Part 1

report：

1. The classification accuracy where k = 1 is 90.6666 about 90.76% 2dp

2. The classification accuracy where k = 3 is 96%， there is more accurate than where k =1 90.67%. I thing the reason behind this is because when k=3 it consider more possibility, it looks for the 3 nearest neighbour, and pick up the class is majority.

3 I thing the advantages of K-Nearest Neighbour method is can be increases accuracy, simple to implement, is flexible to feature the value k can be easily to change, can be handles multi-class cases, and also can do well in practice with enough representative data. The disadvantages is have to read all other datas to find the nearest neighbours, and the data have to be Storage. and I thing the algorithm will be run slowly when we have huge data.

4 I will divide the data into 5 equally-sized subsets, because the data have 75 Iris, so I made each subset have the same amount of Iris. Than I will use 4 subset for training and 1 subset for test, repeated 5 times, and made sure every subset have the opportunity to be the test subset. After calculate all the classification accuracy, average them.Then compare with each other (when k = 1,2,3,4,5…) then the highest classification accuracy is the best solution for this data set.

5 I think use the K-means method will solve the problem.
Step1: Set k initial “means” randomly from the data set.
Step2: Create k clusters by associating every instance with the nearest mean based on a distance measure.
Step3: Replace the old means with the centroid of each of the k clusters (as the new means).
Step4: Repeat the above two steps until convergence (no change in each cluster center).

6 did not implement

Part 2

Reading training data from file hepatitis-training.data
2 categories
16 attributesReading test data from file hepatitis-test.data
Read 27 instances
Read 16 attributes
[AGE, FEMALE, STEROID, ANTIVIRALS, FATIGUE, MALAISE, ANOREXIA, BIGLIVER, FIRMLIVER, SPLEENPALPABLE, SPIDERS, ASCITES, VARICES, BILIRUBIN, SGOT, HISTOLOGY]

Desction tree:

```
 ASCITES = True:
  SPIDERS = True:
   VARICES = True:
    FIRMLIVER = True:
     Class live, prob=100.00 count: 49
    FIRMLIVER = False:
     BIGLIVER = True:
      STEROID = True:
       Class live, prob=100.00 count: 5
      STEROID = False:
       FEMALE = True:
        Class live, prob=100.00 count: 2
       FEMALE = False:
        ANTIVIRALS = True:
         FATIGUE = True:
          Class die, prob=100.00 count: 1
         FATIGUE = False:
```

```
          Class live, prob=100.00 count: 4
        ANTIVIRALS = False:
         Class die, prob=100.00 count: 1
      BIGLIVER = False:
       Class live, prob=100.00 count: 7
    VARICES = False:
     Class die, prob=100.00 count: 1
   SPIDERS = False:
    MALAISE = True:
     Class live, prob=100.00 count: 4
    MALAISE = False:
     BILIRUBIN = True:
      AGE = True:
       Class live, prob=100.00 count: 2
      AGE = False:
       ANOREXIA = True:
        SPLEENPALPABLE = True:
         SGOT = True:
          HISTOLOGY = True:
           Class die, prob=50.00 count: 2
          HISTOLOGY = False:
           Class live
         SGOT = False:
          Class live
        SPLEENPALPABLE = False:
         Class live
       ANOREXIA = False:
        Class live, prob=100.00 count: 1
     BILIRUBIN = False:
      Class live, prob=52.63 count: 19
  ASCITES = False:
   Class die, prob=75.00 count: 12

Test VS Algorithm output :
live live
live live
live live
live live
live live
die live
live live
live live
live live
live live
die die
live live
die die
live live
die die
live live
live live
live live
live live
live live
live live
live live
live live
live die
live live
live live
```

live die
Accurate rate: 88.88888888888889%

2
When I run the process for 10 times the average Accurate rate are : 78.38%.
The outputs are all in the part2 file.

3
After a tree has been built (and in the absence of early stopping discussed below) it may be overfitted.The CART algorithm will repeatedly partition data into smaller and smaller subsets until those final subsets are homogeneous in terms of the outcome variable. In practice this often means that the final subsets (known as the leaves of the tree) each consist of only one or a few data points. The tree has learned the data exactly, but a new data point that differs very slightly might not be predicted well.

To prune leaves from the decision tree：

Minimum error. The tree is pruned back to the point where the cross-validated error is a minimum. Cross-validation is the process of building a tree with most of the data and then using the remaining part of the data to test the accuracy of the tree.

