COMP 307 Assignment 1 Report

PART ONE

The results below were collected classifying the data in the test sets using K-nearest Neighbour. This is a classification algorithm in which an object is assigned to the class most common among it nearest *k* neighbours, with *k* > 0.

Advantages and disadvantages of k nearest neighbour:

Advantages
- Cost of the learning process of k nearest neighbour is 0.
- It is a relatively simple algorithm to implement (compared to other AI classification techniques)
- It is effective on large training data-sets
- It is robust to noisy training data

Disadvantages
- If the training set is very large, though it will still be effective in its classifications, its run time performance will be poor due to having to calculate the distance value for each training instance.
- Performance also depends on the number of features an instance has. Feature selection needs to be done carefully, as the more features used, the more dimensions there are, which means a large training dataset must be used to ensure the nearest neighbours are actually close.
- K nearest neighbour calculates the distance between instances for classification. There are numerous ways to calculate distances, for example weighted distance, inverse square... it can be difficult to know which to use.

Results:

Classifications highlighted in yellow are k =1 incorrect classifications, pink are k = 3 incorrect classifications.

Classname	kNN Class k=1	kNN Class k=3	Classname	kNN Class k=1	kNN Class k=3	Classname	kNN Class k=1	kNN Class k=3
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica	Iris-versicolor	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica	Iris-versicolor	Iris-versicolor
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-versicolor	Iris-versicolor
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-versicolor	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-virginica	Iris-versicolor	Iris-virginica

Tested 75 instances with k=1 and 68.0 were classified correctly.
Accuracy: 90.67%
Tested 75 instances with k=3 and 71.0 were classified correctly.
Accuracy: 94.67%

The results show that increasing k from 1 to 3 yields higher accuracy. This is most likely because when k = 1 there is a danger of over-fitting the data to the training dataset, instead of giving a more general classification which increasing k to 3 allowed, and therefore more instances were correctly classified. It is intuitive then to believe increasing k to a large number would give more accurate reading, but this also has its own issues, mainly under-fitting the data so that it becomes to generalised.

COMP 307 Assignment 1 Report

Kandice McLean
300138073

K-fold Cross-Validation

The steps that would need to be taken in order to classify the instances provided using k-fold cross validation with k = 5 are (assuming we are only using the testing set, not both of them, although the same steps would still apply for both using 150 instances instead of 75):

1. Divide the 75 instances into 5 equal sets of 15
2. For the first set of 15:

    I. Treat it as the test set
    II. Treat the remaining 60 instances as the training set
    III. Use the training set to train the classifier, and apply the test set
    IV. Record the results

3. Repeat the process giving each of the 5 sets a turn at being the test set
4. Average and/or combine the results to produce a single estimation.

K-means Clustering

This method is what should be used if the class labels of the instances are not available for the training set and test set, but there are three clusters.
The steps to be taken to classify using this method are:

1. Set k official means, in this case k =3 since there are three clusters, randomly from the data-set
2. Create k clusters by associating every instance with the nearest mean based on a distance measure
3. Replace the old means with the centroid of each of the k clusters
4. Repeat steps 2 and 3 until convergence occurs.

PART TWO

```
ASCITES = True:
  SPIDERS = True:
    VARICES = True:
      FIRMLIVER = True:
        Class live, prob=1.00
      FIRMLIVER = False:
          BIGLIVER = True:
            STEROID = True:
              Class live, prob=1.00
            STEROID = False:
                FEMALE = True:
                  Class live, prob=1.00
                FEMALE = False:
                    ANTIVIRALS = True:
                      FATIGUE = True:
                        Class die, prob=1.00
                      FATIGUE = False:
                          Class live, prob=1.00
                    ANTIVIRALS = False:
                        Class die, prob=1.00
          BIGLIVER = False:
              Class live, prob=1.00
    VARICES = False:
        Class die, prob=1.00
  SPIDERS = False:
      FIRMLIVER = True:
```

COMP 307 Assignment 1 Report

Kandice McLean
300138073

```
AGE = True:
  Class live, prob=1.00
AGE = False:
  SGOT = True:
    Class live, prob=1.00
  SGOT = False:
    ANTIVIRALS = True:
      Class die, prob=1.00
    ANTIVIRALS = False:
      STEROID = True:
        Class live, prob=1.00
      STEROID = False:
        Class die, prob=1.00
FIRMLIVER = False:
  SGOT = True:
    BIGLIVER = True:
      SPLEENPALPABLE = True:
        Class live, prob=1.00
      SPLEENPALPABLE = False:
        ANOREXIA = True:
          Class die, prob=1.00
        ANOREXIA = False:
          Class live, prob=1.00
    BIGLIVER = False:
      Class die, prob=1.00
  SGOT = False:
    Class live, prob=1.00
ASCITES = False:
  BIGLIVER = True:
    STEROID = True:
      Class die, prob=1.00
    STEROID = False:
      ANOREXIA = True:
        Class die, prob=1.00
      ANOREXIA = False:
        Class live, prob=1.00
  BIGLIVER = False:
    Class live, prob=1.00
```

Total tests run = 27
Total correct classifications using TREE = 21.0
Percentage accuracy = 77.78%
Total correct classifications using Baseline = 23.0
Percentage accuracy = 85.19%

