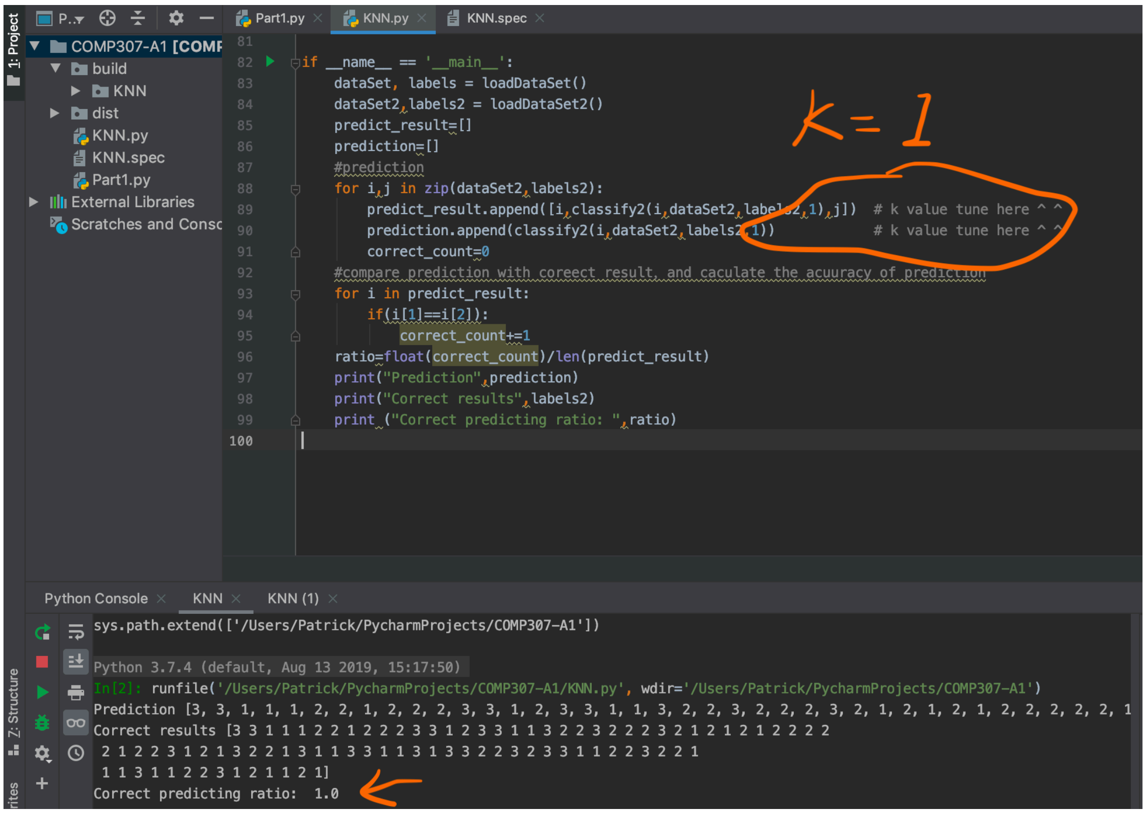
COMP307 Assignment 1

Yan Zichu
300476924

Part 1: Nearest Neighbors Method

Question 1



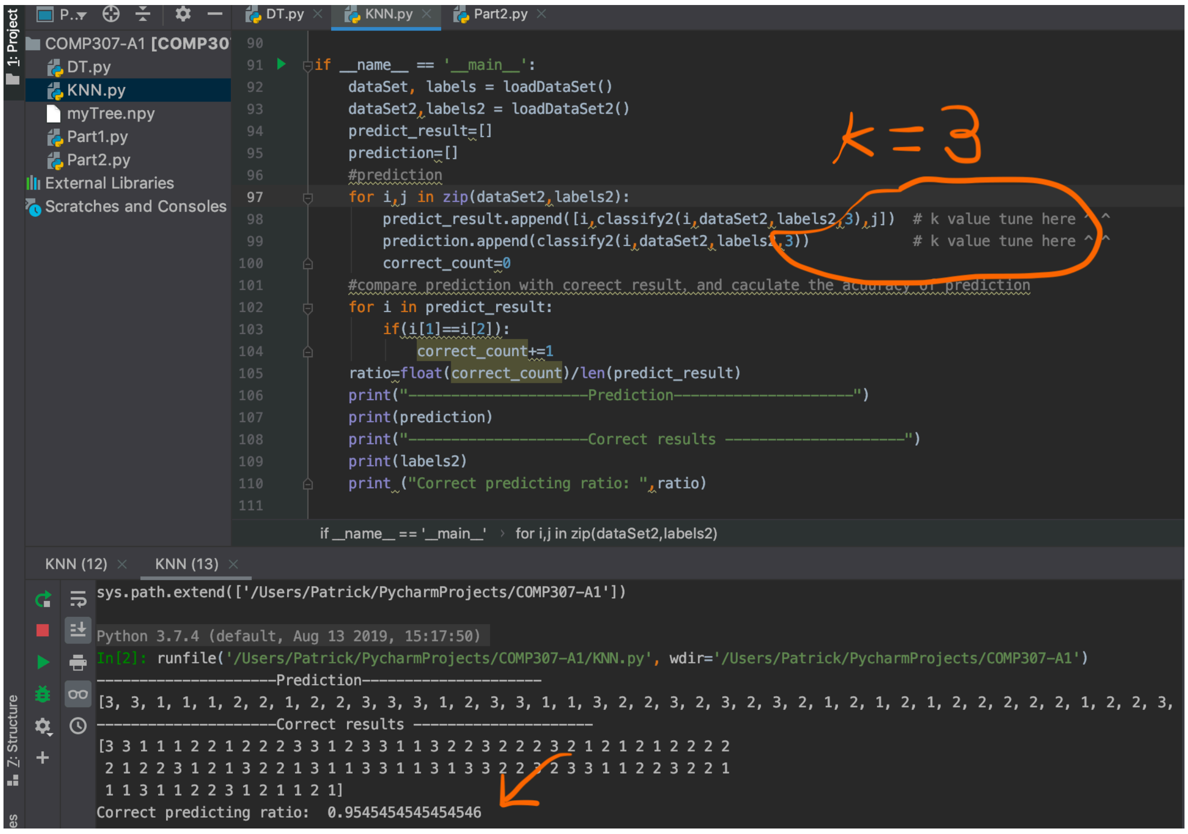
```python
if __name__ == '__main__':
    dataSet, labels = loadDataSet()
    dataSet2,labels2 = loadDataSet2()
    predict_result=[]
    prediction=[]
    #prediction
    for i,j in zip(dataSet2,labels2):
        predict_result.append([i,classify2(i,dataSet2,labels2,1),j])  # k value tune here ^ ^
        prediction.append(classify2(i,dataSet2,labels2,1))           # k value tune here ^ ^
        correct_count=0
    #compare prediction with coreect result, and caculate the acuuracy of prediction
    for i in predict_result:
        if(i[1]==i[2]):
            correct_count+=1
    ratio=float(correct_count)/len(predict_result)
    print("Prediction",prediction)
    print("Correct results",labels2)
    print ("Correct predicting ratio: ",ratio)
```

```
sys.path.extend(['/Users/Patrick/PycharmProjects/COMP307-A1'])

Python 3.7.4 (default, Aug 13 2019, 15:17:50)
In[2]: runfile('/Users/Patrick/PycharmProjects/COMP307-A1/KNN.py', wdir='/Users/Patrick/PycharmProjects/COMP307-A1')
Prediction [3, 3, 1, 1, 1, 2, 2, 1, 2, 2, 2, 3, 3, 1, 2, 3, 3, 1, 1, 3, 2, 2, 3, 2, 2, 2, 3, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1
Correct results [3 3 1 1 1 2 2 1 2 2 2 3 3 1 2 3 3 1 1 3 2 2 3 2 2 2 3 2 1 2 1 2 1 2 2 2 2
 2 1 2 2 3 1 2 1 3 2 2 1 3 1 1 3 3 1 1 3 1 3 3 2 2 3 2 3 3 1 1 2 2 3 2 2 1
 1 1 3 1 1 2 2 3 1 2 1 1 2 1]
Correct predicting ratio:  1.0
```

