Natural Language Processing Justin Johnson

Dr Dingding Wang Z23136514

**Assignment 2**

**Text Classification With LibSVM**

**Introduction**

The WebKB data set containing 2803 training instances and 1396 test instances is used to in combination with LibSVM to conduct text classification. The data set consists of web pages collected from computer science departments of various universities. The web pages are labelled as student, faculty, project, or course. The data set was previously pre-processed with stemming and removal of stop words. The following report outlines all steps required to conduct text classification using LibSVM’s terminal interface and the WebKB data set.

**Data Formatting**

In order to use the WebKB data set for classification with LibSVM, the data must first be converted to a format acceptable by LibSVM. The WebKB data is currently in the form of a text file, where each instance is separated by the new line character. Each instance consists of the instance’s label followed by a sequence of stemmed words.

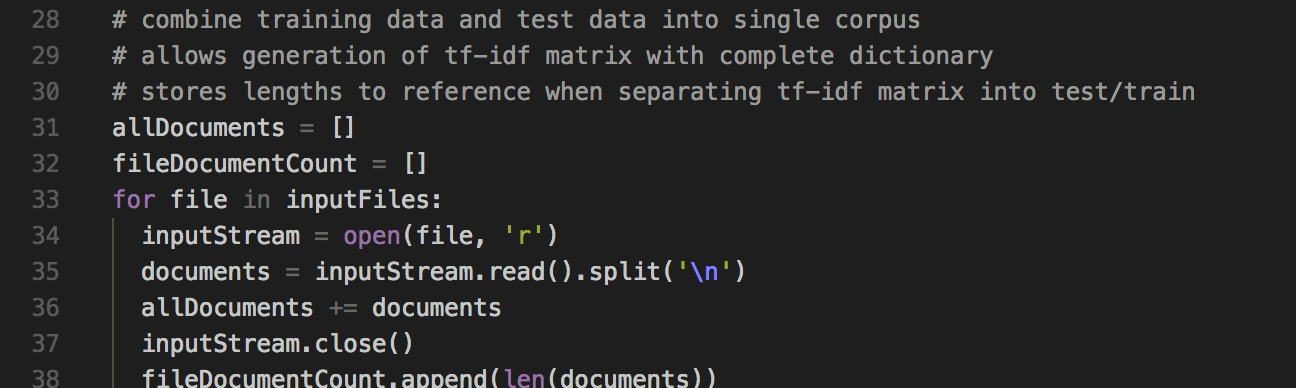
The LibSVM executables require that the data be in the form of:

*label idx:value idx:value … \n*

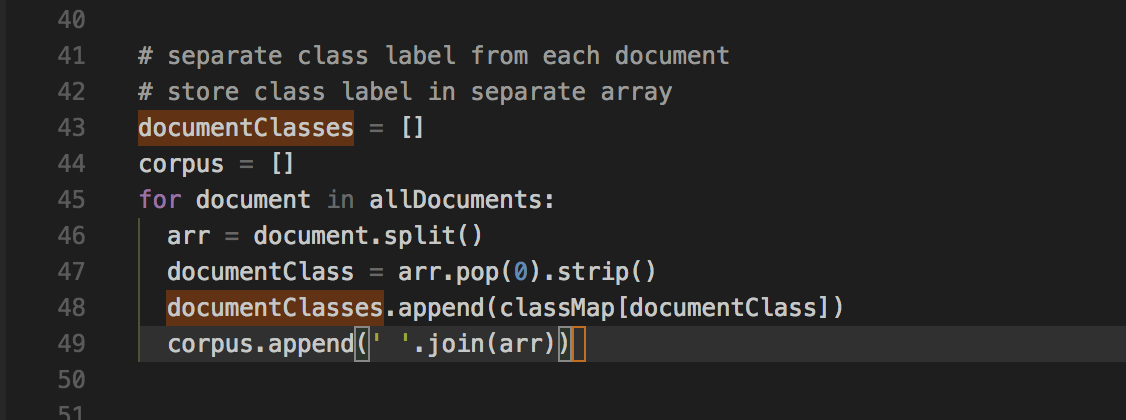
Where every document is separated by new line, starts with the document’s label, followed by a sequence of (index, value) pairs. For this assignment, TF-IDF values are used for the values. Therefore, the index refers to the terms index in the TF-IDF matrix, and the value refers to that terms TF-IDF value.

