***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’).

***File list:***

**.**

├── bonus

│   ├── Jane Austen01.txt

│   ├── Jane Austen02.txt

│   ├── Jane Austen03.txt

│   ├── Jane Austen04.txt

│   └── Jane Austen05.txt

├── data

│   ├── Conan Doyle01.txt

│   ├── Conan Doyle02.txt

│   ├── Conan Doyle03.txt

│   ├── Herman Melville01.txt

│   ├── Herman Melville02.txt

│   └── Herman Melville03.txt

├── decision tree

│   ├── dtfile

│   ├── dt.py

│   ├── texts

│   └── words.txt

├── load\_data.py

├── slp

│   ├── Bias.txt

│   ├── slp.py

│   ├── Weight.txt

│   └── words.txt

├── stopwords.txt

├── test-data

│   ├── Conan Doyle04.txt

│   ├── Conan Doyle05.txt

│   ├── Herman Melville04.txt

│   └── Herman Melville05.txt

├── util.py

**'decision tree'**: directory stores the code for decision tree (dt.py), and the words need to count the frequency in 'words.txt', the best model for two authors classification was saved in the 'dtfile', the 'texts' file stores the object to calculate the threshold for every keyword.

**'slp'**: directory stores the code for logistic classifier (slp.py), and the words need to count the frequency in 'words.txt', the weight matrix and bias of best model for two authors classification was saved in the Weight.txt' and 'Bias.txt' respectively.

**'data'**: directory stores the training data for two authors Conan Doyle and Herman Melville.

**'stopwords.txt'**: txt file stores the common stop words used in the process of calculating the tf-idf value.

**'test-data'**: directory stores the test data.

**'load\_data.py'**: python file to load the data.

**'util.py'**: python file for matrix operation.

**'bonus'**: directory stores the third author’s docs.

***Run file:***

If want to reproduce the experiment results, please enter the directory of corresponding model ('slp' for logistic classifier, 'decision tree' for decision tree) and use following command before any training of model:

$ python3 slp.py --predict '../test-data'

$ python3 dt.py --predict '../test-data'

The optional argument predict and the value '../test-data' is the directory of test data need to predict.

If want to train the model, enter the directory of corresponding model, then use:

$ python3 slp.py --train 1

$ python3 dt.py --train 1

Then will train the model by using training data in 'data' directory.

If want to test the model with new data, please put the test data in 'test-data' directory at first, and delete the original data, or create a new directory, And specify the value of --predict parameter to the name of the new folder. At the same time, if the test data is the original books, specify the value of --per\_class\_max\_docs parameter to the any number of short text you want to extract (when the number of words is enough) IF THE DATA IS THE SIGNLE 250 WORDS SHORT TEXT, PLEASE SPECIFY THE VALUE OF --per\_class\_max\_docs PARAMETER TO 1

E.g. python3 dt.py --predict '../test-data' --per\_class\_max\_docs 1

Besides, for the training process of decision tree model, we can use optional arguments '--max\_depth' and '--threshold', like:

$ python3 dt.py --train 1 --max\_depth 10 --threshold 0.01

For the training process of logistic classifier, we can use optional arguments '--epoch' and '--lr', like:

$ python3 slp.py --train 1 --epoch 1000 --lr 0.001