Prediction : [3, 3, 1, 1, 1, 2, 2, 1, 2, 2, 2, 3, 3, 1, 2, 3, 3, 1, 1, 3, 2, 2, 3, 2, 2, 2, 3, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 2, 1, 3, 1, 1, 3, 3, 1, 1, 3, 1, 3, 3, 2, 2, 3, 2, 3, 3, 1, 1, 2, 2, 3, 2, 2, 1, 1, 1, 3, 1, 1, 2, 2, 3, 1, 2, 1, 1, 2, 1]

Question 2



```python
if __name__ == '__main__':
    dataSet, labels = loadDataSet()
    dataSet2,labels2 = loadDataSet2()
    predict_result=[]
    prediction=[]
    #prediction
    for i,j in zip(dataSet2,labels2):
        predict_result.append([i,classify2(i,dataSet2,labels2,3),j])  # k value tune here ^ ^
        prediction.append(classify2(i,dataSet2,labels2,3))           # k value tune here ^ ^
        correct_count=0
    #compare prediction with coreect result, and caculate the acuuracy of prediction
    for i in predict_result:
        if(i[1]==i[2]):
            correct_count+=1
    ratio=float(correct_count)/len(predict_result)
    print("---------------------Prediction---------------------")
    print(prediction)
    print("---------------------Correct results ---------------------")
    print(labels2)
    print ("Correct predicting ratio: ",ratio)
```

```
sys.path.extend(['/Users/Patrick/PycharmProjects/COMP307-A1'])

Python 3.7.4 (default, Aug 13 2019, 15:17:50)
In[2]: runfile('/Users/Patrick/PycharmProjects/COMP307-A1/KNN.py', wdir='/Users/Patrick/PycharmProjects/COMP307-A1')
---------------------Prediction---------------------
[3, 3, 1, 1, 1, 2, 2, 1, 2, 2, 3, 3, 3, 1, 2, 3, 3, 1, 1, 3, 2, 2, 3, 2, 3, 2, 3, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 3,
---------------------Correct results ---------------------
[3 3 1 1 1 2 2 1 2 2 2 3 3 1 2 3 3 1 1 3 2 2 3 2 2 2 3 2 1 2 1 2 1 2 2 2 2
 2 1 2 2 3 1 2 1 3 2 2 1 3 1 1 3 3 1 1 3 1 3 3 2 2 3 2 3 3 1 1 2 2 3 2 2 1
 1 1 3 1 1 2 2 3 1 2 1 1 2 1]
Correct predicting ratio:  0.9545454545454546
```

As we can see above, One(k=1) neighbor's performance is better than three(k=3). But it is because overfitting. When the value of k is equal to one, we are equivalent to finding only the nearest neighbor, so no matter how to find the result is always correct, the correct rate is always 1, but we make this algorithm for classification, obviously we do not do when k = 1 To the classification operation. When k = 3, we will find the three nearest neighbors, and then determine which category they belong to, so the result will be lower.

Question 3

Advantages:

1. KNN is more suitable for application in low-dimensional space.

2. It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it.

3. No complicated mathematical knowledge, easy to understand and easy to use.

4. It can deal with non linear separable data set.

Disadvantages:

1. I don't think knn can give a reasonable explanation for data classification. Compared with decision trees, knn lacks logic.

2. In my view, one of the main disadvantages is that when the samples are unbalanced, for example, the sample size of one class is large, while the sample sizes of other classes are small, it may lead to a large capacity class among the K neighbors of the sample when a new sample is input The majority of samples.

3. Another disadvantage of this method is that it requires a large amount of calculation, because for each text to be classified, its distance to all known samples must be calculated to obtain its K nearest neighbors.

Question 4

Step1: I will divide the training set into k fold.

Step2: trains model by using k-1 folds and left one is used for testing.

Step3: I will train this model by using using different test set and training set by different combining of k folds.

But, I would like to use

```
from sklearn.model_selection import KFold
kf = KFold(n_splits=2)
```

in sklearn of Python. And finally I would tune the parameter k of KNN classifier to 5.

Question 5

If class labels are not available, I would like to use the unsupervised method K-means Clustering.

Firstly, I will choose 3 as the parameter of K-means value because we have 3 kind of wine to identify.

