Assignment – Building and Testing Classifiers in WEKA
Due date: 11 October 2020 (Week 10)
Assessment Weight: 40%

Note: This is an individual assignment. While it is expected that students will discuss their ideas with one another, students need to be aware of their responsibilities in ensuring that they do not deliberately or inadvertently plagiarize the work of others.

In this assignment you will run a machine learning experiment using Weka. You will generate a model that predicts the quality of wine based on its chemical attributes. You will train the model on the supplied training data.

Submission Instructions

Create a document (Word or equivalent) with your answers to the questions given. Be sure you have answered all the questions. Name this file `A1-firstName-lastName.docx`.

Marks Breakdown
This assignment consists of nine (7) questions worth 80 marks, divided among two parts:

Part 1 (Q1)	10 marks
Part 2 (Q2-4)	30 marks
Part 3 (Q5-7)	40 marks

Your answers will be marked according to the rubric at the end of this document.

Task #1 - Classification for Wine
The Wine Dataset
The dataset files are provided for you:

- `wine_train.arff` (labeled training set, 1599 instances)

This dataset is adapted from:
P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. *Modelling wine preferences by data mining from physicochemical properties.* In Decision Support Systems, Elsevier, 47(4):547-553. ISSN: 0167-9236.

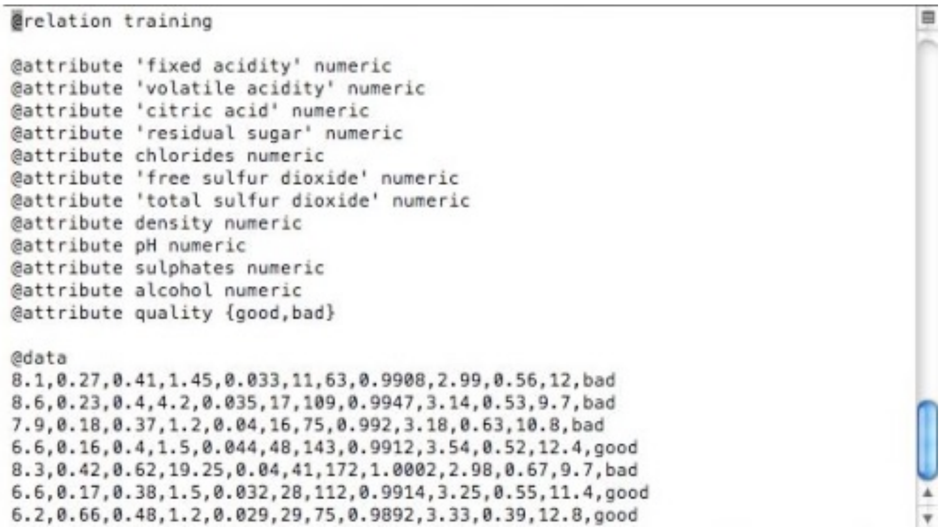
This dataset contains data for variants of the Portuguese "Vinho Verde" wine. For each variant, 11 chemical features were measured. Each of these is a numeric attribute. They are:

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides numeric
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol

Each variant was tasted by three experts. Their ratings have been combined into a single quality label: "good" or "bad" Therefore this is a *binary classification* problem with *numeric attributes*.

Task 1 - Part 1: Examine/Explore the Data [10 marks]
It is a good idea to inspect your data by hand before running any machine learning experiments, to ensure that the dataset is in the correct format and that you understand what the dataset contains.

Firstly, view `wine_train.arff`. You should see something like this:

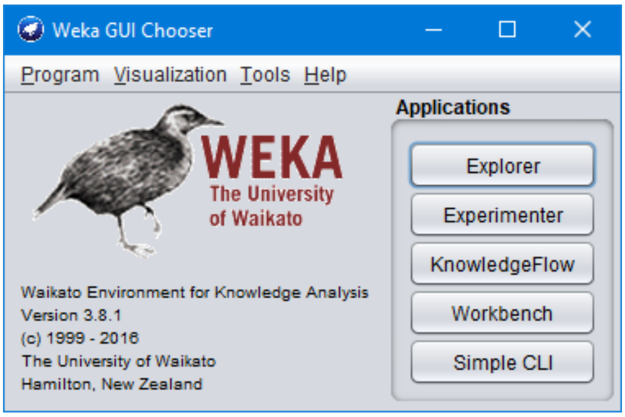


```
@relation training

@attribute 'fixed acidity' numeric
@attribute 'volatile acidity' numeric
@attribute 'citric acid' numeric
@attribute 'residual sugar' numeric
@attribute chlorides numeric
@attribute 'free sulfur dioxide' numeric
@attribute 'total sulfur dioxide' numeric
@attribute density numeric
@attribute pH numeric
@attribute sulphates numeric
@attribute alcohol numeric
@attribute quality {good,bad}

@data
8.1,0.27,0.41,1.45,0.033,11,63,0.9908,2.99,0.56,12,bad
8.6,0.23,0.4,4.2,0.035,17,109,0.9947,3.14,0.53,9.7,bad
7.9,0.18,0.37,1.2,0.04,16,75,0.992,3.18,0.63,10.8,bad
6.6,0.16,0.4,1.5,0.044,48,143,0.9912,3.54,0.52,12.4,good
8.3,0.42,0.62,19.25,0.04,41,172,1.0002,2.98,0.67,9.7,bad
6.6,0.17,0.38,1.5,0.032,28,112,0.9914,3.25,0.55,11.4,good
6.2,0.66,0.48,1.2,0.029,29,75,0.9892,3.33,0.39,12.8,good
```

The files are in ARFF (Attribute-Relation File Format), a text format developed for Weka. At the top of each file you will see a list of attributes, followed by a data section with rows of comma separated values, one for each instance.

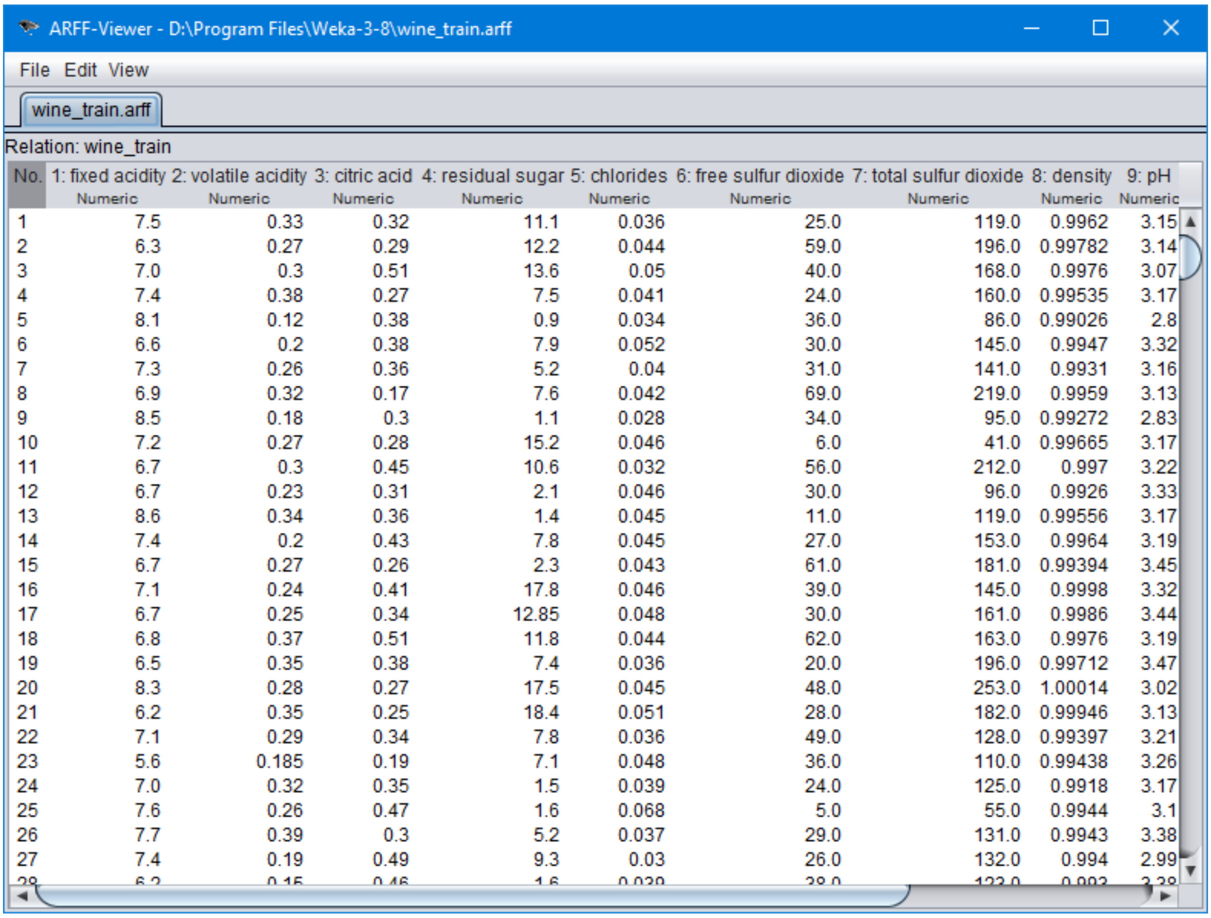
For this assignment you will not need to deal with the ARFF format directly, as Weka will handle reading and writing ARFF files for you. In future experiments you may have to convert between ARFF and another data format. (You can close the text editor.)

