Laboratory Solution Week 2- 2018 S2

1. You will need to have access to the WEKA package, see week 1 lab sheet.

2. The data files for this lab can be found at /KDrive/SEH/SCSIT/Students/Courses/COSC2111/DataMining/data

3. Start Weka and load the file: ../data/arff/UCI/iris.arff. Run the IBK classifier with 66% split.

(a) From the output, find the number of examples in the training and test sets?

After importing ”iris.arff” data set into weka, the splitting the dataset into training and testing sets by setting “Percentage split” box into %66 under Test Options in the up left hand side of classifier tab, you would get the following output as shown in fig 1.

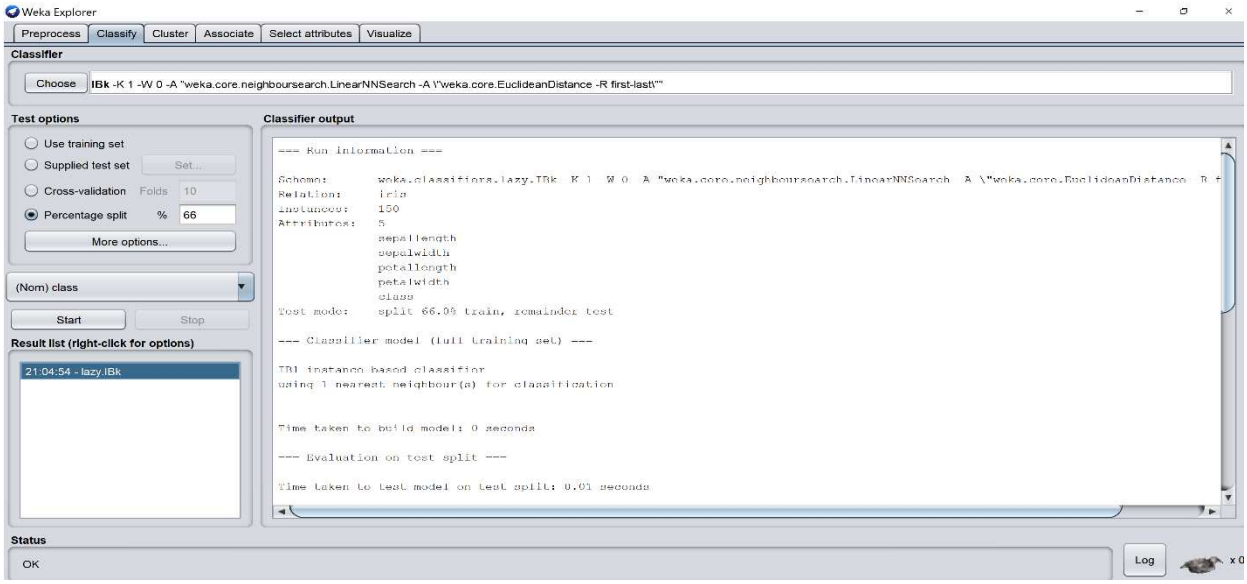

Figure 1. Output of running IBK classifier on iris data set where data splitting is chosen by percentage split option.

As it can be seen, the total number of instances in this data set is 150 with 5 attributes of sepal length, sepal width, petal length, petal width and the last attribute that is addressed class (i.e., target value). Since percentage split is set on %66 so %66 of 150 instances is 99 instances. That means training set contains 99 instances and looking the figure 2, 51 instances are left as testing set that total of them ends up to 150 (i.e., 51 (instances in test set) + (150 – 51= 99) instances in the training set).

(b) What is the test error rate?

As it can be seen in figure 2, the error test means number of misclassified instances. Therefore %3.92 is the test error rate or it can be said that 2 instances out of 51 instances from testing set have been incorrectly classified.

