Machine Learning and Data Mining with Weka

What you need

1. Program: Weka
2. How to install Weka: Go to this website http://www.cs.waikato.ac.nz/ml/weka/downloading.html and download the weka version that is supported by your Operating System (OS).
3. Now you should be good to run weka.

What will be Covered in this Lab

This lab will go over the very basics of what data mining and machine learning is. Also it will go over what Weka is and show you how to run a very simple test in Weka. We will go over a couple of the classifiers that are used in data mining and in machine learning.

What is Machine Learning and Data Mining?

Machine learning is the ability for a computer to be able to learn and predict the outcome of a dataset based on the information provided by that dataset. Data mining is very similar to machine learning since it uses many of its techniques in an attempt to find the best possible way for a machine to solve a real world problem using machine learning algorithms. Data mining however tends to deal with more issue of scalability, like how to mine from a large number of tuples that cannot be fed into algorithms directly, or of a statistical aspect, like finding out how many people are willing to buy your brand given seemingly random information while machine learning is purely for finding ways for machines to operate with little to no human interactions.

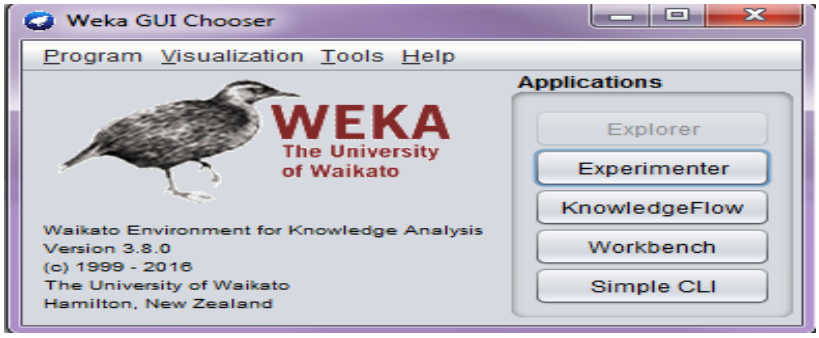
What is Weka?

Weka is a machine learning program that runs on Java. It is a very powerful program that is capable of using predefined algorithms to predict possible outcomes of supplied datasets. It simplifies the processes of machine learning and data mining by giving the user an easy to use interface to navigate through.

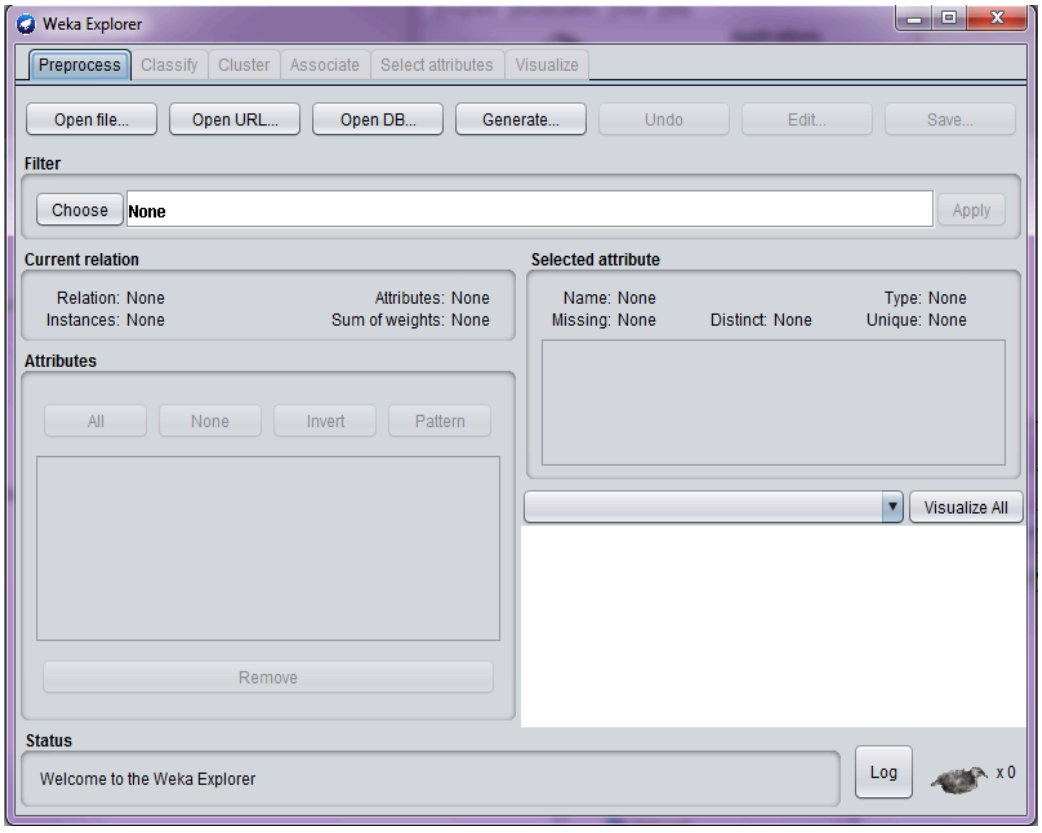
How to use Weka

The Explorer button

Once you have downloaded weka start it either by clicking on the start icon, or by going to where you had downloaded weka and start it with the weka <version> shortcut. Now you should see a screen similar to this.



