COMP307 Report – Cam Olssen

Part 1 (K-Nearest-Neighbour):

a. With k=1, this is the output of predicted versus actual labels for each instance in wine-test.

```
Instance 0:     Label : 3 | Prediction: 3
Instance 1:     Label : 3 | Prediction: 3
Instance 2:     Label : 3 | Prediction: 3
Instance 3:     Label : 1 | Prediction: 1
Instance 4:     Label : 1 | Prediction: 1
Instance 5:     Label : 1 | Prediction: 1
Instance 6:     Label : 2 | Prediction: 1
Instance 7:     Label : 2 | Prediction: 2
Instance 8:     Label : 1 | Prediction: 1
Instance 9:     Label : 2 | Prediction: 2
Instance 10:    Label : 2 | Prediction: 2
Instance 11:    Label : 2 | Prediction: 3
Instance 12:    Label : 3 | Prediction: 3
Instance 13:    Label : 3 | Prediction: 3
Instance 14:    Label : 1 | Prediction: 1
Instance 15:    Label : 2 | Prediction: 2
Instance 16:    Label : 3 | Prediction: 3
Instance 17:    Label : 3 | Prediction: 3
Instance 18:    Label : 1 | Prediction: 1
Instance 19:    Label : 1 | Prediction: 1
Instance 20:    Label : 3 | Prediction: 3
Instance 21:    Label : 2 | Prediction: 2
Instance 22:    Label : 2 | Prediction: 2
Instance 23:    Label : 3 | Prediction: 3
Instance 24:    Label : 2 | Prediction: 2
Instance 25:    Label : 2 | Prediction: 2
Instance 26:    Label : 2 | Prediction: 2
Instance 27:    Label : 3 | Prediction: 3
Instance 28:    Label : 2 | Prediction: 2
Instance 29:    Label : 1 | Prediction: 1
Instance 30:    Label : 2 | Prediction: 2
Instance 31:    Label : 1 | Prediction: 1
Instance 32:    Label : 2 | Prediction: 2
Instance 33:    Label : 1 | Prediction: 1
Instance 34:    Label : 2 | Prediction: 2
Instance 35:    Label : 2 | Prediction: 2
Instance 36:    Label : 2 | Prediction: 2
Instance 37:    Label : 2 | Prediction: 2
Instance 38:    Label : 2 | Prediction: 2
Instance 39:    Label : 1 | Prediction: 1
Instance 40:    Label : 2 | Prediction: 2
Instance 41:    Label : 2 | Prediction: 2
Instance 42:    Label : 3 | Prediction: 3
Instance 43:    Label : 1 | Prediction: 1
Instance 44:    Label : 2 | Prediction: 2
Instance 45:    Label : 1 | Prediction: 1
Instance 46:    Label : 3 | Prediction: 3
Instance 47:    Label : 2 | Prediction: 2
Instance 48:    Label : 2 | Prediction: 2
Instance 49:    Label : 1 | Prediction: 1
Instance 50:    Label : 3 | Prediction: 3
Instance 51:    Label : 1 | Prediction: 1
Instance 52:    Label : 1 | Prediction: 1
Instance 53:    Label : 3 | Prediction: 3
Instance 54:    Label : 3 | Prediction: 3
Instance 55:    Label : 1 | Prediction: 1
Instance 56:    Label : 1 | Prediction: 1
Instance 57:    Label : 3 | Prediction: 3
Instance 58:    Label : 1 | Prediction: 2
Instance 59:    Label : 3 | Prediction: 3
Instance 60:    Label : 3 | Prediction: 3
Instance 61:    Label : 2 | Prediction: 1
Instance 62:    Label : 2 | Prediction: 2
Instance 63:    Label : 3 | Prediction: 3
Instance 64:    Label : 2 | Prediction: 2
Instance 65:    Label : 3 | Prediction: 3
Instance 66:    Label : 3 | Prediction: 3
Instance 67:    Label : 1 | Prediction: 1
Instance 68:    Label : 1 | Prediction: 1
Instance 69:    Label : 2 | Prediction: 2
Instance 70:    Label : 2 | Prediction: 2
Instance 71:    Label : 3 | Prediction: 3
Instance 72:    Label : 2 | Prediction: 2
Instance 73:    Label : 2 | Prediction: 2
Instance 74:    Label : 1 | Prediction: 1
Instance 75:    Label : 1 | Prediction: 1
Instance 76:    Label : 1 | Prediction: 1
Instance 77:    Label : 3 | Prediction: 3
Instance 78:    Label : 1 | Prediction: 1
Instance 79:    Label : 1 | Prediction: 1
Instance 80:    Label : 2 | Prediction: 2
Instance 81:    Label : 2 | Prediction: 2
Instance 82:    Label : 3 | Prediction: 3
Instance 83:    Label : 1 | Prediction: 1
Instance 84:    Label : 2 | Prediction: 2
Instance 85:    Label : 1 | Prediction: 1
Instance 86:    Label : 1 | Prediction: 1
Instance 87:    Label : 2 | Prediction: 2
Instance 88:    Label : 1 | Prediction: 1
```

The algorithm with k=1 has a prediction accuracy rate of 95.505618 percent, or 85 out of 89 correct guesses.

b. With k=3, the accuracy rate is 95.505618% again, the same as k=1.

c. One of the largest advantages of KNN is the lack of a training period, as the data itself is a model for future prediction, making it very time-efficient. This also means that new data can be added at any time. It is also easy to implement, as the only thing that needs to be calculated is the distance between points. However, KNN does not work well with a large dataset, as it is a slow algorithm and calculating the distance between high numbers of points quickly becomes inefficient.

d. The steps to implement k-fold cross validation where k=5 are:

1. Divide the data into 5 sets (folds)

2. Pick one fold to use as the test set, and use the other four as training sets.

3. Repeat this for each fold, giving each one a turn to be the test set.

4. Take the output from each of our five folds and average it for our estimated accuracy.

Part 2 (Decision Tree):

a. The decision tree algorithm correctly classified 19 out of 25 instances, for an accuracy rate of 76%. The most common category in both files is “live”, which occurs 20 out of 25 times in the test dataset. As such, the baseline predictor is slightly more accurate, with an accuracy rating of 80%. However, with a more even ratio of live to die, the decision tree algorithm prove more accurate than the baseline.
b. i. Find ‘twig’ nodes – nodes who only have leaves as children. Based on which of them have the lowest information gain, remove their children and turn the twig node into a leaf. Continue until the tree is at desired size.
   ii. Pruning lowers the accuracy of the training set, as it will not learn the optimal parameters as well for the training set.
   iii. No – pruning reduces accuracy on the training set as it does not learn the optimal parameters as well, but it also prevents overfitting of the test set and allows the algorithm to adapt better to unknown data.

Part 3 (Perceptron):

1. When testing and training on the same data, the perceptron had an accuracy of 94%. When running on the split dataset, it had an accuracy of 96% for the training data and 85% for the test data. It found a correct set of weights, and the accuracy remained consistent between runs.
2. Evaluating the perceptron on its performance for the training data does not tell us much about the algorithm’s effectiveness as that is the data that it learns how to predict from. In order to gauge its effectiveness we need to examine its performance on data that it does not already know – the testing data.