Project 4: Machine Learning Report

One of these things is not like the other! While your other projects have been programming projects, this project is meant to provide some exposure into machine learning tools and real world data sets that are publicly available.

Namely, you will experiment with one of those aforementioned real world data sets and explore how various supervised machine learning algorithms can be applied to find patterns in the data. You will gain experience using a common data mining and machine learning library and write a report about your data set and the algorithms used.

Jump to Section

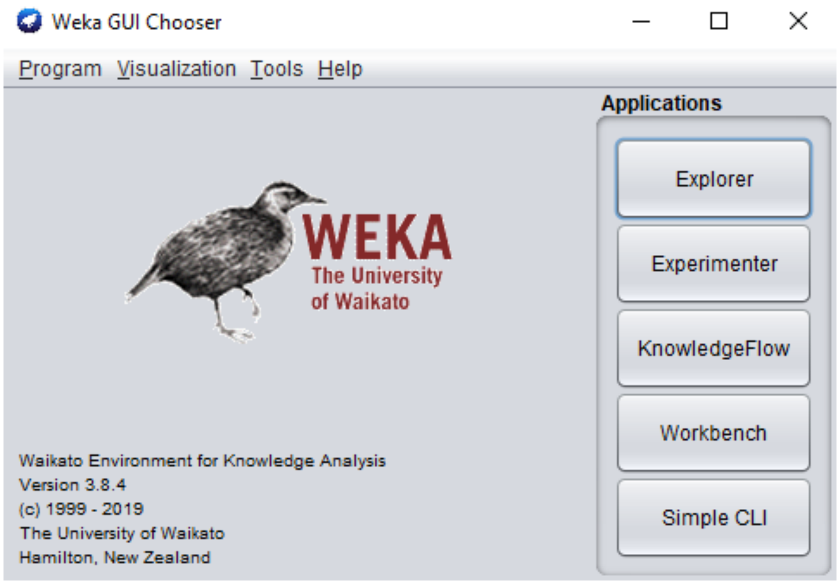
Weka Machine Learning Library

For this project, you'll be using the Weka machine learning library. The first thing you should do is click on that link and then click the download link to find the appropriate version of Weka to install for your machine. Also feel free to peruse Weka's site for an overview as to what Weka actually is! Feel free to download whatever is offered under the "Stable version" heading for the operating system of your choice (i.e., you don't need a "developer" version).

Weka provides a common framework in which many common machine learning algorithms have been implemented. Even if this is the first time *you* are learning about it, it is a well-known and frequently used tool, so there are a number of excellent tutorials that explain how to use it. Here is one page that covers many useful Weka topics, including an overview of the ARFF file type (Attribute Relationship File Format), which is how Weka wants data organized. For those of you undaunted by long manuals, here is the complete manual! Though this page is a nice summary on .arff files.