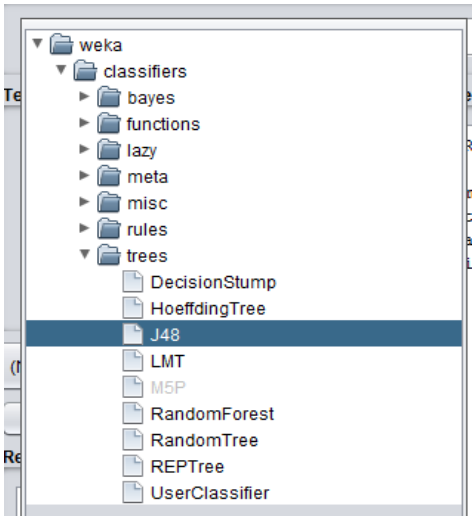
Now click on the *Explorer* button and you should see this new window.



Now we can run some tests to see how Weka would handle a well defined dataset with its predefined classifiers. Open a file called *weather.nominal.arff*. This file can be found at Weka-version\data, for example: C:\Program Files\Weka-3-8\data, in the Preprocess tab you can

see the many predefined attributes that this file had already been given. Note: if you cannot find this file for some reason you can download it from their github at this link: https://gist.github.com/myui/2c9df50db3de93a71b92

Here you can edit these attributes in anyway that you would like, but for now we will leave them as is. Now we can run a test and see how Well weka can predict the outcome defined by this dataset by going to the *Classify* tab. Here we can choose a classifier to run over this data set. Select *J48* from the trees classifiers.



Now click *start* and you should see a similar output in the classifier output box as shown here.

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

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.556    0.600    0.625      0.556    0.588      -0.043   0.633     0.758     yes
                 0.400    0.444    0.333      0.400    0.364      -0.043   0.633     0.457     no
Weighted Avg.    0.500    0.544    0.521      0.500    0.508      -0.043   0.633     0.650

=== Confusion Matrix ===

 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no
```

Here we can see some very cool information that was collected from running J48. We can see that it correctly classified 7 of the 14 total instances. The kappa statistic gives the result of the revealed accuracy versus the accuracy that was expected. The mean absolute error is given by the error between the the true value and what was expected. The root mean squared error is given by the difference between the value that the classifier got and the value it predicted. The relative absolute error is given by the magnitude between the true value and the predicted value. The root relative squared error is the error given by the absolute error divided by the ZeroR's error. ZeroR is a classifier that justs takes into consideration the top attribute in deciding what the outcome should be. Considering the small size of the dataset and the few attributes this is a fairly good output, but we can do better.

Next, choose the *NaiveBayes* classifier found under the Bayes classifiers. Click start and you should get a similar output as shown here.

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

Correctly Classified Instances           8               57.1429 %
Incorrectly Classified Instances         6               42.8571 %
Kappa statistic                         -0.0244
Mean absolute error                      0.4374
Root mean squared error                  0.4916
Relative absolute error                 91.8631 %
Root relative squared error             99.6492 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778    0.800    0.636      0.778    0.700      -0.026   0.578     0.697     yes
                 0.200    0.222    0.333      0.200    0.250      -0.026   0.578     0.557     no
Weighted Avg.    0.571    0.594    0.528      0.571    0.539      -0.026   0.578     0.647

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 4 1 | b = no
```

It appears that NaiveBayes yield a slightly better prediction on this dataset, although Naivebayes will not always get a better outcome.

Weka Classifiers

Classifiers are algorithms that are able to classify new data based upon the attributes from the dataset. There are many many classifiers in Weka and for that reason we will not go over all of the classifiers.

J48

This classifier is based off of the C4.5 algorithm, which is referred to by many as a statistical classifier. It basically builds a tree with leaves that branch out with each new attribute and gives a + or - based on what is gathered from the test.

Naive Bayes

This classifier is based off of Bayes theorem, which states that the probability of an event happening may be determined by a given condition related to that event. This is summarized in a formula such as: P(A|B) = (P(B|A)P(A))/P(B).

Random Forest

Random forest is an extremely good classifier when it comes to having a lot of noise in a dataset. Noise is referring to the unwanted data that is being mixed in with the data that you want to use to predict accurately what the outcome should be. It is similar to that of J48 in that it uses trees to sort the dataset of what is and what is not important.

Weka's ability to let the user choose which classifier is best to use and to alter them with ease makes it stand out above most machine learning programs.

What can Weka be Used For?

There are many uses for Weka and data mining in general. First it can be used to help identify similarities between events and their outcomes. This can be useful in marketing, security, health fields, and many more situations. For example with data mining it is possible to identify over 50% of American citizens by their birthday, sex, and the city that they were born in.

Hands-on Tasks

Now, you know how to build a model to fit a dataset, as hands-on tasks in this lab, accomplish the following:

1. Open the *credit-g.arff* data file.
2. Build a model with 10-fold cross-validation scheme using two models, say both *NaiveBayes* and *Support Vector Machine (Classify → functions → SMO)*
    a. What is 10-fold cross validation scheme?
3. Document parameters that each algorithm used (and any changes that you made).
4. Compare the results of a few major metrics across two runs from two models: say, precision, recall, F-measure, ROC area (per class vs. weighted macro)

5. Analyze your results: does one algorithm perform better, regardless of all parameter settings? Any changes in performance per different parameters? Using a particular evaluation metric, do you see opposite results between two algorithms? Why?

Future Labs on Weka

Our future labs will go into greater details on how to use Weka with different datasets and how to modify the classifiers so that you can get the truest probability possible. We also want to go over the importance of gathering information and how collecting a good dataset is more important than how good your algorithm is. Another thing we plan to do in our future labs is to focus on how to be able to use Weka as a security tool for a network so that it could correctly identify network attacks.

Conclusion

Data mining and machine learning are very powerful tools and can be applied to just about any situations that has well defined datasets.

If you want a more indepth view of Weka for now you can watch the Youtube series that go over Weka in great detail at https://www.youtube.com/user/WekaMOOC and go to the data mining with Weka playlist.