Report

Part 1

1) AI Prediction for k-Nearest Neighbours when k = 1 in order of appearance, the first 25 entries expect “Iris-setosa”, the next 25 expect “Iris-versicolor” and the last 25 expect “Iris-virginica”:

Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-versicolor
Iris-versicolor
Iris-virginica Expected: “Iris-versicolor”
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica Expected: “Iris-versicolor”
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor

Nathan Dunning

Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”
Iris-versicolor Expected: “Iris-virginica”
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”

Accuracy: 68/75 = 90.67%

Nathan Dunning

2) AI Prediction for k-Nearest Neighbours when k = 3 in order of appearance, the first 25 entries expect “Iris-setosa”, the next 25 expect “Iris-versicolor” and the last 25 expect “Iris-virginica”:

Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica Expected: “Iris-versicolor”
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor

Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”
Iris-versicolor Expected: “Iris-virginica”
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica

Accuracy = 72/75 =96%

As seen from the two outputs of the AI’s predictions with a ‘k’ value of 1 and a ‘k’ value of 3, it is evident that having a ‘k’ value of 3 provides better accuracy for the predictions. From this example it provided about a 5.3% increase in the accuracy of the AI’s prediction.

3) Some advantages of using the k-Nearest Neighbour search technique is that it is very robust to noisy data and anomalies won't have big weights on the final prediction. This technique is good with large training datasets as more neighbours reduce uncertainties of the final prediction. Some disadvantages of this search technique includes determining a value for 'k', this causes complications as it may be hard to determine which 'k' will provide the most accurate result for the dataset. It is also very expensive to run this algorithm as we require iterating through all nodes in our training set for each node in our test set.

4) In order to apply the k-fold cross validation technique for this problem we would first split up our dataset into 'k' equal subsets, we would then assign the first subset to be our test set and use the other k-1 subsets as our training set. We would then iterate through and our second subset would then become our training set and the other remaining k-1 subsets to be our test set. After we completed k iterations we can average our results/accuracy to get our true accuracy for the AI.

5) I would use the k-means clustering method for the problem if there were no class labels. Firstly, since we know there are 3 types of flowers we are trying to identify (domain knowledge) we are going to set k to 3. We would set the values of 'k' randomly from the dataset. I would then iterate through all instances in the dataset and find the euclidean distance to each of the different k's and assign it to the cluster with the smallest distance. After iterating through the dataset and all instances are assigned to a cluster I would then replace the centroid to the centre of the cluster by finding the mean of all values assigned to that cluster. Repeat until the centroid is stable (convergence)

Part 2

1) Running the decision tree using ‘hepatitis-training.dat’ and ‘hepatitis-test.dat’ the overall classification accuracy was 21/27 (77.7%). Comparing the accuracy of the decision tree to the baseline classifier taken from the most frequent class in the dataset, we can see that there is a difference as the test set contains 23/27 ‘live’ and 4/27 ‘die’. The decision tree classifier that was created looked like:

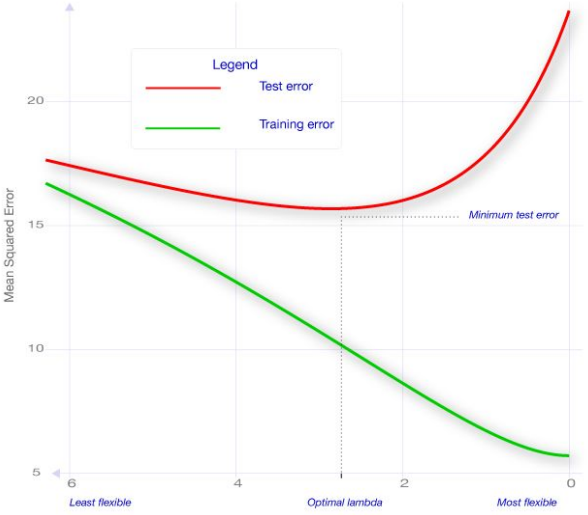
```
ASCITES = True:
  SPIDERS = True:
    VARICES = True:
      FIRMLIVER = True:
        Category live, prob = 100% : /49
      FIRMLIVER = False:
        BIGLIVER = True:
          STEROID = True:
            Category live, prob = 100% : /5
          STEROID = False:
            FEMALE = True:
              Category live, prob = 100% : /2
            FEMALE = False:
              ANTIVIRALS = True:
                FATIGUE = True:
                  Category die, prob = 100% : /1
                FATIGUE = False:
                  Category live, prob = 100% : /4
              ANTIVIRALS = False:
                Category die, prob = 100% : /1
        BIGLIVER = False:
          Category live, prob = 100% : /7
    VARICES = False:
      Category die, prob = 100% : /1
  SPIDERS = False:
    FIRMLIVER = True:
      AGE = True:
        Category live, prob = 100% : /1
      AGE = False:
        SGOT = True:
          Category live, prob = 100% : /1
        SGOT = False:
          ANTIVIRALS = True:
            Category die, prob = 100% : /4
          ANTIVIRALS = False:
            STEROID = True:
              Category live, prob = 100% : /1
            STEROID = False:
              Category die, prob = 100% : /1
    FIRMLIVER = False:
      SGOT = True:
        BIGLIVER = True:
          SPLEENPALPABLE = True:
            Category live, prob = 100% : /4
          SPLEENPALPABLE = False:
            ANOREXIA = True:
```

```
                    Category die, prob = 100% : /2
                ANOREXIA = False:
                  Category live, prob = 100% : /1
            BIGLIVER = False:
              Category die, prob = 100% : /3
          SGOT = False:
            Category live, prob = 100% : /10
ASCITES = False:
  BIGLIVER = True:
    STEROID = True:
      Category die, prob = 100% : /7
    STEROID = False:
      ANOREXIA = True:
        Category die, prob = 100% : /2
      ANOREXIA = False:
        Category live, prob = 100% : /2
  BIGLIVER = False:
    Category live, prob = 100% : /1
```

2) Running the file in splits and recording different trials
   a) 31/37
   b) 31/37
   c) 30/37
   d) 28/37
   e) 28/37
   f) 25/37
   g) 31/37
   h) 25/37
   i) 28/37
   j) 29/37

   The after a total of 10 runs the average accuracy calculated was 286/370 (77.2%)

3) You would prune leaves from a decision tree by setting a classification accuracy threshold (eg. 95%) and if a node has a classification accuracy of 95% or greater (19 of 20 nodes are true) you would not need to split the node an extra time and just return true - in this case, as there are significantly more true's that falses'. This would reduce training accuracy because by doing this you add more randomness to your model, this ensures that we do not overfit our data. By deciding not to split nodes when it does not fully represent the model, we may be looking for the classification to return false but it may return true as the accuracy of that node has reached 95%. Furthermore this may increase accuracy for our test set as we have



not overfit our model to our training set hence it may be able to return correct values that do not fully match the training model. As we can see from the graph we want to stop training the model when the test error is at the minimum, further training past this point will decrease training error but will increase test error as stated above.

4) Impurity is not a good measure when there are 3 or more classifications as a decision tree will only stop when all (or most - if pruning) of the information is in a single class or attribute. Impurity is a count of number of occurrences that are either true or false therefore impurity gives us a number to show how pure a class is. By having it more than 2 classes means that it becomes hard to distinguish how many other classes are in the node. As there is no bias, the variance of this model is much larger than other AI methods hence, overfitting is much more likely to occur. Also by having more classifications, the tree will only grow at an increasing rate.

Part 3

1) The accuracy of my perceptron will vary between 95-98% accuracy for the set number of max iteration. The accuracy of the perceptron does not reach 100% which as a result does find a fully correct set of weights to achieve convergence. When setting the max accuracy to 98% the perceptron will eventually find a correct set of weights however not for 100%.
2) Because we can keep on training our weights for perceptron with the training set until the error is at its absolute minimum but by doing this we are overfitting our model so that it can predict the classifications from the training data accurately however if we test it with unseen data, the error will be much higher as it is not general enough (not enough randomness). No additional data was generated or used to further test.