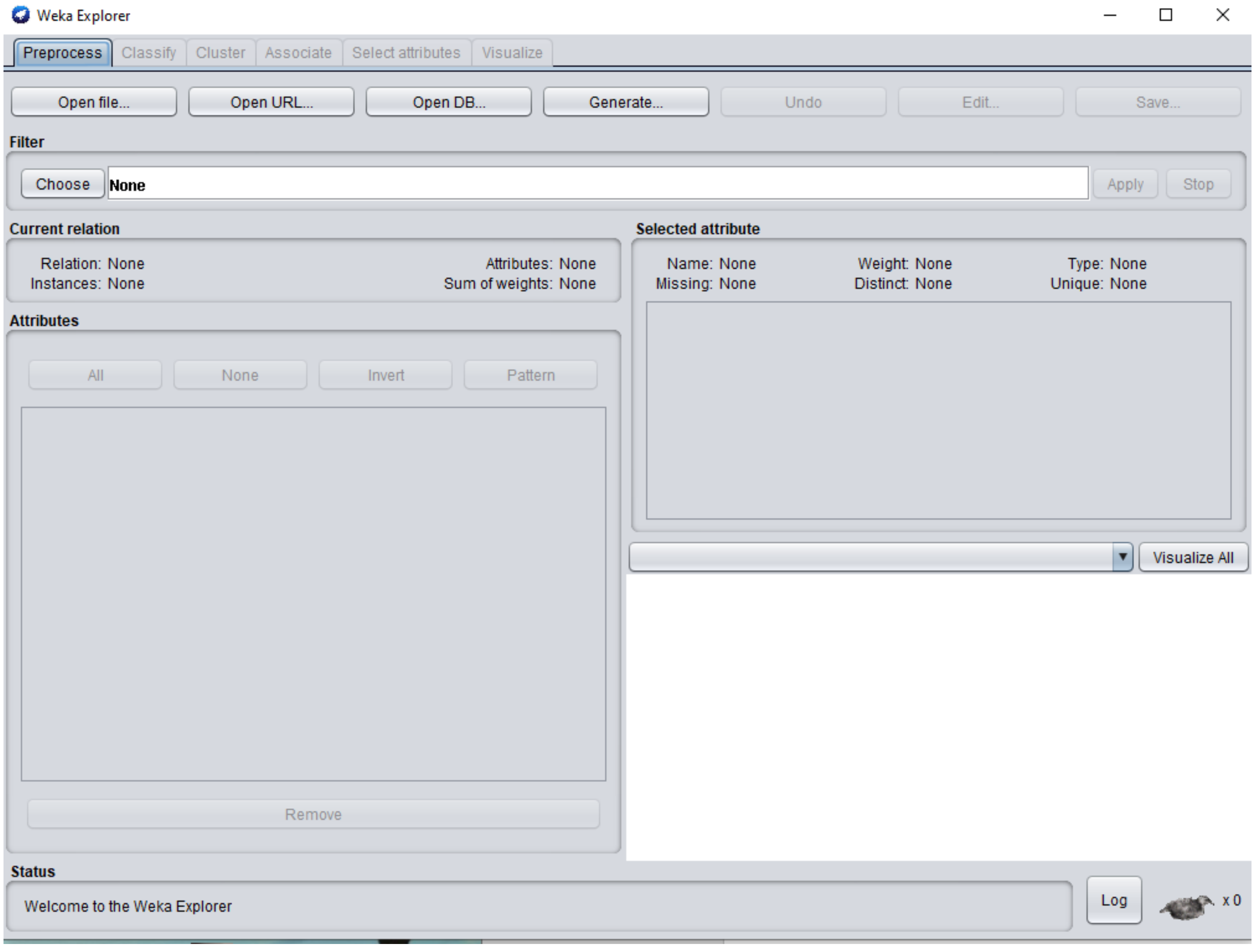
The following steps will help you get started with Weka by running two simple classifiers. After you have downloaded, installed, and started Weka…

You should see a screen with a few buttons and a Weka bird! Let's dig in!



Waikato Environment for Knowledge Analysis
Version 3.8.4
(c) 1999 - 2019
The University of Waikato
Hamilton, New Zealand

Step 1: Click the *Explorer* Button:



Filter: None

Relation: None
Instances: None
Attributes: None
Sum of weights: None

Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

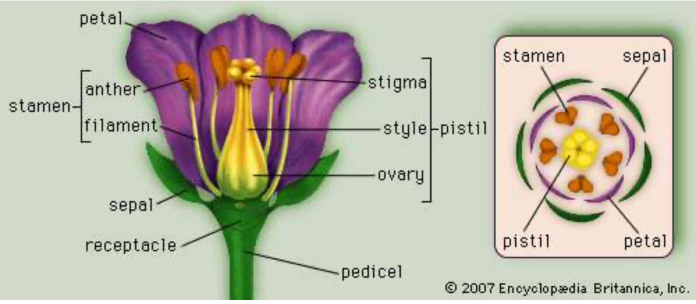
That should take you to a screen that looks like this! This window lets you load in a data set. It presents some basic statistics and visualizations for the data. For these examples we'll be working with the data set iris.arff. I'm including it on Moodle so you have it as an example, but technically you already grabbed it when you set up Weka (it should be in the "data" directory inside your installation of Weka, along with several other data sets, too)!

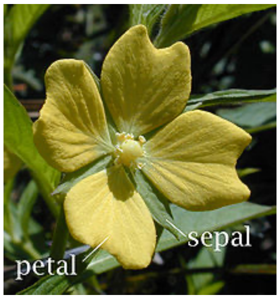
Step 2: Load the data set.

