***Instruction:***

The file ‘data’ stores the training data with three books per author (Conan Doyle, Herman Melville) by number 1-3, the file ‘test-data’ stores the test data with two books per author by number 4-5. In the experiment, we extract 100 250-words short texts for each book, which can be de designated by command argument ‘per\_class\_max\_docs’. The directory ‘slp’ and ‘decision tree’ store the code of two models and binary file respectively. Besides, for code readability, I coded ‘util.py’ and ‘load\_data.py’ to process the matrix operations and loading data. The ‘stopwords.txt’ stores the common stop words for calculating the notional word with the highest tf-idf value, because some words don’t make sense, and the word frequency is very high. For each file of corresponding model, the ‘words.txt’ stores the words need to count the frequency.

The bonus part needs to add books of Jane Austin in ‘bonus’ file to ‘data’ and ‘test-data’ files (1-3 in ‘data’, 4-5 in ‘test-data’).

***Choose of features:***

The selected 18 features can be divided into three categories:

The frequency of function words 'he she the of how a had could' and 9 words 'miss one mr. like said upon mrs. old holmes' which selected according to the highest value of tf-idf (each author three words), and the average length of word for each doc to be predicted.

The frequency of function words and the average length of word for each doc can reflect the author's writing style. Words with the highest tf-idf value can be considered as keywords for each author's short text.

The tf-idf value can calculated as follow:

where count the frequency of word .

However, in the decision tree (dt) model, if there are too many features, the dt will be overfitting and use too many irrelevant input features. In the logistic classifier, if the features are too few, it is difficult for logistic classifier to learn the characteristics of the author. In the following experiment results, will compare the relationship between feature number and model performance.

***Training process and Results:***

**The training process of decision tree**, I tune the information gain threshold ∈ {0.005, 0.01} and the depth ∈ {4, 6, 8, 10, 12} under the same data set to get the best model (the entropy cutoff and the depth cutoff). And I process the feature with the average of keywords frequency for each author as the threshold:

where the is the total frequency of the keyword in total 250 words short texts, is the total number of short texts of each author; is the frequency of each keyword, is the threshold (i. e. mean frequency) of each keyword.

**Experiment results:**

First, fix the threshold at 0.005 to operate the entropy cutoff, then tune the depth to operate the depth cutoff. When the number of depth increases until 8, the accuracy is increasing, after then, the performance drops. Then, I tune the threshold to 0.01, and redo the experiment.

a. Original 18 features.

The results are reported in following Table 1:

Table 1. Decision tree results with original 18 features

|  |  |  |
| --- | --- | --- |
| Depth | Threshold | Accuracy |
| 4 | 0.005 | 0.7450 |
| 6 | 0.005 | 0.7525 |
| 8 | 0.005 | 0.7650 |
| 10 | 0.005 | 0.7950 |
| 12 | 0.005 | 0.7750 |
| 4 | 0.01 | 0.7450 |
| 6 | 0.01 | 0.7475 |
| 8 | 0.01 | 0.7650 |
| 10 | 0.01 | **0.7950** |
| 12 | 0.01 | 0.7775 |

b. Delete some functional words - 'the of how a had could', that is only left 11 + 1 features.

Follow the previous configuration, I redo the experiment and reported the accuracy results.

Table 2. Decision tree results with 12 features

|  |  |  |
| --- | --- | --- |
| Depth | Threshold | Accuracy |
| 4 | 0.005 | 0.7450 |
| 6 | 0.005 | 0.7525 |
| 8 | 0.005 | 0.7600 |
| 10 | 0.005 | 0.7625 |
| 12 | 0.005 | 0.7625 |
| 4 | 0.01 | 0.7450 |
| 6 | 0.01 | 0.7500 |
| 8 | 0.01 | 0.7600 |
| 10 | 0.01 | 0.7625 |
| 12 | 0.01 | 0.7625 |

Comparing the results of the two experiments, we can see that the performance of dt is not good when the number of features are deleted.

In the final, I choose the best model with the depth 10, threshold 0.01, and use 18 features, which obtain 79.50% accuracy.

**The training process of logistic classifier**, I tune the learning rate∈ {0.01, 0.001} and epochs∈ {1000, 3000, 5000, 10000} according to the accuracy on test data, and use the normalized (min-max normalization) original features (word frequency without processing by the threshold).

**Experiment results:**

a. Original 18 features.

The results are reported in following Table 3:

Table 3. Logistic classifier results with original 18 features

|  |  |  |
| --- | --- | --- |
| Epoch | Learning rate | Accuracy |
| 1000 | 0.01 | 0.6950 |
| 3000 | 0.01 | 0.7250 |
| 5000 | 0.01 | 0.7325 |
| 10000 | 0.01 | 0.7350 |
| 1000 | 0.001 | **0.8000** |
| 3000 | 0.001 | 0.7800 |
| 5000 | 0.001 | 0.7875 |
| 10000 | 0.001 | 0.7900 |

b. Delete some notional words (the highest tf-idf value words) – only left 'he she the of how a had could', that is only left 8 + 1 features.