Secondly, I will train model under this circumstances `n_clusters=2` .

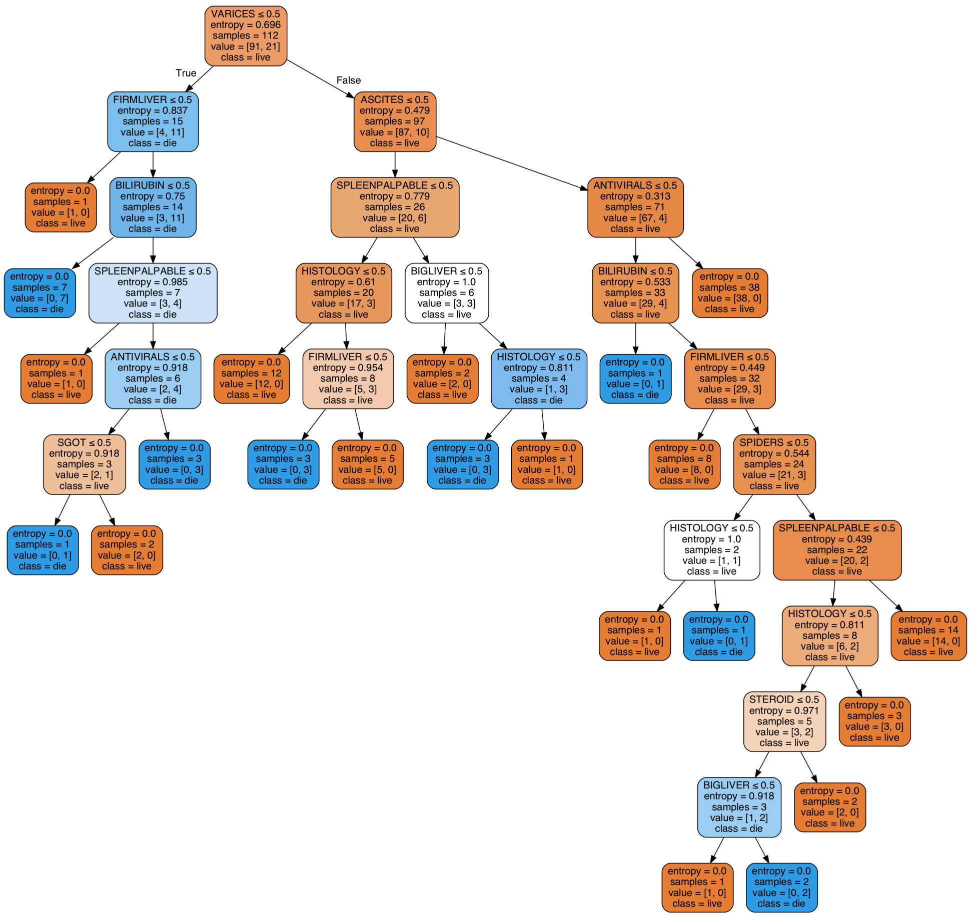
Then I will use the model to predict the data from test set and check the prediction result with the original class labels in test set. And I will tune the parameters based on the performance of model.