To load a data set, click on the "Open File…" button in the upper left, and choose iris.arff from wherever it currently is on your file system.

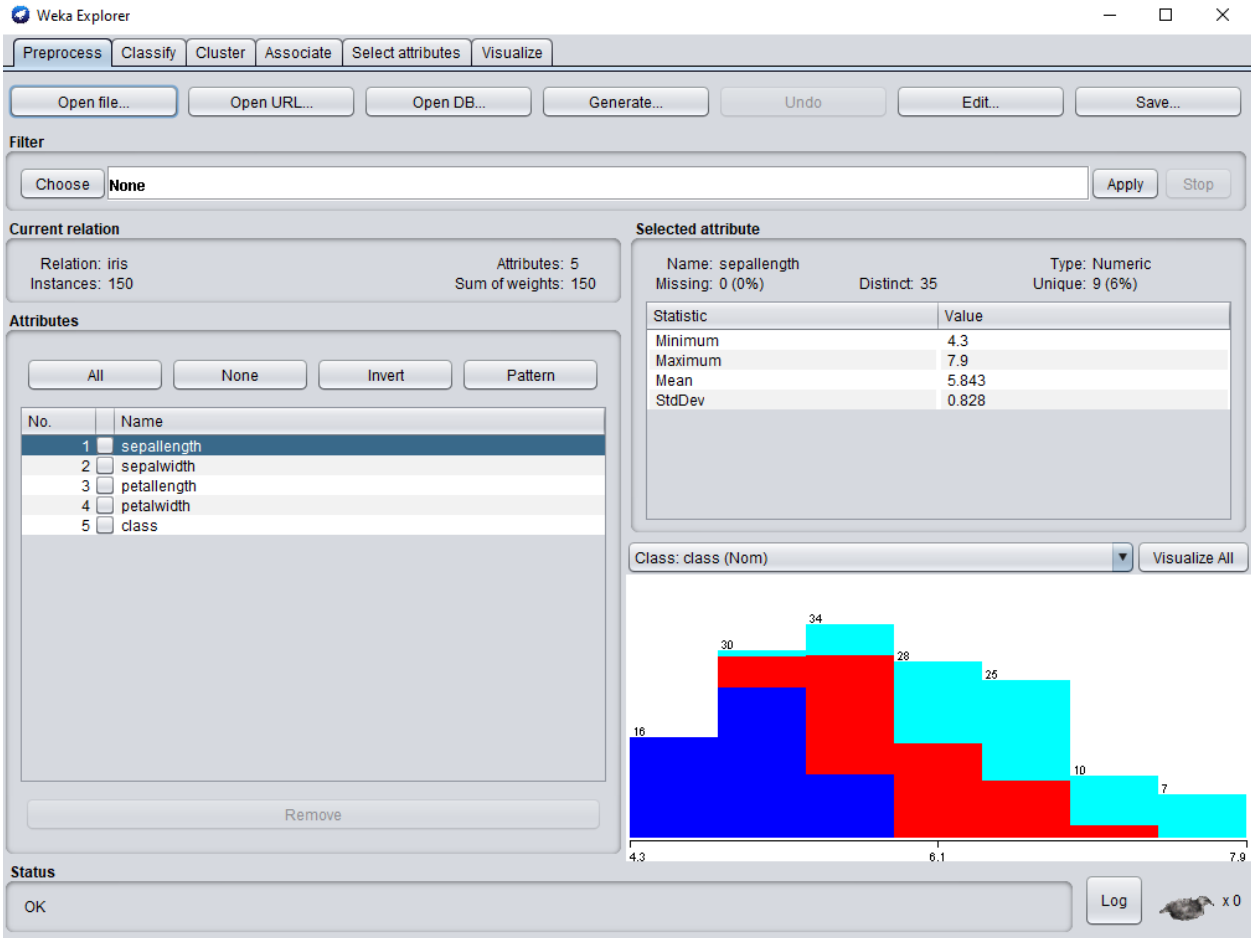
This data set is a commonly used example in machine learning. It gives 150 observations of iris flowers of three species. Each observation includes the length and width of the sepal and the petals.

