Homework 4

Collaborators:

Name: Youchao Zhang

Student ID: 3170100125

Problem 4-1. K-Nearest Neighbor

In this problem, we will play with K-Nearest Neighbor (KNN) algorithm and try it on real-world data. Implement KNN algorithm (in *knn.py*), then answer the following questions.

(a) Try KNN with different K and plot the decision boundary.

Answer:

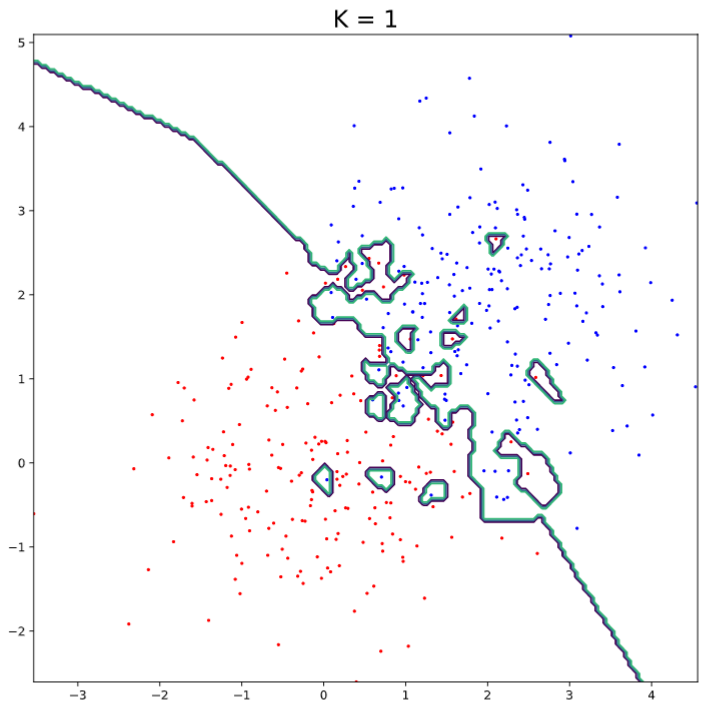

Figure 1: K = 1

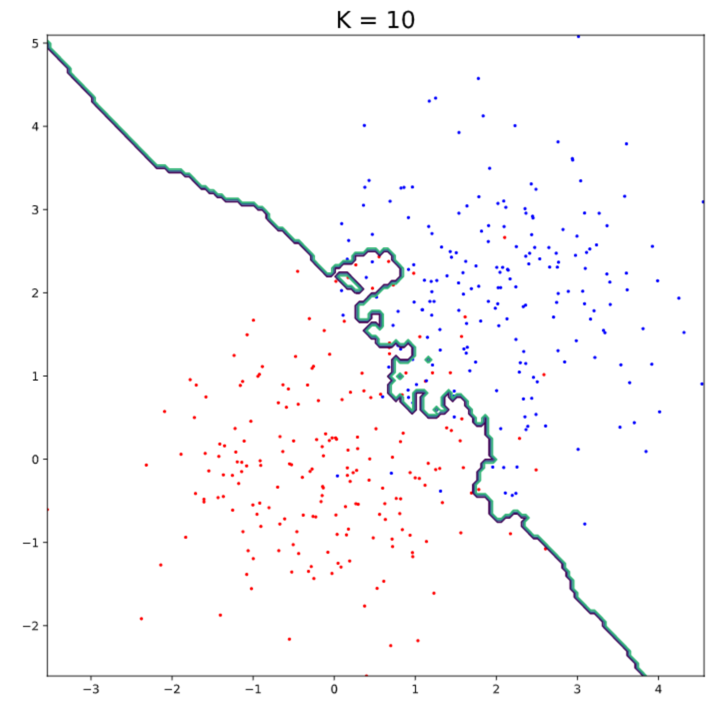

Figure 2: K = 10

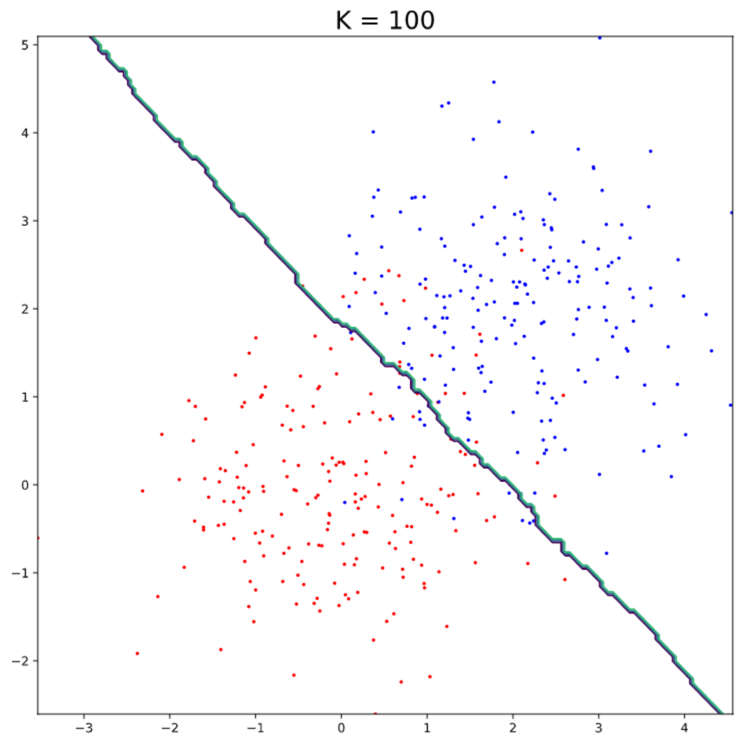

Figure 3: K = 100

(b) We have seen the effects of different choices of K. How can you choose a proper K when dealing with real-world data ?

Answer:
1.If the K value is small, the model complexity is high, over-fitting is prone to occur, the estimation error of learning will increase, and the prediction result is very sensitive to the instance points of the neighbors
