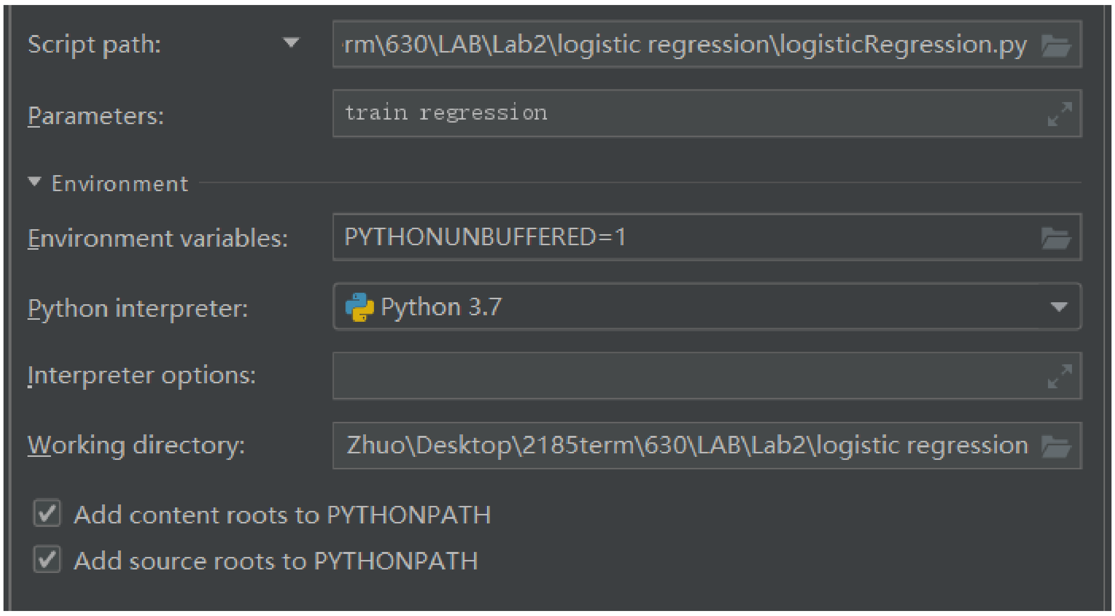
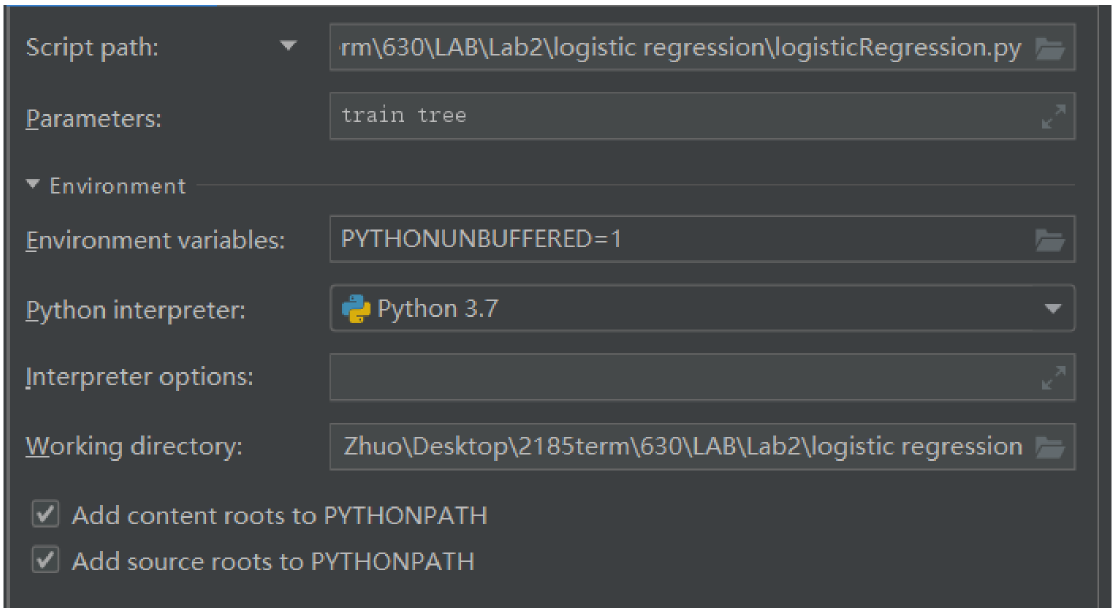
Zhuo Liu
CSCI-630-02 Foundations of FIS
Lab2 report