Although knowledge of flowers is not needed for this assignment, here are pictures depicting what petals and sepals are.





Step 3: Admire the loaded data set.



Filter: None

Current relation

Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150

Attributes

No.		Name
1	□	sepallength
2	□	sepalwidth
3	□	petallength
4	□	petalwidth
5	□	class

Selected attribute

Name: sepallength
Missing: 0 (0%)
Distinct: 35
Type: Numeric
Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Class: class (Nom)

Status

OK

After you've loaded up the Iris data, the explorer window should look something like the above.

In machine learning, we are often concerned with **labels**. Given certain knowledge of the attributes and features of a piece of data (e.g., the petal and sepal dimensions of a flower) can we correctly predict some label to apply to that data (e.g., what species the flower is). Here, the label is indeed the species of the flower: setosa, versicolor, or virginica. This data set is **a labeled data set**, so the species of each observed flower is provided (under the “class” attribute).

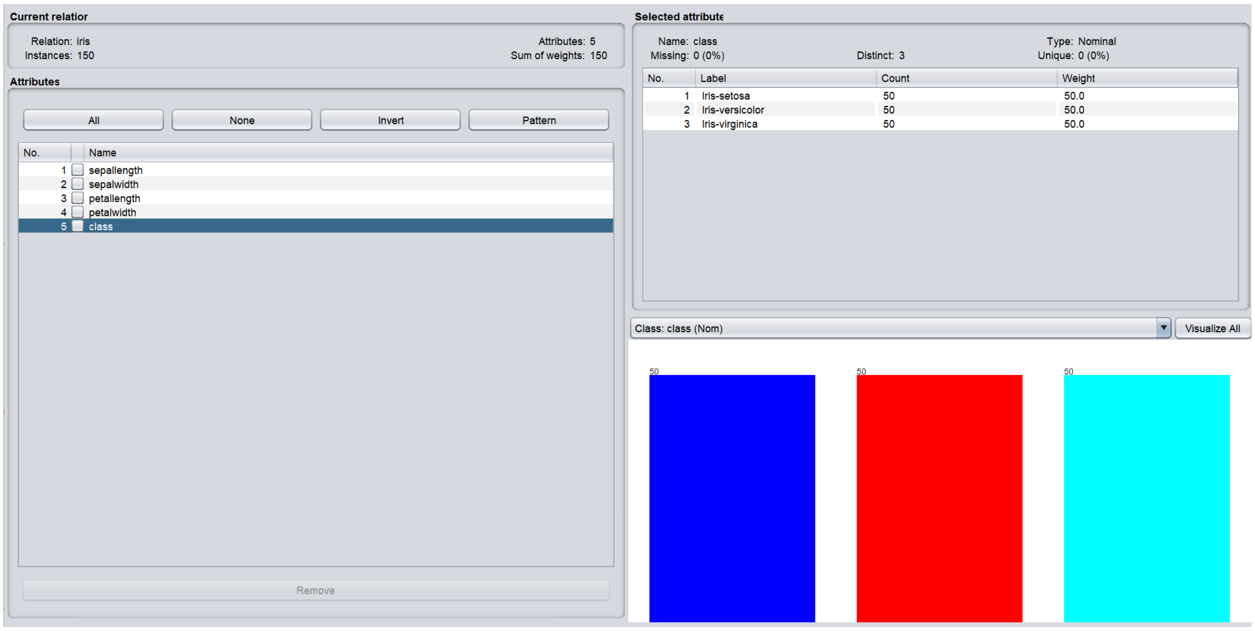
You also may find it helpful to know that Iris is a *genus* of flower, that has hundreds of different *species*. Here are pictures of the three species we are working with here.