Part 2: Decision Tree Leaning Method

Question 1

Baseline classifier accuracy: 0.7916666666666666

The model prediction accuracy: 0.68



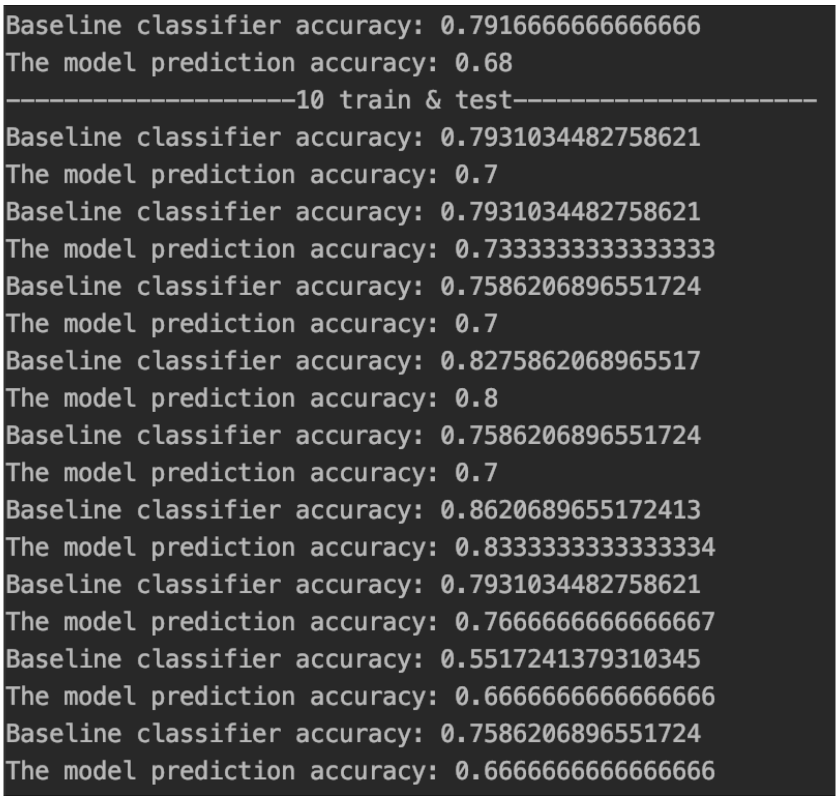
VARICES ≤ 0.5 entropy = 0.696 samples = 112 value = [91, 21] class = live
True
False
FIRMLIVER ≤ 0.5 entropy = 0.837 samples = 15 value = [4, 11] class = die
ASCITES ≤ 0.5 entropy = 0.479 samples = 97 value = [87, 10] class = live
entropy = 0.0 samples = 1 value = [1, 0] class = live
BILIRUBIN ≤ 0.5 entropy = 0.75 samples = 14 value = [3, 11] class = die
SPLEENPALPABLE ≤ 0.5 entropy = 0.779 samples = 26 value = [20, 6] class = live
ANTIVIRALS ≤ 0.5 entropy = 0.313 samples = 71 value = [67, 4] class = live
entropy = 0.0 samples = 7 value = [0, 7] class = die
SPLEENPALPABLE ≤ 0.5 entropy = 0.985 samples = 7 value = [3, 4] class = die
HISTOLOGY ≤ 0.5 entropy = 0.61 samples = 20 value = [17, 3] class = live
BIGLIVER ≤ 0.5 entropy = 1.0 samples = 6 value = [3, 3] class = live
BILIRUBIN ≤ 0.5 entropy = 0.533 samples = 33 value = [29, 4] class = live
entropy = 0.0 samples = 38 value = [38, 0] class = live
entropy = 0.0 samples = 1 value = [1, 0] class = live
ANTIVIRALS ≤ 0.5 entropy = 0.918 samples = 6 value = [2, 4] class = die
entropy = 0.0 samples = 12 value = [12, 0] class = live
FIRMLIVER ≤ 0.5 entropy = 0.954 samples = 8 value = [5, 3] class = live
entropy = 0.0 samples = 2 value = [2, 0] class = live
HISTOLOGY ≤ 0.5 entropy = 0.811 samples = 4 value = [1, 3] class = die
entropy = 0.0 samples = 1 value = [0, 1] class = die
FIRMLIVER ≤ 0.5 entropy = 0.449 samples = 32 value = [29, 3] class = live
SGOT ≤ 0.5 entropy = 0.918 samples = 3 value = [2, 1] class = live
entropy = 0.0 samples = 3 value = [0, 3] class = die
entropy = 0.0 samples = 3 value = [0, 3] class = die
entropy = 0.0 samples = 5 value = [5, 0] class = live
entropy = 0.0 samples = 3 value = [0, 3] class = die
entropy = 0.0 samples = 1 value = [1, 0] class = live
entropy = 0.0 samples = 8 value = [8, 0] class = live
SPIDERS ≤ 0.5 entropy = 0.544 samples = 24 value = [21, 3] class = live
entropy = 0.0 samples = 1 value = [0, 1] class = die
entropy = 0.0 samples = 2 value = [2, 0] class = live
HISTOLOGY ≤ 0.5 entropy = 1.0 samples = 2 value = [1, 1] class = live
SPLEENPALPABLE ≤ 0.5 entropy = 0.439 samples = 22 value = [20, 2] class = live
entropy = 0.0 samples = 1 value = [1, 0] class = live
entropy = 0.0 samples = 1 value = [0, 1] class = die
HISTOLOGY ≤ 0.5 entropy = 0.811 samples = 8 value = [6, 2] class = live
entropy = 0.0 samples = 14 value = [14, 0] class = live
STEROID ≤ 0.5 entropy = 0.971 samples = 5 value = [3, 2] class = live
entropy = 0.0 samples = 3 value = [3, 0] class = live
BIGLIVER ≤ 0.5 entropy = 0.918 samples = 3 value = [1, 2] class = die
entropy = 0.0 samples = 2 value = [2, 0] class = live
entropy = 0.0 samples = 1 value = [1, 0] class = live
entropy = 0.0 samples = 2 value = [0, 2] class = die