Train command line operation



```
Script path:               rm\630\LAB\Lab2\logistic regression\logisticRegression.py
Parameters:                train regression
Environment
Environment variables:     PYTHONUNBUFFERED=1
Python interpreter:        Python 3.7
Interpreter options:
Working directory:         Zhuo\Desktop\2185term\630\LAB\Lab2\logistic regression
[✓] Add content roots to PYTHONPATH
[✓] Add source roots to PYTHONPATH
```

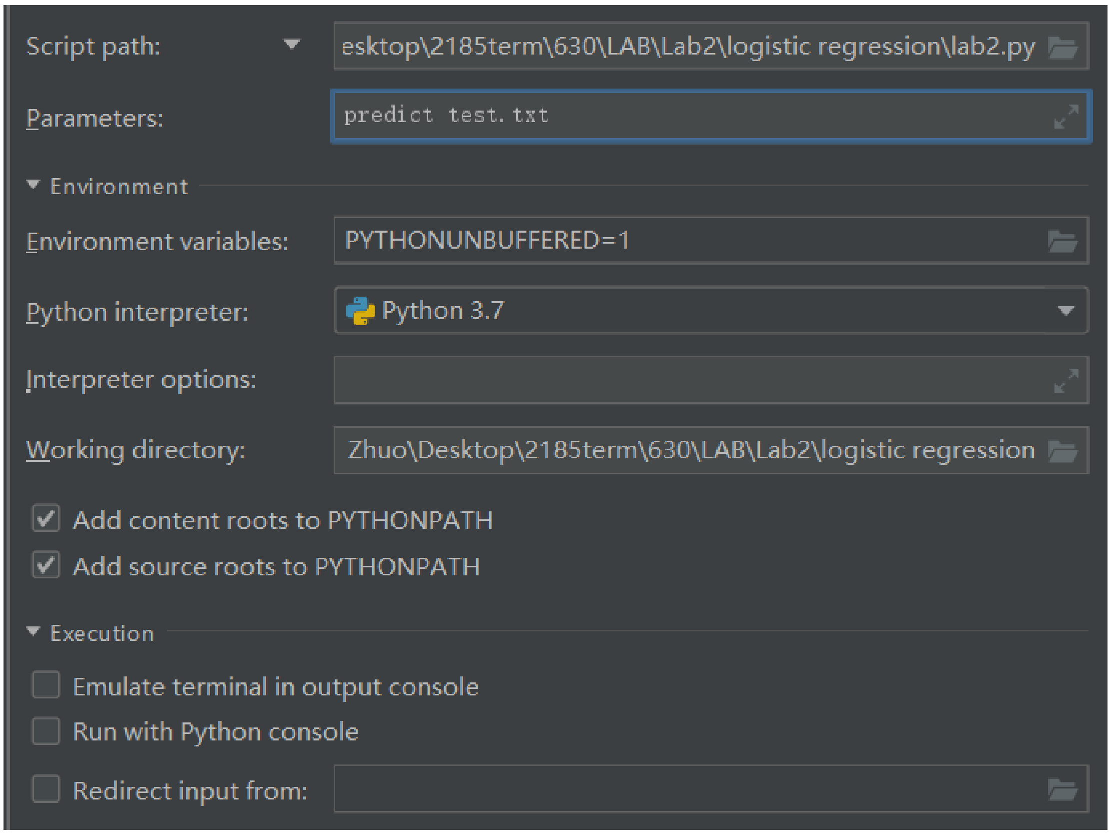
If you want to do training operation for the logistic regression part, please input as the figure shown above. It will take a little bit longer, because in order to get an accurate training model of weights array. I set the iteration to be 1000 times.
Please Note: Every time after you finish training process, please keep Lab2.py, 4signature.txt, whale.txt, test.txt, my_file.npy and delete all other irrelevant files. Or there will be a mistake if you want to train another model. If you just want to predict, you don't have to do anything.



```
Script path:               rm\630\LAB\Lab2\logistic regression\logisticRegression.py
Parameters:                train tree
Environment
Environment variables:     PYTHONUNBUFFERED=1
Python interpreter:        Python 3.7
Interpreter options:
Working directory:         Zhuo\Desktop\2185term\630\LAB\Lab2\logistic regression
[✓] Add content roots to PYTHONPATH
[✓] Add source roots to PYTHONPATH
```

If you want to do training operation for the decision tree, please input as the figure shown above.