Another way to view .arff files is using the WEKA *ArffViewer* tool. Once you start WEKA, you will get a screen like the following:



Waikato Environment for Knowledge Analysis
Version 3.8.1
(c) 1999 - 2016
The University of Waikato
Hamilton, New Zealand

From the *Tools* menu choose *ArffViewer*. In the window that opens, choose *File→Open* and open one of the data files. You should see something like the following:



Relation: wine_train

No.	1: fixed acidity	2: volatile acidity	3: citric acid	4: residual sugar	5: chlorides	6: free sulfur dioxide	7: total sulfur dioxide	8: density	9: pH
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	7.5	0.33	0.32	11.1	0.036	25.0	119.0	0.9962	3.15
2	6.3	0.27	0.29	12.2	0.044	59.0	196.0	0.99782	3.14
3	7.0	0.3	0.51	13.6	0.05	40.0	168.0	0.9976	3.07
4	7.4	0.38	0.27	7.5	0.041	24.0	160.0	0.99535	3.17
5	8.1	0.12	0.38	0.9	0.034	36.0	86.0	0.99026	2.8
6	6.6	0.2	0.38	7.9	0.052	30.0	145.0	0.9947	3.32
7	7.3	0.26	0.36	5.2	0.04	31.0	141.0	0.9931	3.16
8	6.9	0.32	0.17	7.6	0.042	69.0	219.0	0.9959	3.13
9	8.5	0.18	0.3	1.1	0.028	34.0	95.0	0.99272	2.83
10	7.2	0.27	0.28	15.2	0.046	6.0	41.0	0.99665	3.17
11	6.7	0.3	0.45	10.6	0.032	56.0	212.0	0.997	3.22
12	6.7	0.23	0.31	2.1	0.046	30.0	96.0	0.9926	3.33
13	8.6	0.34	0.36	1.4	0.045	11.0	119.0	0.99556	3.17
14	7.4	0.2	0.43	7.8	0.045	27.0	153.0	0.9964	3.19
15	6.7	0.27	0.26	2.3	0.043	61.0	181.0	0.99394	3.45
16	7.1	0.24	0.41	17.8	0.046	39.0	145.0	0.9998	3.32
17	6.7	0.25	0.34	12.85	0.048	30.0	161.0	0.9986	3.44
18	6.8	0.37	0.51	11.8	0.044	62.0	163.0	0.9976	3.19
19	6.5	0.35	0.38	7.4	0.036	20.0	196.0	0.99712	3.47
20	8.3	0.28	0.27	17.5	0.045	48.0	253.0	1.00014	3.02
21	6.2	0.35	0.25	18.4	0.051	28.0	182.0	0.99946	3.13
22	7.1	0.29	0.34	7.8	0.036	49.0	128.0	0.99397	3.21
23	5.6	0.185	0.19	7.1	0.048	36.0	110.0	0.99438	3.26
24	7.0	0.32	0.35	1.5	0.039	24.0	125.0	0.9918	3.17
25	7.6	0.26	0.47	1.6	0.068	5.0	55.0	0.9944	3.1
26	7.7	0.39	0.3	5.2	0.037	29.0	131.0	0.9943	3.38
27	7.4	0.19	0.49	9.3	0.03	26.0	132.0	0.994	2.99
28	6.2	0.15	0.46	1.6	0.029	28.0	123.0	0.993	3.38