To achieve this format, a python script (prepare-data.py) was defined to read WebKB input files, calculate TF-IDF values, and write documents to output files in the form required by LibSVM.

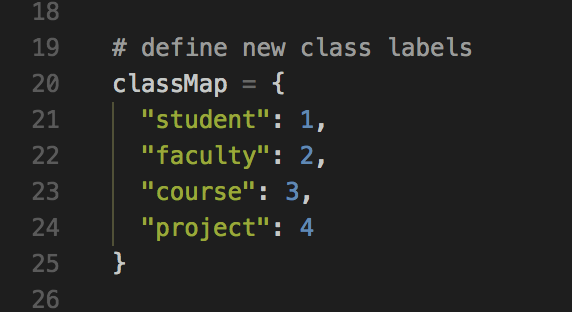
First, the prepare-data python script reads the WebKB training and test data sets and constructs a list of all documents found in both training and test data. The data is combined prior to calculating TF-IDF scores to ensure that all vocabulary is included in the TF-IDF calculations. The length of training and test data sets is recorded, as it will be required to split the data back into two sets.



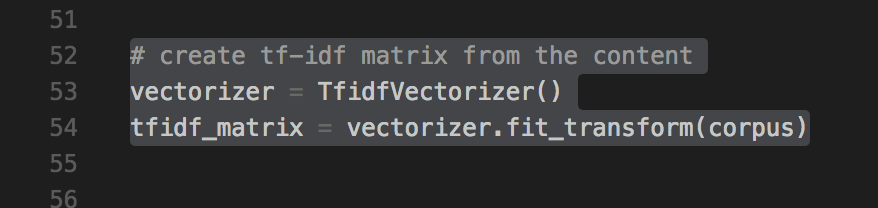
The class labels are removed from the documents and stored in a separate list, because they should not be included in the TF-IDF calculations and they will need to be referenced at a later time.



During construction of the documentClasses list, the document labels are mapped to integer values to comply with LibSVM format.

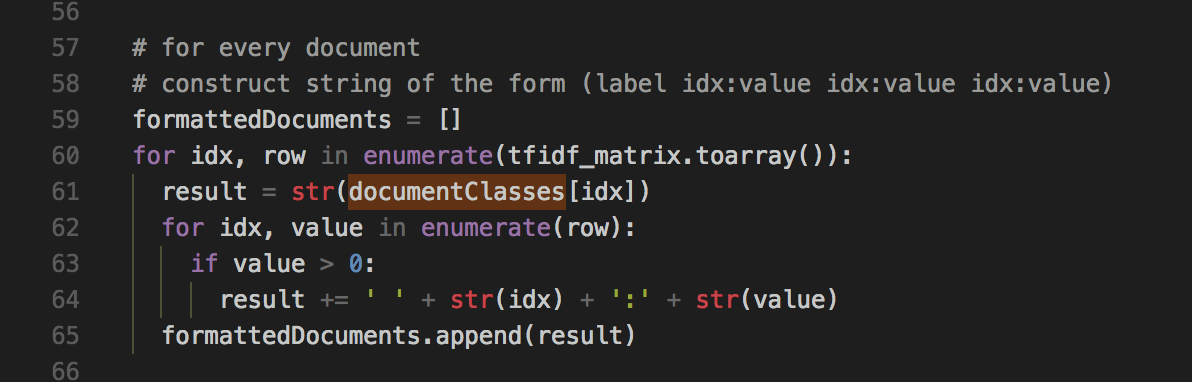


Next, Scikit-learn’s TfidfVectorizer is used to construct the TF-IDF matrix using the complete data set (training and test data combined).