Predict command line operation



Script path: esktop\2185term\630\LAB\Lab2\logistic regression\lab2.py

Parameters: predict test.txt

Environment

Environment variables: PYTHONUNBUFFERED=1

Python interpreter: Python 3.7

Interpreter options:

Working directory: Zhuo\Desktop\2185term\630\LAB\Lab2\logistic regression

✓ Add content roots to PYTHONPATH

✓ Add source roots to PYTHONPATH

Execution

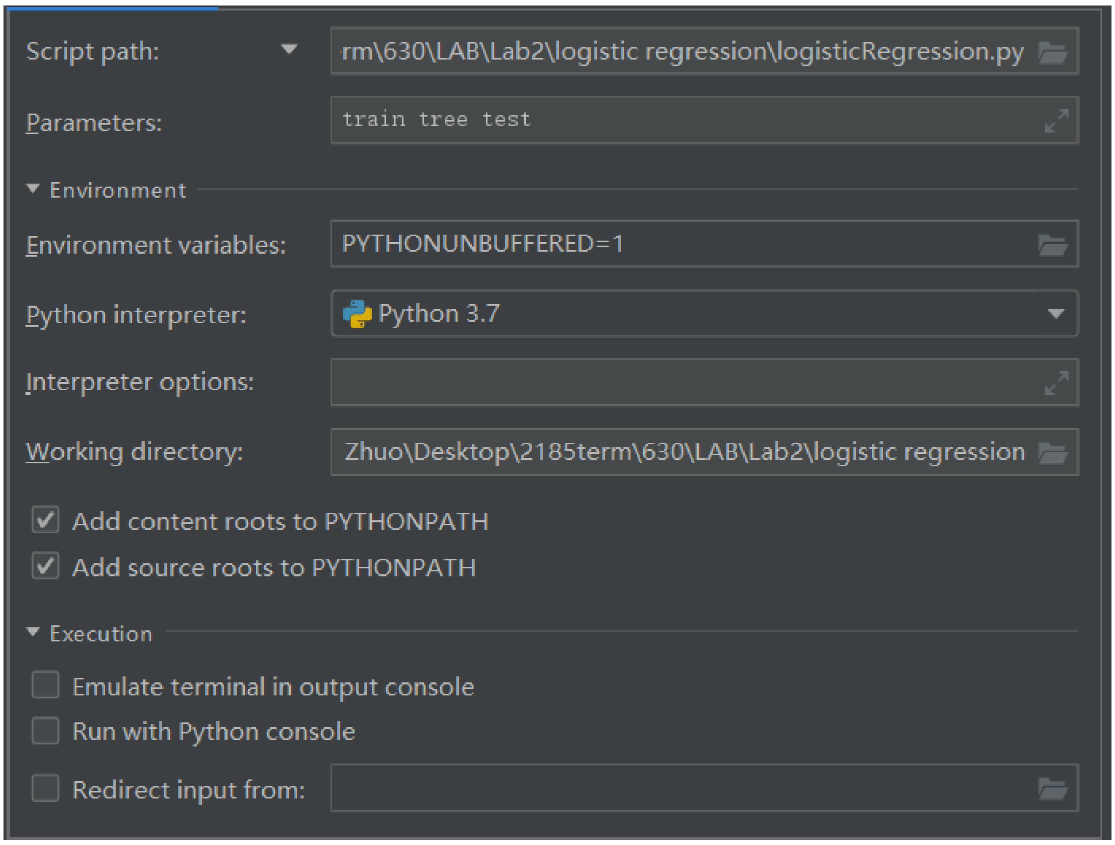
☐ Emulate terminal in output console

☐ Run with Python console

☐ Redirect input from:

If you want to do predict operation for the logistic regression part, please input as the figure shown above. You can put any 250 words in the test.txt file.

If the output is closer to 1, it means the book is written by Conan Doyle. If the output is closer to 0, the book is written by Herman Melville.


Script path: rm\630\LAB\Lab2\logistic regression\logisticRegression.py

Parameters: train tree test

Environment

Environment variables: PYTHONUNBUFFERED=1

Python interpreter: Python 3.7

Interpreter options:

Working directory: Zhuo\Desktop\2185term\630\LAB\Lab2\logistic regression

✓ Add content roots to PYTHONPATH

✓ Add source roots to PYTHONPATH

Execution

☐ Emulate terminal in output console

☐ Run with Python console

☐ Redirect input from:

If you want to do predict operation for the decision tree part, please input as the figure shown above. Please put 250 words text into test.txt file.

