Part1

1.

```
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-virginica
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica
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-versicolor
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor
Iris-versicolor
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor

k= 1
the classification accuracy on the test set of the basic
nearest neighbour method is 0.9066666666666666
```

2.

```
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
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
Iris-versicolor
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
k= 3 the classification accuracy on the test set of the basic
nearest neighbour method is 0.96
```

Comment: the accuracy where k =3 is a lot higher than k= 1, since (0.96>0.906). In other words, the performance of classification where k = 3 works better than the one where k=1.

3.

Advantages:

- Simple to implement
- Robust to noisy training data

- Flexible to feature or distance choices
- Naturally handles multi-class cases
- Can do well in practice with enough representative data.
- Effective if the training data is large.

Disadvantages:

- Large search problem to find nearest neighbours
- Must know we have a meaningful distance function
- Computation cost is quite high because we need to compute distance of each query instance to all training samples. Some indexing (e.g. K-D tree) may reduce this computational cost
- Need to determine value of parameter K

4.

When k = 5, using K-fold cross for above question.

Steps:

1. divide all data into 5 different subsets
2. each subset of data would be a test subset once, the rest of other 4 subsets would be used as training dataset.
3. run step2 in 5 times separate learning experiments
4. The final result is that average the 5 test results from those k experiments in step3
5. Finally, we could predict the type of iris flower, according to the final result of each instance

5.

1. Set k initial “means” randomly from the data set.
2. Create k clusters by associating every instance with the nearest mean based on a distance measure.

$$C(x) = \arg\min dist(c_i, x) \qquad i= 1,...k$$

3. Replace the old means with the centroid of each of the k clusters (as the new means).

$$C_i = 1/|S_i| \sum_{Xi \in S_i} X_i$$

4. Repeat the above two steps until convergence (no change in each cluster center).

Part2

```
1 ASCITES = True:
  SPIDERS = True:
   VARICES = True:
    FIRMLIVER = True:
     Class live
    FIRMLIVER = False:
     BIGLIVER = True:
      STEROID = True:
       Class live
      STEROID = False:
       FEMALE = True:
        Class live
       FEMALE = False:
        ANTIVIRALS = True:
         FATIGUE = True:
          Class die
         FATIGUE = False:
          Class live
        ANTIVIRALS = False:
         Class die
     BIGLIVER = False:
      Class live
   VARICES = False:
    Class die
  SPIDERS = False:
   SPLEENPALPABLE = True:
    ANOREXIA = True:
     AGE = True:
      Class live
     AGE = False:
      MALAISE = True:
       SGOT = True:
        Class live
       SGOT = False:
        HISTOLOGY = True:
         Class die
        HISTOLOGY = False:
         BILIRUBIN = True:
          Class live
         BILIRUBIN = False:
          Class live
      MALAISE = False:
       Class die
    ANOREXIA = False:
     Class live
   SPLEENPALPABLE = False:
    Class die
 ASCITES = False:
  Class die
```

```
The accuracy is 0.74
Baseline Classifier is 0.8
```

The accuracy is lower than baseline in this test data file. The reason I think that the classifier I created by using impurity measurement might have overfitting problem, which may performance even worse than dummy classifier in some test files.

2.

run 10-pairs training and test files
"hepatitis-training-run**.dat"
"hepatitis-training-test**.dat"

```
accuracy 1 :0.8108108108108109
accuracy 2 :0.8378378378378378
accuracy 3 :0.918918918918919
accuracy 4 :0.8378378378378378
accuracy 5 :0.8378378378378378
accuracy 6 :0.7297297297297297
accuracy 7 :0.7837837837837838
accuracy 8 :0.6216216216216216
accuracy 9 :0.7837837837837838
accuracy 10 :0.7837837837837838
Average accuracy for 10 files: 0.7945945945945946
```

3.

(a)

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.

(b)

“Pruning” (removing) some of leaves of the decision tree will always make the decision tree less accurate on the training set. This is easily result in overfitting error on training set. In fact, training set would have some unexpected data. That is why it will reduce accuracy on training set.

(c)Because overfitting will remove some unexpected data from training set, which will improve the accuracy of test data.

4.

For example, if there are only two classes, once one class does not exist, the node is pure. This is correct. However, if there are more than two classes, one of classes does not exist, by using impurity measure, the node was pure. In fact, in this case, this node is not pure because there are more than one classes left.

Part3

1. the perceptron classifier performance quite well.
Running about 100+ training cycles with 0.2 as learning rate, which generate a correct set of weights.
Yes, it always can find a correct set of weights with 0.2 learning rate and 1000 Maximum Limits of training cycles.
2.Because the perceptron is generated by only using training set without using set of test data to check it performance after being generated.
I think that we can get a set of the correct set of weights. All set of weights could make the accuracy of training data set reach to 100%, but every set of weights might be different because random features every time. Once we got a set of correct set of weights, we could find the better performance set of weight, which is the one has highest probability in the set of set of weights.