Here you see the same data as in the text editor, but parsed into a spreadsheet-like format. Although you will not need the ArffViewer for this assignment, it is a useful tool to know about when working with Weka. (You can close the ArffViewer window.)

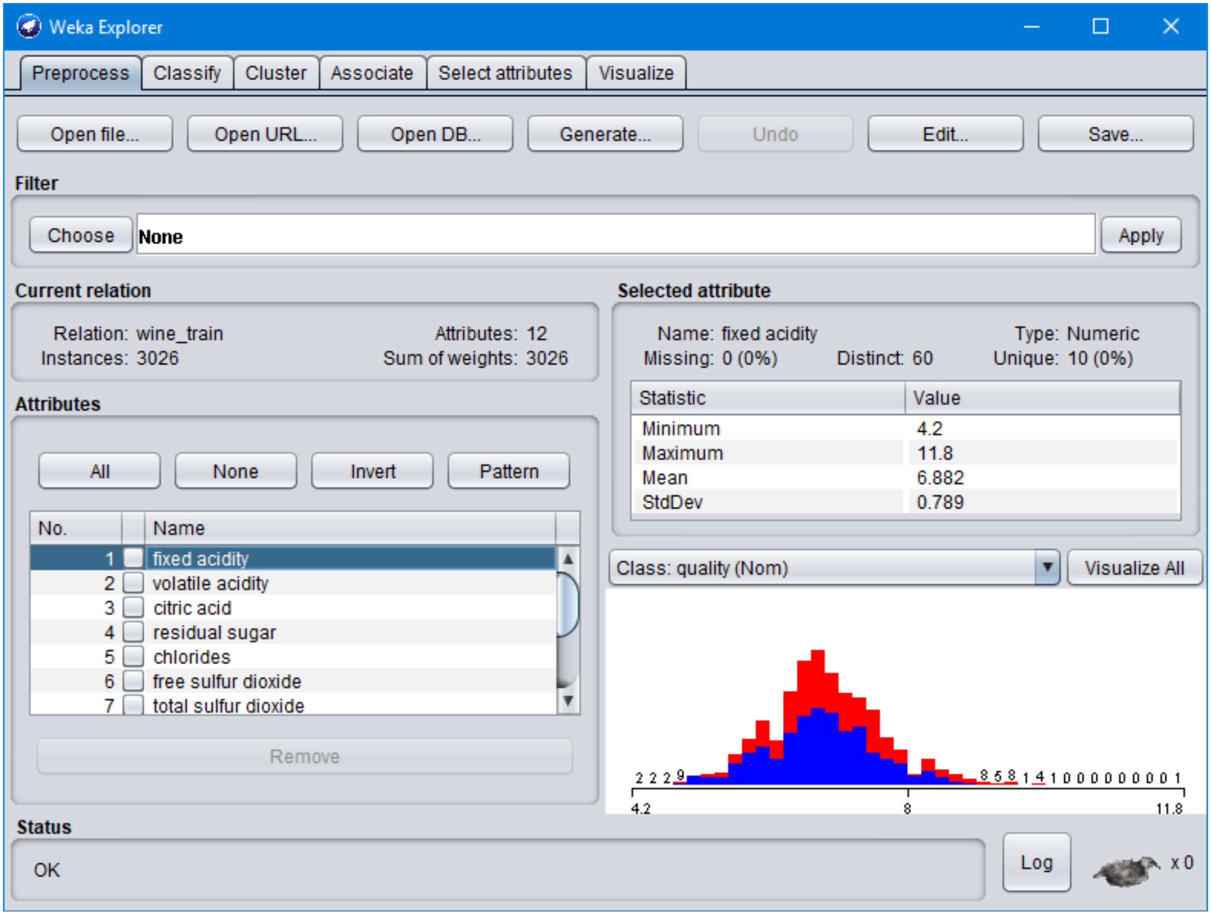
Important Note

You may find that the ARFF files are grayed out and that the *All Files* option needs to be selected from the *File Format* dropdown menu for the files to be selectable. However, the ARFF Viewer may still not read the files properly. If such is the case, it is likely that a .txt extension got appended to the filename when the files were downloaded. However, even if the files are downloaded without .txt getting appended or an inadvertently added .txt extension is removed, the ARFF Viewer may have trouble reading the files properly. The following steps should resolve the issue:

1. View the ARFF in your Web browser by clicking on the link in the instructions or open the downloaded ARFF file in a text editor.
2. Copy all the text and paste it to a new text file.
3. If you copied the ARFF contents from the downloaded ARFF file, it is recommended that you do not overwrite the downloaded ARFF file when saving the new file on the next step. Instead, delete the downloaded ARFF file.
4. Save the new text file with a **.arff** extension, carefully making sure that a .txt extension does not get appended.
5. Open the newly saved ARFF file in the Weka ARFF Viewer to verify the Viewer can display the file in the manner illustrated in the image above.