Iris Setosa

Iris Versicolor

Iris Virginica

So, to illustrate this, if you click on the “class” attribute along the left…



Current relatior

Relation: iris
Instances: 150

Attributes: 5
Sum of weights: 150

Attributes

No.		Name
1	□	sepallength
2	□	sepalwidth
3	□	petallength
4	□	petalwidth
5	□	class

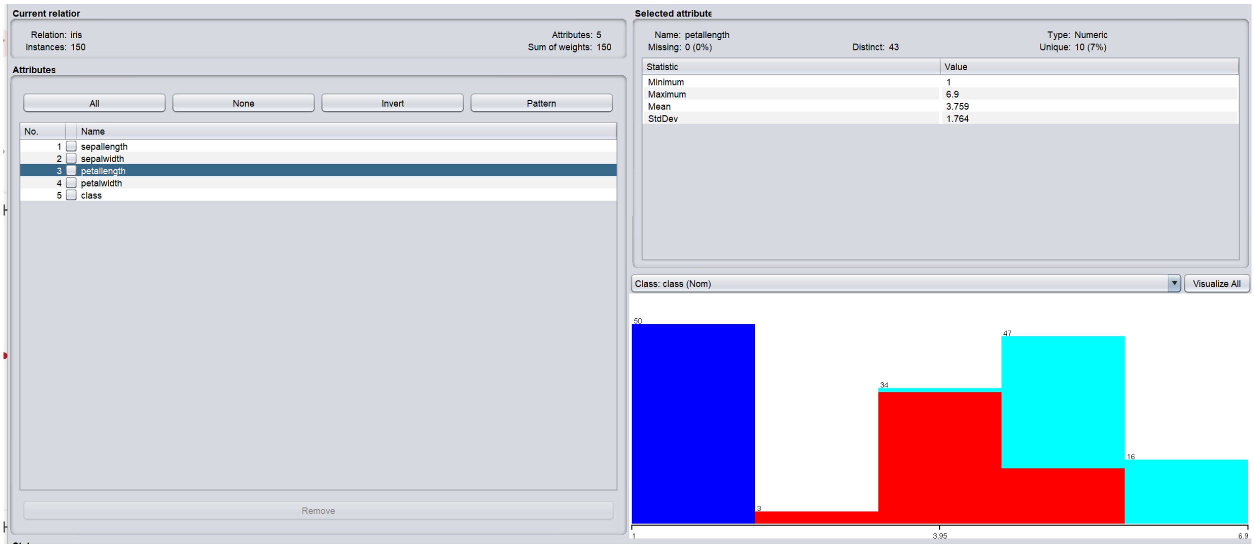
Selected attribute

Name: class
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	Iris-setosa	50	50.0
2	Iris-versicolor	50	50.0
3	Iris-virginica	50	50.0

Class: class (Nom)

You’ll see on the right that there are those three different colored bars. Those represent the fact that there are three unique “classes”, with 50 entries apiece. If you were to instead click on the petallength attribute



Current relatior

Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150

Attributes

No.		Name
1	□	sepallength
2	□	sepalwidth
3	□	petallength
4	□	petalwidth
5	□	class

Selected attribute

Name: petallength
Missing: 0 (0%)
Distinct: 43
Type: Numeric
Unique: 10 (7%)

Statistic	Value
Minimum	1
Maximum	6.9
Mean	3.759
StdDev	1.764

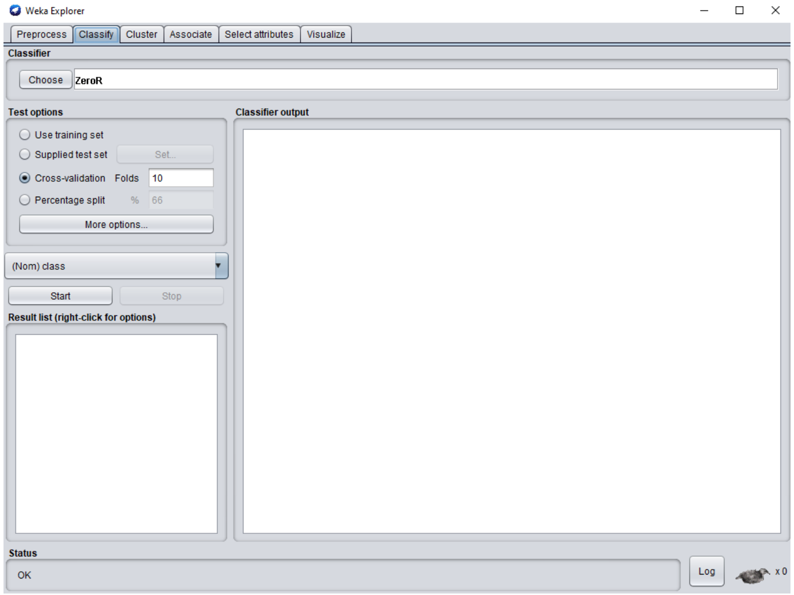
Class: class (Nom)

We can see a different graph. It's a little hard to tell in the above image, but it is saying that there are 50 examples that have a petallength between 1 and 2.18 (hover your mouse over the blue bar to see the range). The examples are color coded by their class labels, this tells us that ALL of the ires-setosa examples have shorter petals than the other types of iris.