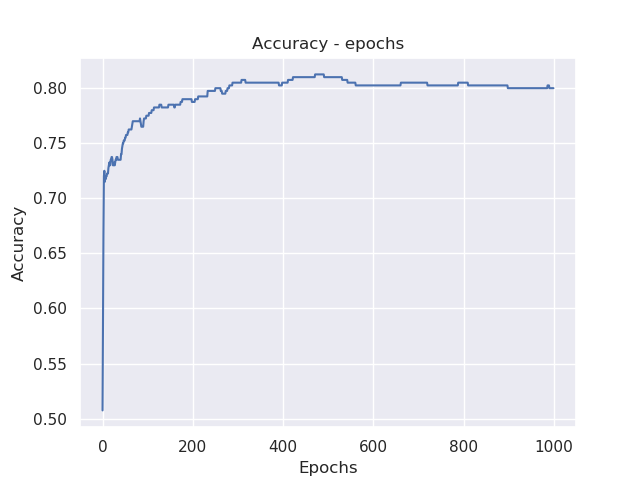
Table 4. Logistic classifier results with 12 features

|  |  |  |
| --- | --- | --- |
| Epoch | Learning rate | Accuracy |
| 1000 | 0.01 | 0.5825 |
| 3000 | 0.01 | 0.5925 |
| 5000 | 0.01 | 0.5925 |
| 10000 | 0.01 | 0.5925 |
| 1000 | 0.001 | 0.7050 |
| 3000 | 0.001 | 0.6925 |
| 5000 | 0.001 | 0.6975 |
| 10000 | 0.001 | 0.6875 |

First, I set the learning rate to 0.01. Since the accuracy of the model has been increasing when the model epoch number is 1000, 3000, 5000. Therefore, the epoch number is gradually increased until the model performance is converged, i. e. epoch 10000. Then, I set the learning rate to 0.001, and obtain the best accuracy when the epoch is set to 1000.And from the accuracy-epochs curve figure (Figure 1.), we can see the logistic classifier model gradually converge.

Besides, compare the results whether use the highest tf-idf value words, when the highest tf-idf value words is not used, the logistic classifier performance is poor.

Figure 1. The accuracy-epochs curve of best logistic classifier model



In the final, I choose the best logistic classifier model with the epoch 1000, learning rate 0.01, and original 18 features.

***Bonus:***

**How to classify the three authors:**

To classify three authors, logistic classifier uses the softmax classification function, and the weight matrix (is the dimension of feature).

For the cross entropy loss function, the gradient of is not affected by the activation function:

**Experiment results:**

We redo the experiment under the same configuration in two authors classification.

The results are reported in following text:

**The results of decision tree**

a.Original 18 features, and processing feature as two authors classification.

Table 1. Decision tree results with original 18 features

|  |  |  |
| --- | --- | --- |
| Depth | Threshold | Accuracy |
| 4 | 0.005 | 0.7183 |
| 6 | 0.005 | 0.6750 |
| 8 | 0.005 | 0.7217 |
| 10 | 0.005 | 0.6800 |
| 12 | 0.005 | 0.6767 |
| 4 | 0.01 | 0.7183 |
| 6 | 0.01 | 0.6716 |
| 8 | 0.01 | 0.7217 |
| 10 | 0.01 | 0.6750 |
| 12 | 0.01 | 0.6783 |

b. Delete some functional words - 'the of how a had could', that is only left 11 + 1 features.

Table 2. Decision tree results with 12 features

|  |  |  |
| --- | --- | --- |
| Depth | Threshold | Accuracy |
| 4 | 0.005 | 0.7183 |
| 6 | 0.005 | 0.6750 |
| 8 | 0.005 | 0.7350 |
| 10 | 0.005 | 0.7367 |
| 12 | 0.005 | 0.7300 |
| 4 | 0.01 | 0.7183 |
| 6 | 0.01 | 0.7250 |
| 8 | 0.01 | 0.7333 |
| 10 | 0.01 | **0.7417** |
| 12 | 0.01 | 0.7250 |

Under the same configuration, for three authors classification, we obtain the best 74.17 % accuracy when the depth is set to 10, and the threshold is set to 0.01, but use only 12 features.

Compare to two authors classification, the accuracy drops about 5 percent, this may be due to insufficient number of features.

**The results of logistic classifier**

a. Original 18 features.

Table 3. Logistic classifier results with original 18 features

|  |  |  |
| --- | --- | --- |
| Epoch | Learning rate | Accuracy |
| 1000 | 0.01 | 0.7217 |
| 3000 | 0.01 | 0.7483 |
| 5000 | 0.01 | 0.7600 |
| 10000 | 0.01 | 0.7617 |
| 1000 | 0.001 | 0.7616 |
| 3000 | 0.001 | 0.7633 |
| 5000 | 0.001 | 0.7600 |
| 10000 | 0.001 | **0.7667** |

b. Delete some notional words (the highest tf-idf value words) – only left 'he she the of how a had could', that is only left 8 + 1 features.

Table 4. Logistic classifier results with 12 features

|  |  |  |
| --- | --- | --- |
| Epoch | Learning rate | Accuracy |
| 1000 | 0.01 | 0.5483 |
| 3000 | 0.01 | 0.5667 |
| 5000 | 0.01 | 0.5683 |
| 10000 | 0.01 | 0.5716 |
| 1000 | 0.001 | 0.6600 |
| 3000 | 0.001 | 0.6600 |
| 5000 | 0.001 | 0.6616 |
| 10000 | 0.001 | 0.6700 |

Under the same configuration, we obtain the best 76.67% accuracy at epoch 10000 and learning rate 0.001. And from the accuracy-epochs curve figure (Figure 1.), we can see the logistic classifier model gradually converge.

Besides, compare the results of two authors classification, the accuracy drops about 4 percent, which is similar to the decision tree, maybe also due to the lack of selected features.

Figure 1. The accuracy-epochs curve of best logistic classifier model