If the output is “Y”, it means the book is written by Conan Doyle. If the output is “N”, the book is written by Herman Melville.

Feature Defense

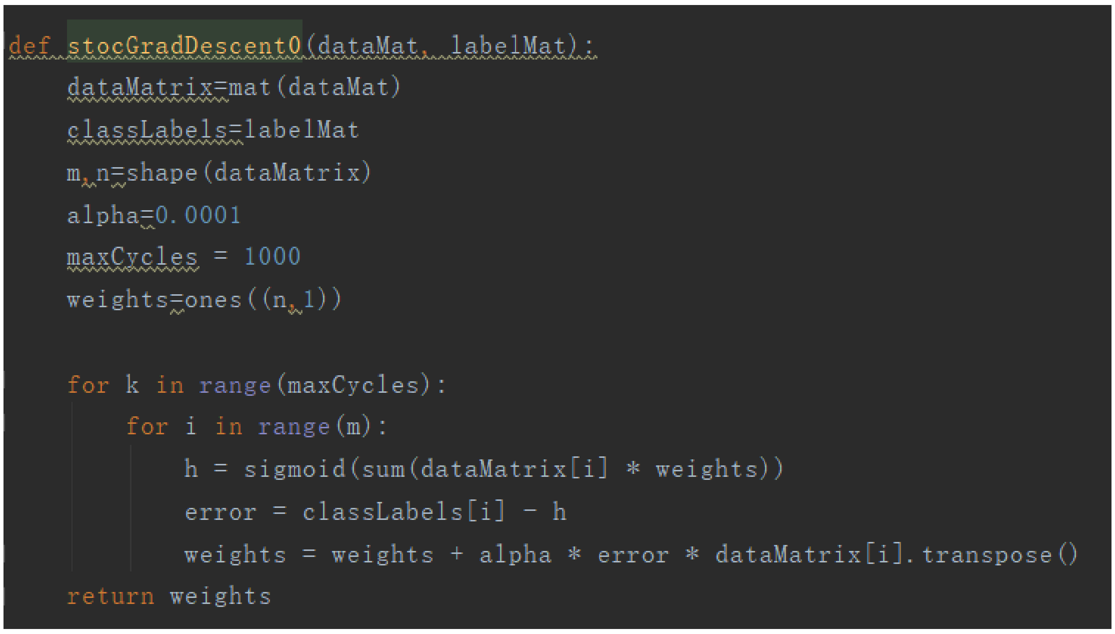
I have chosen 10 features in my program. These features are shown as followed:

1) proposition which is also considered to be “the”,
2) the exclamation mark,
3) question mark,
4) the adjectives which end in “ed” or “ive”,
5) adverbs which end in “ly”,
6) Sherlock: the first name of the main character in Conan Doyle’s novels
7) Holmes: the last name of the main character in Conan Doyle’s novels
8) He
9) She
10) The adjectives which end in “ing”

The reason why I choose these five features is shown as followed:
The frequency of proposition may show the writing habit of the author.
The amount of exclamation marks and question marks can express emotion of specific author which can be used as a feature.
The number of adjectives and adverbs are kind of rhetorical devices which can also give us the hint who wrote the book.
Since in most of Conan Doyle’s novel, Sherlock Holmes is the main character. From my perspective, Sherlock and Holmes can be used as two features to improve the accuracy.
The type of Conan Doyle’s novels is detective novel, so most of the characters are male. So in my opinion, he and she could be used to defense my choice.
The reason why I use adjectives which end in “ing” as one feature is largely same as adjectives which end in “ed” or “ive”. The more features I choose, the more accuracy my program will be. (But we should avoid overfitting)

Training Process

There are two training processes in my code. The first is a logistic regression.