The columns with the red and light-blue stacked together means that there were some examples of the other two types of Iris that had petallengths in the same range, though we can see that iris-virginica tended to be longer than iris-versicolor.

Anyway, that's what we're looking at! Perhaps it is already getting your brains thinking about what attributes might be best to use to classify different examples? Speaking of which…

Step 4: Click on the Classify Tab



Weka Explorer

Classifier

ZeroR

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

Classifier output

Result list (right-click for options)

Status

OK

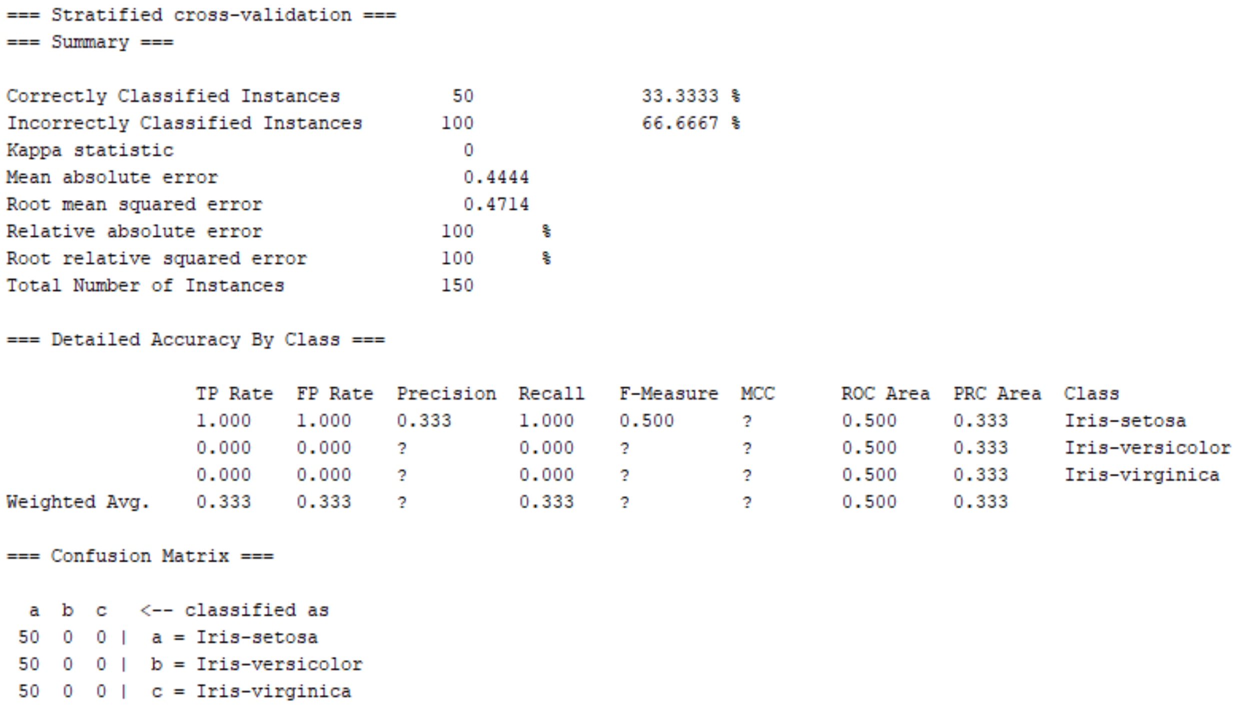
This lets you choose different classifier algorithms to run on your data! The default classifier, “ZeroR” should be selected already – you should be able to see it along the top.

There’s also lots of different test options to choose between. The default test option of “Cross-validation” with 10 folds should already be selected.

If these default options aren’t selected, ah, select them!