The above tree and its associated results was populated using the decision tree algorithm.
In this particular experiment, the baseline accuracy, which is calculated by placing results in the highest occurring class, was higher than the accuracy of the tree. However, this does not mean that the baseline classification method is better than using a decision tree.
The baseline classifier accuracy correlates to how the classes are spread over the given dataset. For example, if the dataset is exactly half of class 1 and half of class 2, then the accuracy of the baseline classifier is going to be 50%, and if there is only 1% class 1 and 99% class 2, then the baseline accuracy will be 99%. Therefore although the baseline classifier is more accurate in this particular experiment, it is not consistent enough to be used as an accurate classification method.
In general, the tree should be more accurate over random data sets than the baseline, theoretically achieving approximately the same accuracy on each data set.
To test this theory, another experiment is run over 11 different sets of data, with the results below:

Kandice McLean
300138073

Test Number:	Tree Accuracy %	Baseline Accuracy %
1	83.78	91.89
2	83.78	89.19
3	81.08	86.49
4	75.68	78.38
5	75.68	89.19
6	67.57	78.38
7	83.79	89.19
8	67.57	86.49
9	75.68	83.78
10	78.38	81.08
11	82.61	36.96
Average	77.7818181818	81.0018181818

Although baseline did quite well in all the tests, you can see in test 11 its accuracy was extremely low. This is because Test 11 was purposely made to be around half of class 1 and class 2 to show the effects that this has. Because the data selection is usually random, there is no guarantee of this not happening, so baseline classification is not a good consistent method of classification. The tree has a much lower range of accuracy, and is unaffected by the spread of classes over instances.

An issue with decision trees is that the cost of the algorithm is quite high on larger dataset. Pruning the leaves can reduce the size of the tree, meaning computing the class costs can be reduced. However there is an issue with pruning trees as it can result in a loss of accuracy.
One of the simplest forms of pruning a tree is reduced error pruning. Starting at the leaves, each node is replaced by its most popular class. If accuracy is not affected then the changes are retained, else the nodes are changed back to their previous state. While this is somewhat basic, it has the advantage of being simple and fast to implement.
The loss of accuracy from pruning the tree is related to the idea of under/over fitting data. Because the tree was originally generated from the data set, any changes that are made without the training sets involvement can result in it under fitting the data.
However, pruning does sometimes have the ability to *increase* accuracy as well. Because the training set in no way represents all of the data as a whole, and is usually selected at random, then there is a possibility that the training data set could be biased. By pruning a biased decision tree it is possible to increase the fit of the overall data set.

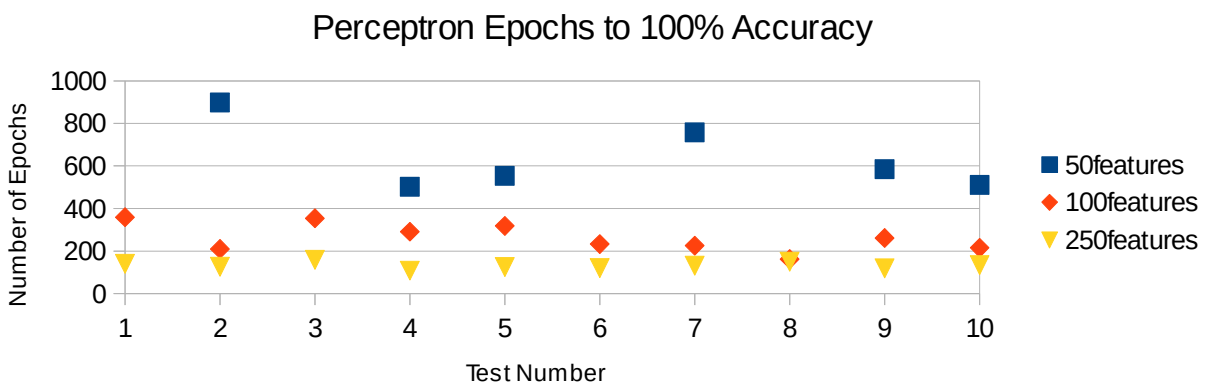
The decision tree depicted in the experiments uses an impurity measure to build the tree. This is possible only because the dataset is binary. If there are three or more possible classes that the decision tree must distinguish then impurity measure is not an appropriate algorithm. This is because there are three criteria of a good impurity generation function:

1. Should be 0 when either class contains all instances
2. Should be maximum if they have equal number of instances
3. Function should be smooth

These three criteria cannot be easily met when there are more than two classes. For example, if there were three classes in the instances used above, and the tree was getting impurity, consider the case if class 1 has 0 instances, class 2 has 3 and class 3 has 4. The impurity would return 0, because class 1 contains 0 instances so must be pure. However, this would not be the case, as both class 2 and 3 contain >0 instances that still need to be weighted.

PART THREE

In this experiment the perceptron was considered successful if it found a set of weights that classified 100% of the images in the image file, at which point the program would print out the successful weights and terminate.
The learning rate was set as a constant 0.2, and maximum epochs was set to 1000. If the number of epochs needed to gain 100% accuracy was more than this, it was considered a failure.
When testing the perceptron, it was found that increasing the number of features reduces the number of epochs required to achieve success. Below are the results of testing this perceptron using 50 features, 100 features and 250 features. Unsuccessful tests for a feature size are not depicted.



However, these tests are not particularly helpful because of the possibility of over fitting. The weights that work with this training set will not necessarily work with all data sets. This is because the perceptron has been trained using this data, and the weights are specifically tailored to this data sets needs. Another data set may need completely different weights, but because we have trained the perceptron to get 100% accuracy on this data set alone, it may not have generalised the weights enough to be successful in classifying other datasets.