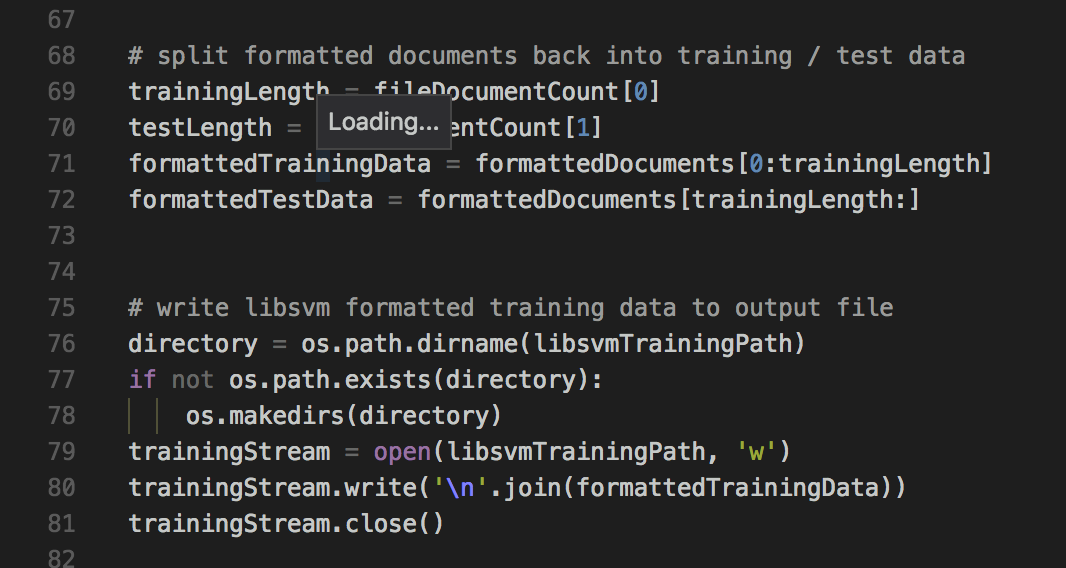


Now that we have the document - TF-IDF matrix, we have all information required to begin writing the documents to their LibSVM compatible files.

A list of LibSVM formatted documents is created by combining the class label with the (index:tf-idf) pairs from the TF-IDF matrix.