The accuracy of baseline is 0.79, while DT model classification accuracy is lower than this value -0.68. The reason is because imbalance of dataset(live is much more than die), the baseline model always predict live as outcomes. And at the same time, the test set have same problem.

Question2

	The model prediction accuracy	Baseline classifier accuracy (2dp)
1	0.7	0.79
2	0.73	0.79
3	0.7	0.76
4	0.8	0.83
5	0.7	0.76
6	0.83	0.86
7	0.77	0.79
8	0.67	0.55
9	0.67	0.75
Average accuracy	0.73	0.764444444444444

These are outcomes from the program running, the above table is from these results.



```
Baseline classifier accuracy: 0.7916666666666666
The model prediction accuracy: 0.68
--------------------10 train & test----------------------
Baseline classifier accuracy: 0.7931034482758621
The model prediction accuracy: 0.7
Baseline classifier accuracy: 0.7931034482758621
The model prediction accuracy: 0.7333333333333333
Baseline classifier accuracy: 0.7586206896551724
The model prediction accuracy: 0.7
Baseline classifier accuracy: 0.8275862068965517
The model prediction accuracy: 0.8
Baseline classifier accuracy: 0.7586206896551724
The model prediction accuracy: 0.7
Baseline classifier accuracy: 0.8620689655172413
The model prediction accuracy: 0.8333333333333334
Baseline classifier accuracy: 0.7931034482758621
The model prediction accuracy: 0.7666666666666667
Baseline classifier accuracy: 0.5517241379310345
The model prediction accuracy: 0.6666666666666666
Baseline classifier accuracy: 0.7586206896551724
The model prediction accuracy: 0.6666666666666666
```

Question 3

(a) The pruning process is to check a group of nodes with the same parent node, and determine whether the increase in entropy will be less than a specified threshold if they are merged. If this is the case, the leaf nodes will be merged into a single node, and the merged new node contains all possible result values. This approach helps to avoid overfitting, so that the prediction results made by the decision tree are not more special than the actual conclusions obtained from the data set.

So the algorithm will traverse down all the paths of the tree to the nodes that only contain leaf nodes. The function combines the result values of the two leaf nodes to form a new list, and also tests the entropy. If the change in entropy is less than the value specified by the minimum gain parameter, the leaf node may also become the object of deletion and the object of merging with other nodes.

(b) Pruning leaves could reduce the overfitting. Overfitting can give us fake but Satisfactory results on training set.This can be understood as our model has already backed the training set, but it is clear that when we apply this model to other data sets, our too clever model can not distinguish.

(c) Pruning leaves can increase the accuracy of test sets classification due to we reduce the overfitting which make our model learnt more and deeper than directly learnt by rote.

Question 4

According to the formula for calculating impurity, when there are only two categories, an impurity of 0 means the highest purity because it does not contain another category. But when there are more than two categories, we can assume (when there are three) that the C of ABC is zero. According to the formula (P(A)*P(B)*P(C) = 0.6 * 0.4 * 0 = 0), we get that the impurity of this node is zero, but this node still contains two categories, still Is impure. Based on this concern, I did not use Gini impurity as the basis for classification, but I know that our data set is completely binary.

(And through my program, the result of Gini Impurity is better than that of Information Gain.)