2.A larger K value can reduce the estimation error of learning, but the approximate error of learning will increase. Training instances far from the input instance will also play a role in the prediction, making the prediction wrong, and increasing the k value will reduce the complexity of the model .
3.In application, the k value is generally a relatively small value, and cross-validation is usually used to select the optimal K value.

(c) Finish *hack.py* to recognize the CAPTCHA image using KNN algorithm.

Answer:
The accuracy = 100%
There some detail in run.ipynb

Problem 4-2. Decision Tree and ID3

Consider the scholarship evaluation problem: selecting scholarship recipients based on gender and GPA. Given the following training data:

Answer:

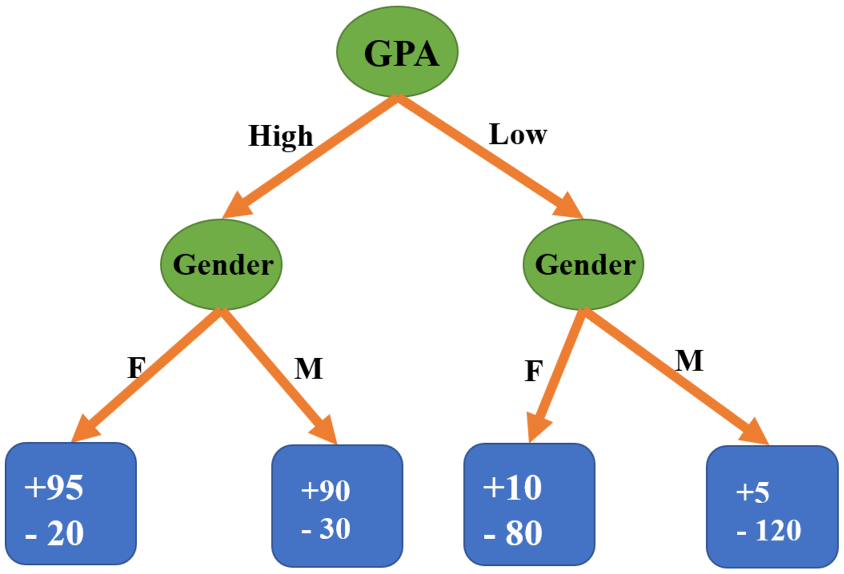

Figure 4: The tree

Problem 4-3. A Walk Through Ensemble Models

In this problem, you will implement a whole bunch of ensemble models and compare their performance and properties.

(a) Implement decision tree algorithm (in decision tree.py), then answer the following questions.