After getting the initial examination on the data files provided using a general text editor or ArffViewer tool, choose the Explorer interface in Weka. (From the Weka GUI Choose click on the *Explorer* button to open the Weka Explorer.) The Explorer is the main tool in Weka, and the one you are most likely to work with when setting up an experiment. For the remainder of this assignment you will work within the Weka Explorer.

The Explorer should open to the "Preprocess" tab. The Preprocess tab allows you to inspect and modify your dataset before passing it to a machine learning algorithm. Click on the button that says "Open file..." and open `wine_train.arff`. You should see something like this when the dataset is correctly uploaded:



Filter: None

Relation: wine_train
Instances: 3026
Attributes: 12
Sum of weights: 3026

Name: fixed acidity
Type: Numeric
Missing: 0 (0%)
Distinct: 60
Unique: 10 (0%)

Statistic	Value
Minimum	4.2
Maximum	11.8
Mean	6.882
StdDev	0.789

No.		Name
1	□	fixed acidity
2	□	volatile acidity
3	□	citric acid
4	□	residual sugar
5	□	chlorides
6	□	free sulfur dioxide
7	□	total sulfur dioxide

Class: quality (Nom)

OK

The attributes are listed in the bottom left, and summary statistics for the currently selected attribute are shown on the right side, along with a histogram. Click on each attribute (or use the down arrow key to move through them) and look at the corresponding histogram. You will notice that many numeric attributes have a "hump" shape; this is a common pattern for numeric attributes drawn from real-world data.

You will also notice that some attributes appear to have outliers on one or both sides of the distribution. The proper treatment of outliers varies from one experiment to another. For this assignment you can leave the outliers alone.

Now answer the question below:

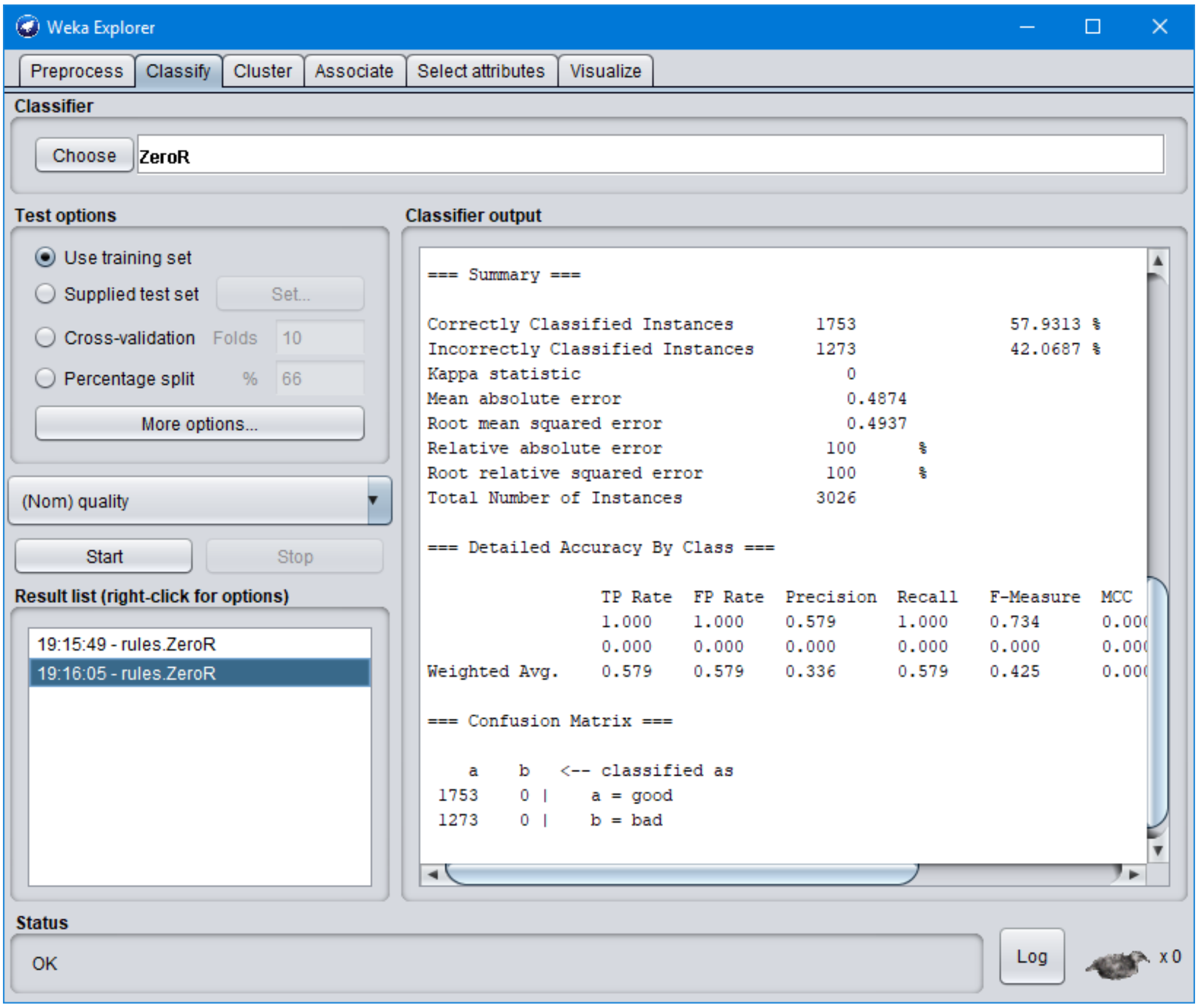
> Question #1: [10 marks]
>
> Which attributes in the training set do not appear to have a "hump" distribution? Which attributes appear to have outliers? (Do not worry too much about being precise here. The point is for you to inspect the data and interpret what you see.)
>
> Based on the histogram, which attribute appears to be the most useful for classifying wine, and why?

Task 1 – Part 2: Classifier Basics [30 marks]

In this section you will train a couple of basic classifiers on the data.

