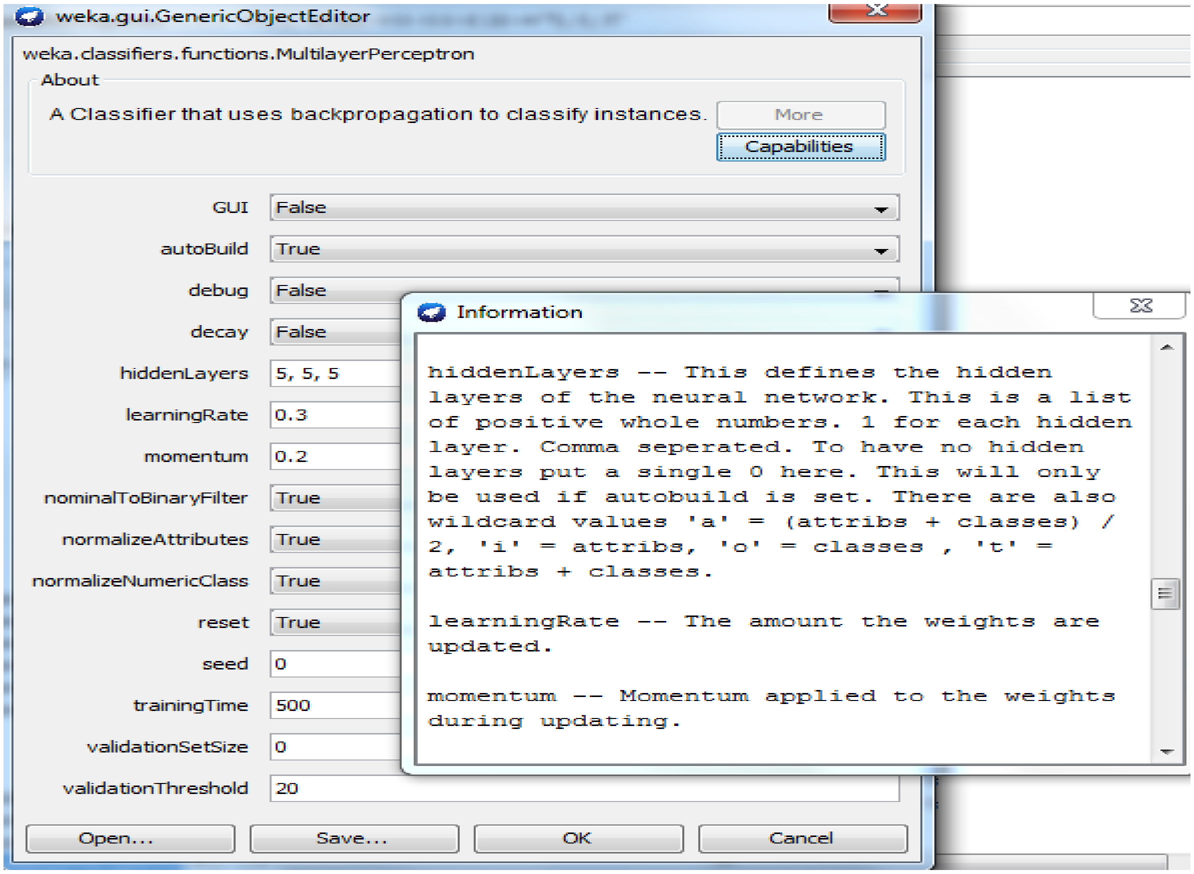
Task 1.6. Let us select the RandomForest algorithm under “trees”. This will create a large number of trees and use voting as the final prediction. How many trees are used by default? Run the program. How long did it take? What is the observed accuracy?

Task 1.7. Let us select IBk under “lazy” option. This is the k-NN algorithm. By default, k=1. Run the algorithms and observe the accuracy. Rerun it with k=5. Is there any difference in accuracy?

Task 1.8. Let us select Logistic under “functions” option. This is equivalent to a single sigmoid neuron and very similar to linear regression. Using default setting and run the algorithm and report the accuracy.

Task 1.9. Select MultilayerPerceptron option under “functions” option. Set “hiddenLayers” value to be 5, which will choose 1 hidden layer with 5 neurons. Run the training. How much time does it take and what is the observed accuracy? Set “hidderLayers” value to be (5,5,5) (no space, no parenthesis as shown in the following figure). What does it mean? Run the training. How much time does it take and what is the observed accuracy?

Ways to see definition of the “hiddenLayers” hyperparameters (and of all other hyperparameters) by clicking on the “More” button:



weka.classifiers.functions.MultilayerPerceptron

About

A Classifier that uses backpropagation to classify instances.

GUI: False

autoBuild: True

debug: False

decay: False

hiddenLayers: 5, 5, 5

learningRate: 0.3

momentum: 0.2

nominalToBinaryFilter: True

normalizeAttributes: True

normalizeNumericClass: True

reset: True

seed: 0

trainingTime: 500

validationSetSize: 0

validationThreshold: 20

hiddenLayers -- This defines the hidden layers of the neural network. This is a list of positive whole numbers. 1 for each hidden layer. Comma seperated. To have no hidden layers put a single 0 here. This will only be used if autobuild is set. There are also wildcard values 'a' = (attribs + classes) / 2, 'i' = attribs, 'o' = classes , 't' = attribs + classes.

learningRate -- The amount the weights are updated.

momentum -- Momentum applied to the weights during updating.

Task 1.10. Look back at all the results. Create a table like below. Which is the best algorithms and which is the worst algorithm by accuracy? Did you expect it and why?

Classifier	***Accuracy by default***	***Tuning***	***Accuracy after tunning***	***Running time(sec)***
ZeroR	75.58	None	75.58	0
J48	84.38	Change minimum number of objects on leaf from 2 to 4	84.89	0.22
IBk	81.7	Using 15 nearest neighbors instead of 1 nearest neighbor	87.7	0.03
MultilayerP	83.44	1) 3 hidden layers rather than 1 2) Learning rate 0.4 rather than 0.3 and with decay 3) 2000 iterations rather than 500	89.44	0.01
RandomForest	82.64	Change the minimum number of objects of leaf from 2 to 4, this made accuracy increase 0.04, then increased the number of trees from 30 to 60, accuracy increased 0.44	86.64	567.99
LogisticReg	…	…	…	…

Task 2. Repeat all steps in Task 1 on diabetes.arff data set (it’s still under the same folder as iris.arff). Answer the questions from 1.1 to 1.10.

Submission: submit your report through blackboard.