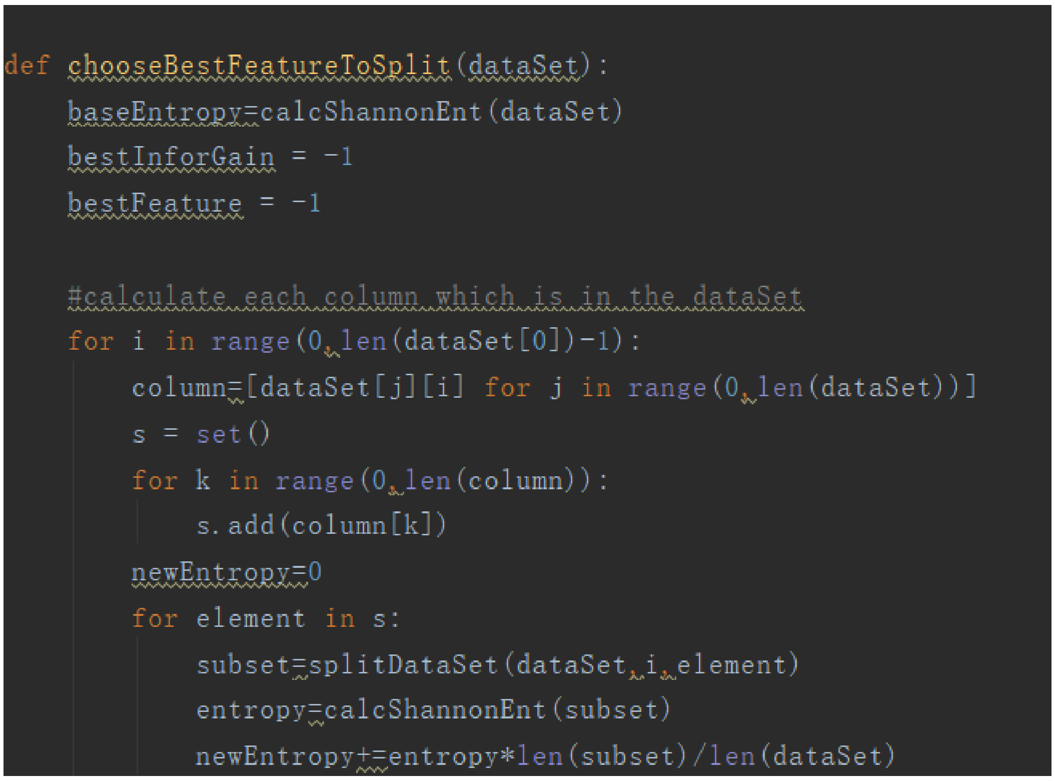


```
def stocGradDescent0(dataMat, labelMat):
    dataMatrix=mat(dataMat)
    classLabels=labelMat
    m,n=shape(dataMatrix)
    alpha=0.0001
    maxCycles = 1000
    weights=ones((n,1))

    for k in range(maxCycles):
        for i in range(m):
            h = sigmoid(sum(dataMatrix[i] * weights))
            error = classLabels[i] - h
            weights = weights + alpha * error * dataMatrix[i].transpose()
    return weights
```

The figure which is shown above can show the main idea of this training process.

I use the concept of single perceptron of neural network. From the very beginning I set all the initial weights to be 1 and learning rate to be 0.0001. I choose sigmoid function to be activation function. Finally I use gradient descent method to return the weights after the training process.