Step 5: Hit the Start button and try to interpret the results.

After you hit start, you should see something that looks like this pop up:



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

Correctly Classified Instances          50               33.3333 %
Incorrectly Classified Instances       100               66.6667 %
Kappa statistic                          0
Mean absolute error                      0.4444
Root mean squared error                  0.4714
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.333      1.000    0.500      ?        0.500     0.333     Iris-setosa
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.333     Iris-versicolor
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.333     Iris-virginica
Weighted Avg.    0.333    0.333    ?          0.333    ?          ?        0.500     0.333

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
 50  0  0 |  b = Iris-versicolor
 50  0  0 |  c = Iris-virginica
```

I know this is a lot all at once, but before you read the next paragraph, try taking a look at it and see if you can figure out what is going on (remember: the goal of these classifiers is to apply labels to flowers, i.e., to figure out what species each flower is). I’ll give you one more hint: ZeroR is the “majority classifier.” What do you think the above data is saying?

Did you make a guess?

ZeroR, again, is the majority classifier. It simply finds which species of flower is most common in the data set and will always return that when asked to classify a flower. Because there are exactly 50 of each kind of flower there is no majority, so it chose setosa arbitrarily. You can see this with the first line under the “detailed accuracy by class" header – it labelled everything as Iris-setosa with a 100 percent rate (but only a .333 percent precision rate – 2 third of the things it labelled as setosa were *not* setosas!).

"TP Rate" and "FP Rate" in that table stands for "True Positive Rate" and "False Positive" rate. True positive is getting at "how many things that were *actually* iris-setosa got labelled as iris-setosa?" And good news – it's 100 percent! Every iris-setosa was successfully labelled as such.

But the False Positive rate is asking "how many things that *WEREN'T* actually iris-setosa accidentally labelled as iris-setosa?" Again, we see that it's 100 percent – we mislabelled *every single other flower.* So, er, not too great.

This is further confirmed with the Confusion Matrix at the bottom. This is a nice way for us to count up how many of each piece of labelled data got classified as what. The column represents "what it got classified as" and the row represents "what it actually was."

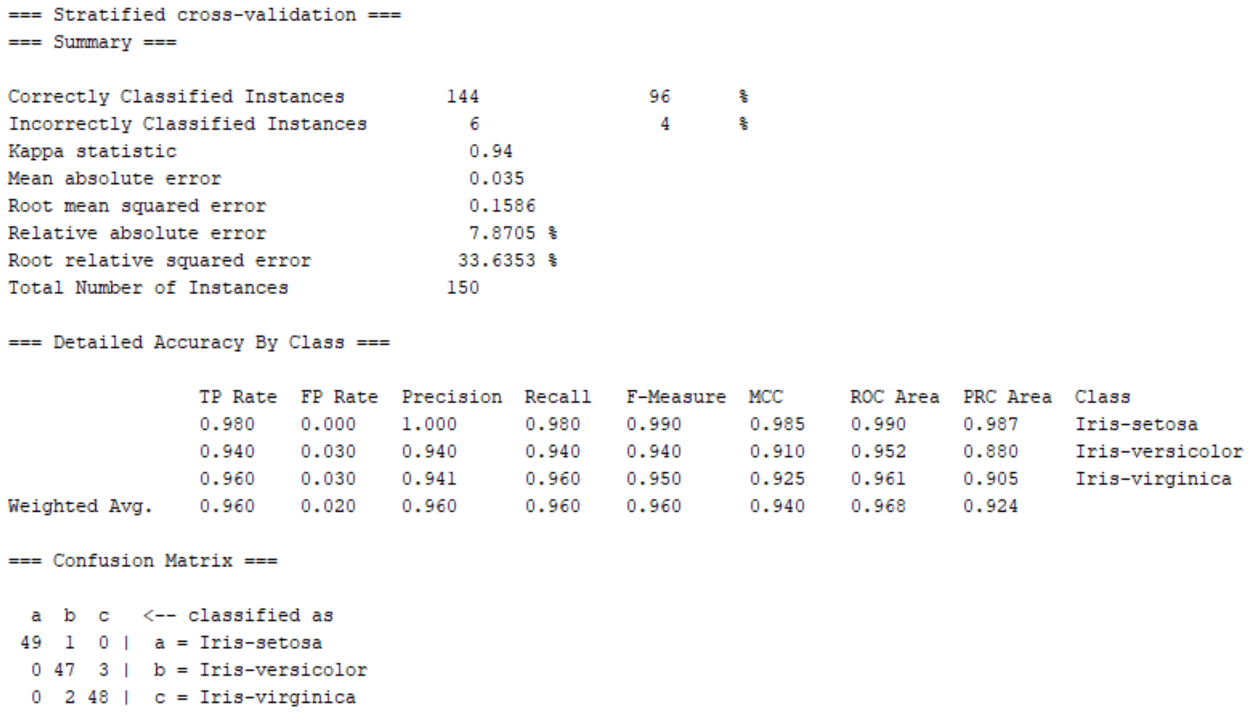
So if we look at the first row, *a* (representing iris-setosa), we see that it classified all 50 iris-setosa examples AS iris-setosa. Great!

If we look at the second row, *b* (representing iris-versicolor), we see that it classified ZERO of them as iris-versicolor, and instead all fifty of them as iris-setosa. Oops. The same story holds for the third row.

Still, correctly classifying 50 of the 150 flowers is something. This will be our baseline classifier.

Step 6: Click the "Choose" button, go to the "trees" folder, and select "J48"

Hitting the choose button revealed a lot of other classifiers that Weka can use. J48 is an improved and feature-rich implementation of the decision tree induction algorithm that we discuss in class. Keep the same test options as before and run this classifier on the data set by clicking the "Start" button.