Baseline Classifier

Click on the "Classify" tab. Choose *ZeroR* as the Classifier if it is not already chosen (it is under the "rules" subtree when you click on the "Choose" button). When used in a classification problem, *ZeroR* simply chooses the majority class. Under "Test options" select "Use training set", then click the "Start" button to run the classifier. You should see something like this:



Classifier

ZeroR

Test options

● Use training set

○ Supplied test set

○ Cross-validation Folds 10

○ Percentage split % 66

(Nom) quality

Result list (right-click for options)

19:15:49 - rules.ZeroR

19:16:05 - rules.ZeroR

Classifier output

```
=== Summary ===

Correctly Classified Instances        1753               57.9313 %
Incorrectly Classified Instances      1273               42.0687 %
Kappa statistic                          0
Mean absolute error                      0.4874
Root mean squared error                  0.4937
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             3026

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 1.000    1.000    0.579      1.000    0.734      0.00
                 0.000    0.000    0.000      0.000    0.000      0.00
Weighted Avg.    0.579    0.579    0.336      0.579    0.425      0.00

=== Confusion Matrix ===

    a    b   <-- classified as
 1753    0 |    a = good
 1273    0 |    b = bad
```

Status

OK

The classifier output pane displays information about the model created by the classifier as well as the evaluated performance of the model. In the Summary section, the row "Correctly Classified Instances" reports the accuracy of the model.

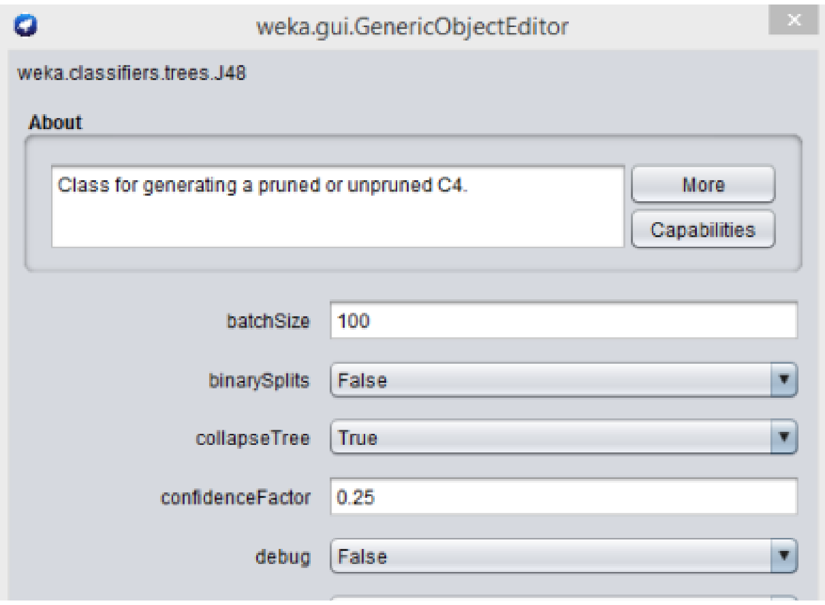
Now answer the question below:

Question #2: [10 marks]

What is the *accuracy* - the percentage of correctly classified instances - achieved by *ZeroR* when you run it on the training set? Explain this number (what this number means …). How is the accuracy of *ZeroR* a helpful baseline for interpreting the performance of other classifiers?

Decision Trees

J48 is the Weka implementation of the C4.5 decision tree algorithm.
Click on the "Choose" button and select *J48* under the "trees" section. Notice that the field to the right of the "Choose" button updates to say "J48 -C 0.25 -M 2". This is a command-line representation of the current settings of *J48*. Click on this field to open up the configuration dialog for *J48*:



weka.gui.GenericObjectEditor

weka.classifiers.trees.J48

About

Class for generating a pruned or unpruned C4.

batchSize: 100

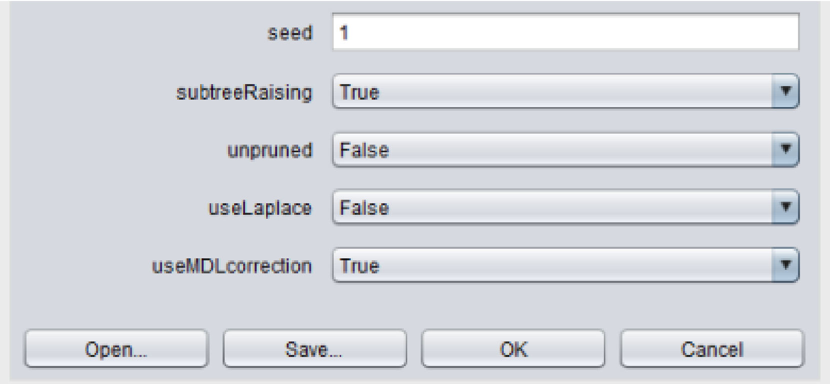
binarySplits: False

collapseTree: True

confidenceFactor: 0.25

debug: False

. . . .