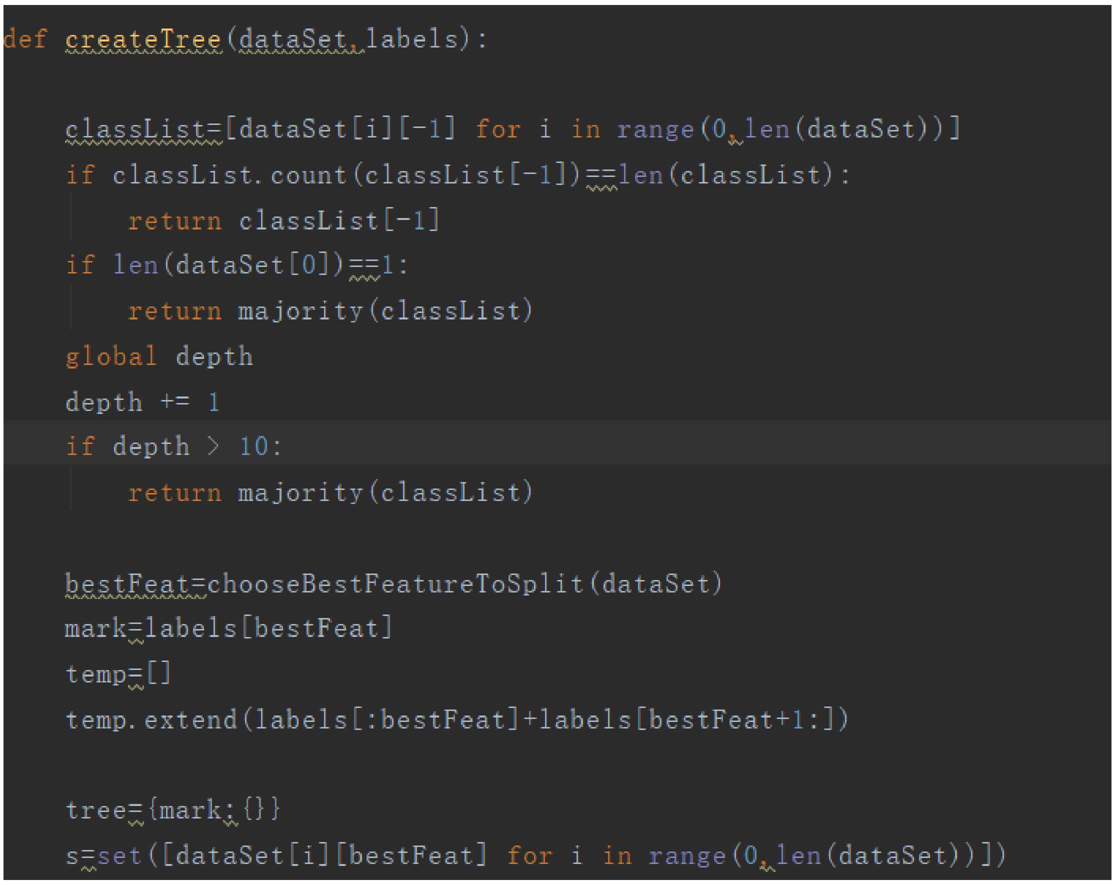


```
def chooseBestFeatureToSplit(dataSet):
    baseEntropy=calcShannonEnt(dataSet)
    bestInforGain = -1
    bestFeature = -1

    #calculate each column which is in the dataSet
    for i in range(0,len(dataSet[0])-1):
        column=[dataSet[j][i] for j in range(0,len(dataSet))]
        s = set()
        for k in range(0,len(column)):
            s.add(column[k])
        newEntropy=0
        for element in s:
            subset=splitDataSet(dataSet,i,element)
            entropy=calcShannonEnt(subset)
            newEntropy+=entropy*len(subset)/len(dataSet)
```

Another training process I choose to use is a decision tree which based on entropy calculation.

As mentioned in the CSCI630 lecture, we should choose the best feature to split the dataset and calculate the entropy accordingly. In my code, finally I get a tree based on the data structure of a dictionary. This dictionary is considered to be the training model.



```python
def createTree(dataSet,labels):

    classList=[dataSet[i][-1] for i in range(0,len(dataSet))]
    if classList.count(classList[-1])==len(classList):
        return classList[-1]
    if len(dataSet[0])==1:
        return majority(classList)
    global depth
    depth += 1
    if depth > 10:
        return majority(classList)

    bestFeat=chooseBestFeatureToSplit(dataSet)
    mark=labels[bestFeat]
    temp=[]
    temp.extend(labels[:bestFeat]+labels[bestFeat+1:])

    tree={mark:{}}
    s=set([dataSet[i][bestFeat] for i in range(0,len(dataSet))])
```

Please see the figure shown above.

In my createTree function, if the depth of the decision tree is greater than a given number, the program immediately return the majority results which means the maximal number of “Y” or “N”. Thus, I can tune the complexity of my decision tree

Logistic regression	5 features	7 features	10 features
accuracy	0.5	0.6	0.7

In the second column, the 5 features which I choose are "the", exclamation mark, question mark, the adjectives which end in "ed" or "ive" and adverbs which end in "ly".
The reason why the accuracy is low is that these five features are not obvious enough, these five features may show the writing habit of specific author. But in my opinion, I should add some more features to improve accuracy.

In the third column, I add two more features which are "Sherlock" and "ing", in some cases, the accuracy is improved. But if I choose a 250 words paragraph without the name of main character, the rate of accuracy will also drop.

In the last column, I have all the features in my program. As I mentioned in the beginning of the report, a full consideration of all the features can improve the overall accuracy. But in some cases, there are some errors.

Decision tree	5 features	7 features	10 features
accuracy	0.6	0.7	0.8

The overall accuracy, decision tree is much better than the logistic regression. It is because the decision is based entropy calculation and we always choose the best feature to split the dataset, if we decision tree reach certain depth, we can even get an overfitting result.