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

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.035
Root mean squared error                  0.1586
Relative absolute error                  7.8705 %
Root relative squared error             33.6353 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.000    1.000      0.980    0.990      0.985    0.990     0.987     Iris-setosa
                 0.940    0.030    0.940      0.940    0.940      0.910    0.952     0.880     Iris-versicolor
                 0.960    0.030    0.941      0.960    0.950      0.925    0.961     0.905     Iris-virginica
Weighted Avg.    0.960    0.020    0.960      0.960    0.960      0.940    0.968     0.924

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

And you can see this one did much better! It was able to correctly classify 144 of the cases, or 96%! Much more than our baseline!

Assignment

Your assignment is to find an existing data set, or to create a data set of your own, and use Weka's supervised machine learning algorithms to find an interesting pattern in the data. You will then write a report summarizing your findings. Your report will be graded according to a grading rubric that is available on Moodle. It must contain three sections:

- **Data Set**: You must explain what your data set describes, including each attribute and the class label. You must submit the data set along with your report. If you obtained this data set from some third party source, you must cite that source. **You may not use one of the example data sets provided by Weka.** Go exploring! Though to get you started, the University of California, Irvine, has a lot of datasets that might suit your fancy! If you find a dataset that you love but that isn't in a .arff file format, Weka has some tools to convert other file formats into .arff files. You can read a little bit about converting data to .arff here! You are also welcome to use *your own data that you produce* though beware: it often takes lots of data (e.g., hundreds, if not thousands, of examples) to do interesting things.

- **Baseline Classifier**: Choose a baseline classifier such as the majority classifier above). Justify why it represents a good baseline for your classification task (this might involve researching about different types of baseline classifiers to pick a good one for you!). Describe the results.
- **Classifier(s):** Choose a more intelligent classifier (such as J48 above) and describe its results relative to the baseline.

For students enrolled **in CSCI 5525**: You must use two "intelligent" classifiers, rather than one. That is to say, you'll still use a baseline classifier, like everyone else, but then you'll two intelligent classifiers to compare the baseline against. You will also need to provide a brief description covering how these classifiers work (roughly 1 page of text per classifier).

Submission and Grading

Reports will be graded according to the grading rubric, available on Moodle.

Please observe the following when submitting your report:

- The report should be a PDF file type, 12 point font, single-spaced, with 1-inch margins. It must include your name.
- You must also submit your data set as a file which can be read by Weka version 3.8 or higher.
- If you chose to use a classifier not included with Weka, you must also submit it in a form suitable for use in Weka 3.8 or higher.

Please upload your report, data, and any other requested files to Moodle before 11:59pm the day of the deadline.