Smallest tree. The tree is pruned back slightly further than the minimum error. Technically the pruning creates a tree with cross-validation error within 1 standard error of the minimum error. The smaller tree is more intelligible at the cost of a small increase in error.

Therefore the decision tree will back to general, and because some of the nodes（special cases）are removed the training did not overfitting so the accuracy on the training set might decrease. But it might improve accuracy on the test set.

4
Because when there are three or more classes the algorithm that use to calculate the purity is not accurate. If In the Situations that there are only 2 class in the subset, the impurity of this subset will become 0 which is 100% pure. Because when one class do not have any instances used the formula to calculate the output will become 0.which is not what I expected.

Part3
A)
For training 99 times I did find the correct set of weights and the result are print out as follow.

training 99 times
Correct Output
11111111111111111111111111111111111111111111111111000000000000000000000000000000000000000000000000000
Predicted
11111111111111111111111111111111111111111111111111000000000000000000000000000000000000000000000000000
Accurate rate: 1.0
The LIMTE SIZE: 1000
The LEARNINGRATE: 0.15
Correct set of weights:
[-2.3457201190110126, -0.46989488218391595, -3.0730866485218655, 1.222608663742406, 9.442451203071045, -1.3248899761542585, -1.5909712126428996, 4.710350427732461, -0.5175801631717751, -2.625384673662716, -3.896834494251762 8, -0.782706716318116, 1.961641049611341, -4.4816819455798225, -7.085120465995411, 0.1916831087276263, 0.08072113569964062, 3.1554233468098616, -4.389504777751581, -0.8647250522623637, -1.54242818888362, 4.353816561808077, 0.34796538698882273, 0.6330531090249971, 3.868589654951077, -0.8194610679541173, -2.7177772579710675, 3.5522924149637993,

-0.1651502042053157, -1.7044102049545926, -1.5644683033841145, -3.582465309174153, -2.5783449593251966, 0.9379920122210071, 0.1993260363508014, 1.8955926698530123, -0.23861670518989675, -1.1987785454583764, -3.0587552766102433, 4.333482874829407, -0.8675731426580658, 4.744515369454645, -0.5704454328070513, -0.9177714709716777, -1.918879899611543, -7.282761248817795, -2.8431250850757417, -0.039238248327406466, -0.5410976816738863, 2.7354015167657004, -0.35390367199068284]

But the Correct set of weights does changes when I run the program every time, because the weights and features does generate randomly at the begin. Therefore the number of trails to find the correct set are different. Also I find out when the learning-rate is 0.2 is the best. The training times for this to finding the correct set of weights is around 100 to 200.

B)
Because the training might Overfitting ,When fitting a statistical model, using too many parameters This will reduce or destroy model generalisation ability. Which will decreasing the accurate rate when doing the tests. although the Accurate rate are up to 100%. To evaluating the perceptron's performance it must use the test data. Then it will given a good measure of its effectiveness. Also it have to stop in some where during the training to make sure it did not overfitting.

I did create additional data for test my correct set of weights. and I got the Accurate rate: 0.7.

Correct Output 0000001111
Predicted 1000011110
Accurate rate: 0.7
The LIMTE SIZE: 1000
The LEARNINGRATE: 0.15
Correct set of weights:
[-2.3457201190110126, -0.46989488218391595, -3.0730866485218655, 1.222608663742406, 9.442451203071045, -1.3248899761542585, -1.5909712126428996, 4.710350427732461, -0.5175801631717751, -2.625384673662716, -3.8968344942517628, -0.782706716318116, 1.961641049611341, -4.4816819455798225, -7.085120465995411, 0.1916831087276263, 0.08072113569964062, 3.1554233468098616, -4.389504777751581, -0.8647250522623637, -1.54242818888362, 4.353816561808077, 0.34796538698882273, 0.6330531090249971, 3.868589654951077, -0.8194610679541173, -2.7177772579710675, 3.5522924149637993, -0.1651502042053157, -1.7044102049545926, -1.5644683033841145, -3.582465309174153, -2.5783449593251966, 0.9379920122210071, 0.1993260363508014, 1.8955926698530123, -0.23861670518989675, -1.1987785454583764, -3.0587552766102433, 4.333482874829407, -0.8675731426580658, 4.744515369454645, -0.5704454328070513, -0.9177714709716777, -1.918879899611543, -7.282761248817795, -2.8431250850757417, -0.039238248327406466, -0.5410976816738863, 2.7354015167657004, -0.35390367199068284]