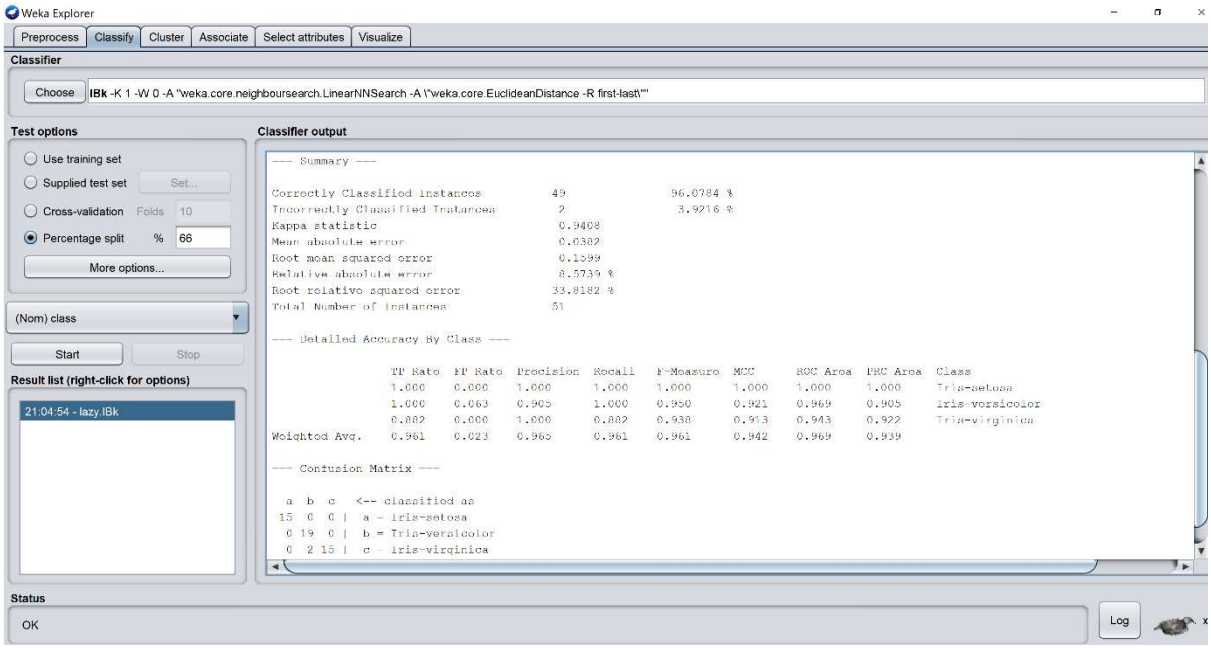

Fig 2. The remaining output from fig 1.

4. Now load the file ../data/lab01-iris-train.arff

(a) Select the IBK classifier. From test options select 'Supplied test set' and for the test file use ../ data/arff/lab01-iris-test.arff. (b) You will see that the training file has 100 examples and the test file has 50 examples. (c) What is the error rate of the IB1 classifier? (d) Explain the difference in error rates. You may want to inspect the files with an editor.

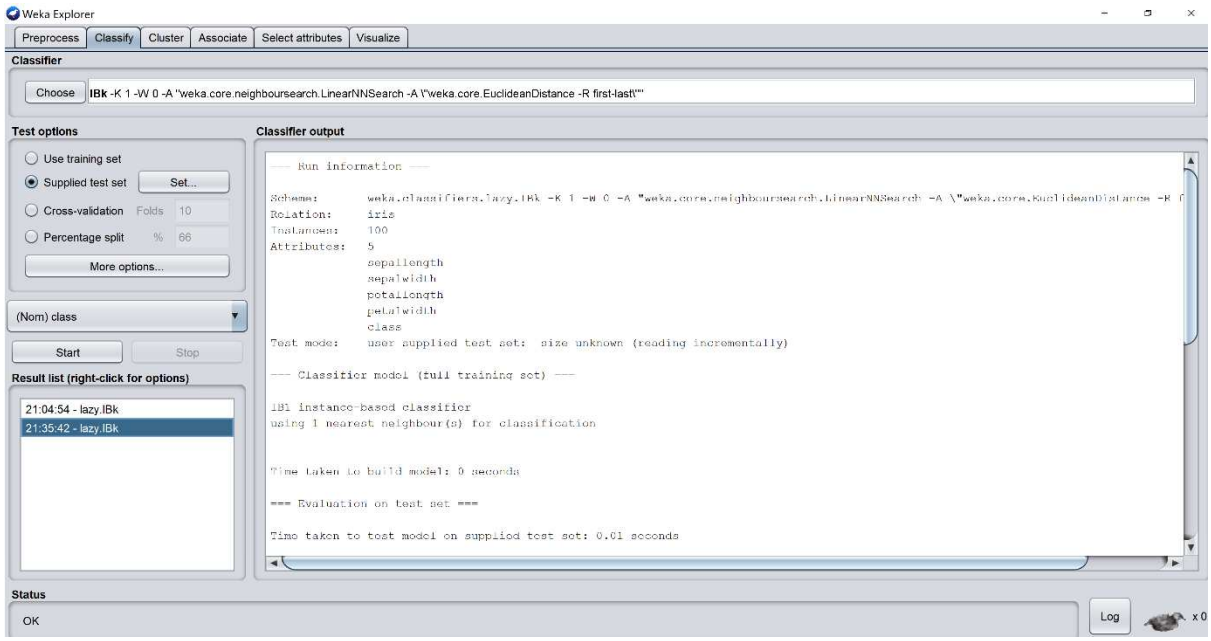

Fig 3. Run of IBK classifier where iris data set has been manually split into two training and testing sets.