seed: 1

subtreeRaising: True

unpruned: False

useLaplace: False

useMDLcorrection: True

Each classifier has a configuration dialog such as this that shows the parameters of the algorithm as well as buttons at the top for more information. When you change the settings and close the dialog, the command line representation updates accordingly. For now we will use the default settings, so hit "Cancel" to close the dialog.

Under "Test options" select "Use training set", then click the "Start" button to run the classifier. After the classifier finishes, scroll up in the output pane. You should see a textual representation of the generated decision tree.

Now answer the question below:

Question #3: [10 marks]

Using a decision tree Weka learned over the training set, what is the most informative single feature for this task, and what is its influence on wine quality? Does this match your answer from Question #1?

Scroll back down and record the percentage of Correctly Classified Instances. Now, under "Test options", select "Cross-validation" with 10 folds. Run the classifier again and record the percentage of Correctly Classified Instances.

In both cases, the final model that is generated is based on all of the training data. The difference is in how the accuracy of that model is estimated.

Now answer the question below:

Question #4: [10 marks]

What is 10-fold cross-validation? What is the *main* reason for the difference between the percentage of Correctly Classified Instances when you used the entire training set directly versus when you ran 10-fold cross-validation on the training set? Why is cross-validation important?

Task 1 – Part 3: Build Your Own Classifier [40 marks]

This is the main part of the assignment. Search through the classifiers in Weka and run some of them on the training set. You may want to try varying some of the classifier parameters as well. **Choose the one** you feel is most likely to generalize well to unseen examples. Feel free to use validation strategies other than 10-fold cross-validation.

When you have built the classifier you want to submit, move on to the following sections.

Saving the Model

To export a classifier model you have built:

1. Right-click on the model in the "Result list" in the bottom left corner of the Classify tab.
2. Select "Save model".
3. In the dialog that opens, ensure that the File Format is "Model object files"
4. Save the model using the naming convention given as instructed (e.g. A1-Darth-Vader.model).

Now answer the question below (Questions #5 and #6):

Question #5: [5 marks]

What is the "command-line" for the model you are submitting? For example, *"J48 -C 0.25 -M 2"*. What is the reported accuracy for your model using 10-fold cross-validation?

Do not submit the example model for your answer.

Question #6: [20 marks]

In a few sentences, describe how you chose the model you are submitting. Be sure to mention your validation strategy and whether you tried varying any of the model parameters.

Building various versions of your selected model
With the classifier model you selected, try to vary the model by setting various configurations. You can do this by trying different values for parameters. For example, if you chose the J48 decision tree classifier, your initial model might be built using default configuration (parameter setting). To validate your model, you can try to set alternative option “True” for “unpruned” parameter. The result of running this model with the same train/test data set may be different. Try yourself to configure variously parameters and save them as different versions of your classifier model.

Now answer the question below:

Question #7: [15 marks]

For your selected classifier model, investigate what kind of parameters are available to be set in Weka and briefly summarise of the role of each parameter.

Throughout the various configuration testing, summarise your findings about the effect (on the result of testing the model using test data) of three different parameters of your choice.

Marking Rubric

	Exemplary	Good	Satisfactory	Limited	Very Limited
	9-10	7-8	5-6	3-4	0-2
For each question, Questions 1-9	Answer demonstrates excellent knowledge of machine learning and data science, is well-written, and very well-justified.	Exhibits aspects of exemplary (left) and satisfactory (right)	Answer demonstrates sound knowledge of machine learning and data science and provides justification.	Exhibits aspects of satisfactory (left) and very limited (right)	Answer demonstrates flawed knowledge of machine learning and/or provides incoherent justification. Or Answer is absent or negligible.