Lab Exercise 1 – Supervised Learning

# Download and Install

Download the weka package (stable version):

[http://www.cs.waikato.ac.nz/ml/weka/downloading.html](https://webmail.cdm.depaul.edu/owa/redir.aspx?C=Sji860xxtkypBRvTjdOxdl2v7VOip9EIyDW3M1BQ1CCP2blYx-CmxzbY7KhjAxq5gWsgVmCJgGc.&URL=http%3a%2f%2fwww.cs.waikato.ac.nz%2fml%2fweka%2fdownloading.html" \t "_blank)

It's an easy to use GUI for doing machine learning. Follow the instructions for your OS.

# Getting Started

To launch: click on the .app icon for Mac, or in windows, click on the weka.jar file from the install directory. Then click on the “Explorer” button. You can load an arff file (their custom format) from the pre-process tab. Click on “Open file…”, pick the arff file, and you should see it load the file attributes in attributes pane. You can look at the distribution of the data for each attribute by clicking on it.

Before doing analysis of the data, it’s important you specify the “class” attribute using the right hand pane under “Preprocess”. This is the variable in the data that you are trying to predict. In a lot of the data files this will already be specified.

To use the different machine learning algorithms, you will need to use the “Classify” tab, even when doing regression (predicting a continuous attribute). To pick an algorithm, click on the “Choose” button. You’ll see a tree structure. The location of the different algorithms below are indicated with respect to this tree structure. To change any of the parameters for each algorithm, click on the text box to the right of the “Choose” button, that contains the name of the currently selected algorithm.

# Exercises

Then undertake the following exercises. You will also find the datasets referenced below in a zip file in the course materials in D2L under Content.

* Load play tennis dataset into weka
  + <http://axon.cs.byu.edu/~martinez/classes/478/stuff/labhints/tennis.arff>
  + Train k-nearest neighbor on the data (classifiers > lazy > lbk)
    - Try various values of k. Which has the lowest error?
    - Is this a good classifier for this dataset?
  + Train a Decision Tree (j48) on the data set (classifier > trees > j48)
    - What does the resulting tree look like?
      * (right click on the tree in the results list, and select ‘visualize tree’
    - Does it make sense?
    - What’s the classification error?
    - Why do you think that is?
  + Train a pruned DT on the dataset
    - (Click on j48, set reducedErrorPruning -> true)
    - What does the tree look like? Why?
      * Note: You can right click on the result and get a better visualization of the tree
    - What’s the classification error?
    - Look at the confusion matrix. What do you see?
* Load the iris dataset into weka
  + <http://mlr.cs.umass.edu/ml/machine-learning-databases/iris/iris.data>
    - (Save as a ‘.arff’ file)
    - Classification exercise (labels are nominal) with inputs that are continuous
  + Train k-nearest neighbor on the data
    - Change distance weighting from none to “Weight by 1/distance”.
    - How does the mean absolute error change with that setting on vs off? Why?
  + Train a j48 Decision Tree on it
    - What’s the classification error?
    - What’s the tree look like?
  + Train a pruned j48 Tree
    - What’s the classification error?
    - How has the tree changed shape?
* Larger Dataset - Kaggle Titanic Data
  + <https://github.com/birchsport/titanic/blob/master/test.arff>
    - Data is **imbalanced** (290 survived, 1911 died)
  + Train knn on this dataset for a few different values of k (1,5,10,20)
    - Does upping k affect the classification accuracy much?
    - Why do you think this is?
    - Change distance weighting from none to “Weight by 1/distance”.
    - Does this improve the accuracy?
    - Why \ Why not?