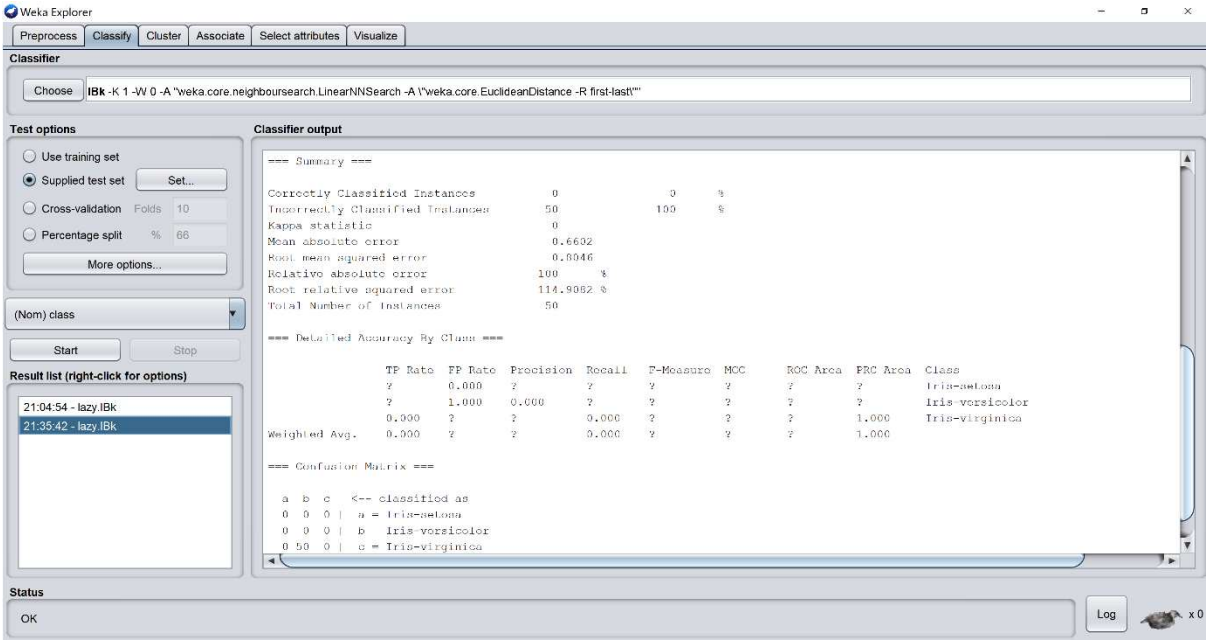

Fig 4. Remaining of the weka output by running IBK classifier on lab01-iris-train.arff and test from fig3.

As you can see from figure 3 and 4 which they are output from run of IBK classifier on Supplied test set option, the total number of training instances from figure 3 is 100 and the total number of testing instances is 50 from figure 4. The error rate is shown as 50 or % 100. That means the IBK classifier has not learnt anything on this data set. In order to explore and investigate the reason, one approach is to open both “lab01-iris-train.arff” and “ lab01-iris-test.arff” by any editor like Notepad++. As you noticed iris data set has 3 classes of virginica, versicolor and setosa. By opening up these two data sets, you would notice that “lab01-iris-test.arff” only contains virginica class while “lab01-iris-train.arff” only contains two other classes of setosa and versicolor. That means your model has not been trained on the virginica class and that s why the accuracy of this model is zero.

5. Repeat the above exercise with OneR. Is there any difference?

Yes. You would get the same result because the trained model is still biased on just two classes of setosa and versicolor.

6. Load the file ../data/arff/UCI/splice.arff into Weka.

7. Select the ZeroR classifier from Rules. Right click on “ZeroR” and follow the help windows to determine what this classifier does.

8. Run the classifier, what is the accuracy?

Correctly Classified Instances 557 51.3364 %

9. Run the IBK classifier with default parameters. What is the accuracy?

Correctly Classified Instances 807 74.3779 %

10. Run the classifier for increasing values of K. Does there seem to be an optimal value for K?. [You might want to plot Accuracy vs K.]

Table1. Accuracy of different runs of IBK classifier on Splice data set by increasing number of k.

IBK Run	K	%Accuracy (Correctly Classified Instances)
1	1	74.3779 %
2	2	71.7051 %
3	3	78.0645 %
4	4	77.8802 %
5	5	79.2627 %
6	6	80.1843 %
7	7	81.0138 %
8	15	86.1751 %
9	30	86.9124 %
10	50	88.2028 %
11	100	89.8618 %
12	200	91.6129 %
13	500	91.4286 %
14	1000	61.0138 %
15	1500	51.3364 %

As it can be seen from table 1, increasing number of K boosts up the accuracy of classifier on splice data set. According to the above table, the higher k, the more accuracy. However, plotting table 1, it would be noticed that there is a spike (k=500) where the accuracy has been dropped and by increasing k , the accuracy of IBK will have decreasing trend.

11. IBK has a number of parameters. Explore the effect of changing "distance weighting" and "nearestNeighbourSearchAlgorithm". Summarise the effect of each choice.

12. Now apply the OneR classifier to splice.arff with default parameters.

The accuracy of OneR classifier on splice data set with default parameters (where batchsize=binsize=100) is 25.0691 % which is very low. There is an underfitting situation, the model is too weak and simple. It seems that a model with more attributes in needed. This is why OneR is not a suitable classifier for this data set and we need to use more sophisticated classifier.

13. Explore the effect of changing bin size.

It seems that changing bin size does not have any effect on accuracy of OneR .

14. What do you conclude from the results of ZeroR, IBK and OneR.

ZeroR error is 51.3, Best IBK is 91.48 best OneR is 25.0. There is no single dominant attribute. All, or some combination of all of the attributes is needed.

15. Choose some other files from the ../data/arff/UCI/ directory and investigate the performance of these three classifiers.