Answer:

(i)

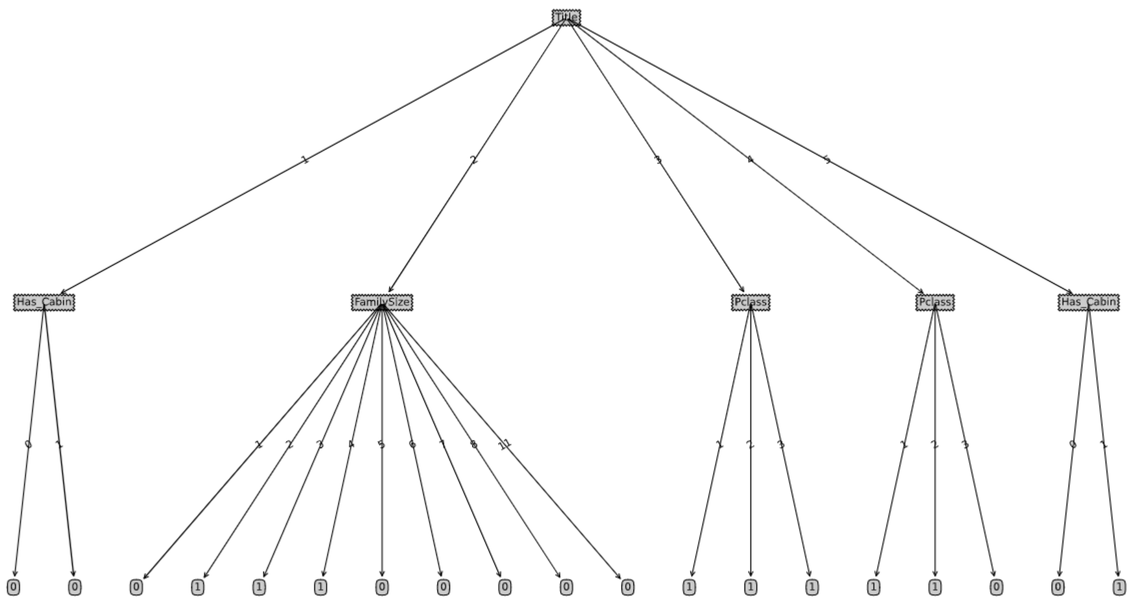

Figure 5: The tree

Accuracy on train set: 0.8013371537726839

(ii)

(a)"Title"=5

(b)"Has Cabin"=5

(c)Survived

(iii)

The larger the max depth, the smaller the min data leaf, the larger the variance of the model, and the easier the decision tree model is to overfit. Conversely, the greater the deviation of the model, the poorer the learning ability of the model, and the easier it is to underfit.

(iv)

The training error rate = 12.6%

The validation error rate = 23.3%

I think it is overfitting.

(v)
Best parameters: criterion = infogain ratio, max depth = 9, min samples leaf = 4
Accuracy on train set: 85.9%
Accuracy on test set : 80.2%
(vi)
The selection methods of different features have a great impact on the construction of decision trees, and cross-validation should be used to select the best.

(b) Implement Random Forest (in random forest.py), then answer the following questions.

Answer:

(i)
The depth of the number increases, the number of smallest leaf nodes decreases. Because the fitting ability of each tree is generally insufficient, and even overfitting occurs, but in random forest, after the number of trees increases, the predictive fitting ability of the entire model will be significantly improved
(ii)
number = 10
train error rate=20.9% test error rate=21.9%
number = 100
train error rate=19.9% test error rate=20.2%
(iii)
1) Reduce the impact of outliers: Because random forest has selected part of the data to build multiple decision trees, even some individual decision trees will cause inaccurate predictions due to the impact of outliers, but the prediction results are based on multiple decision trees The result obtained reduces the influence of outliers.
2) Reduce the possibility of over-fitting, because the decision tree uses all the features and samples, it is prone to over-fitting (that is, it has a good effect on the training sample, but has a poor effect on the test set), random forest Many decision trees are constructed by using some features of some samples (replacement sampling is adopted). Feature and data are reduced on a single decision tree, reducing the possibility of over-fitting.
3)Mainly reduces the variance of the model
(iv)
Best parameters :
criterion = entropy, max depth = 11, min samples leaf = 1
Accuracy on train set: 0.803
Accuracy on test set: 0.812

(c) Implement Adaboost (in adaboost.py), then answer the following questions.

Answer:

(i)
Random forest reduces the variance of decision trees by combining a variety of complex classifiers
The adaboost algorithm obtains a classifier with little bias by linearly combining each type of classifier, so the decision tree stump is more suitable
(ii)
number = 10
train error rate=12.0% test error rate=27.0%
number = 100
train error rate=10.0% test error rate=26.6%
(iii)
Adaboost is an iterative algorithm. Its core idea is to train different classifiers (weak classifiers) for the same training, and then group these weak classifiers to form a stronger final classifier (strong classifier).
reduce the bias
(iv)
Best parameters :
criterion = entropy, max depth = 10, min samples leaf = 1
Accuracy on train set: 0.873
Accuracy on test set: 0.782