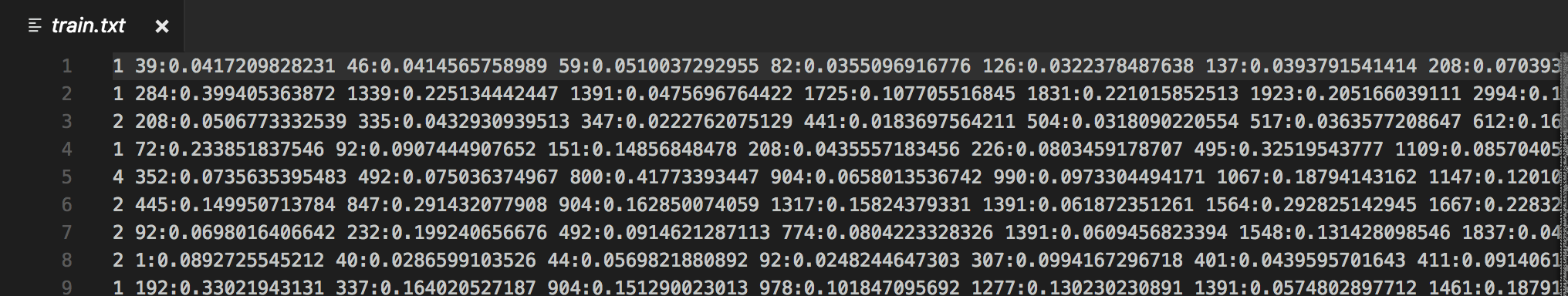


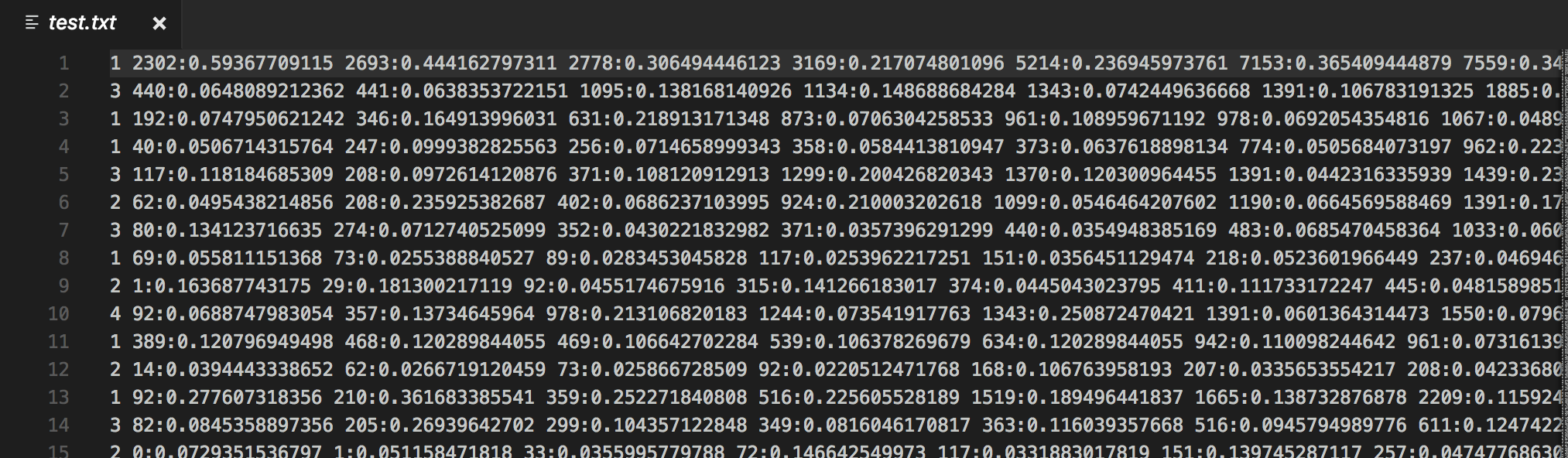
The formattedDocuments list now contains a list of all documents, both test and training sets, in the form required by LibSVM. All that is left to do is to partition the formattedDocuments back into two sets, training data and test data.



Now that the data is partitioned back into the original training and test sets and written to output files, these files can be supplied to LibSVM to conduct classification.

Below is a preview of the LibSVM formatted data sets, train.txt and test.txt:



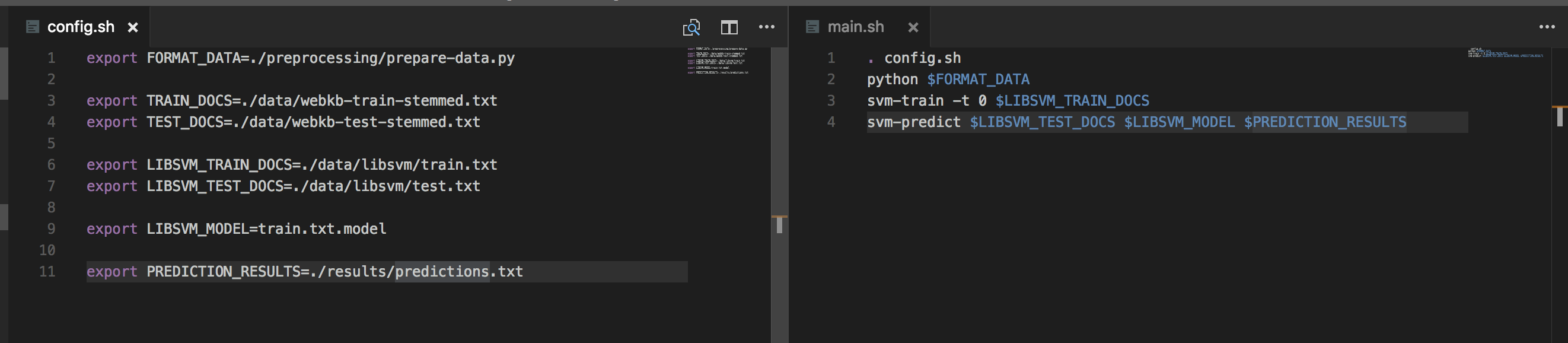


**Text Classification with LibSVM Linear Kernel**

Since LibSVM can be run from the terminal, a shell script (main.sh) has been written to automate all steps in the text classification process.

A config.sh (left) file was created to improve re-usability. The config.sh file defines the locations of the data formatting python script, the input WebKB data sets, paths to store the LibSVM-formatted files once converted, and the path to write the predicton results to.

The main.sh (right) file utilizes the config file to complete data formatting, model training, and test data evaluation. The entire process from data preparation to results is automated with the main.sh script.

****

main.sh execution explanation:

1. Load configuration from config.sh

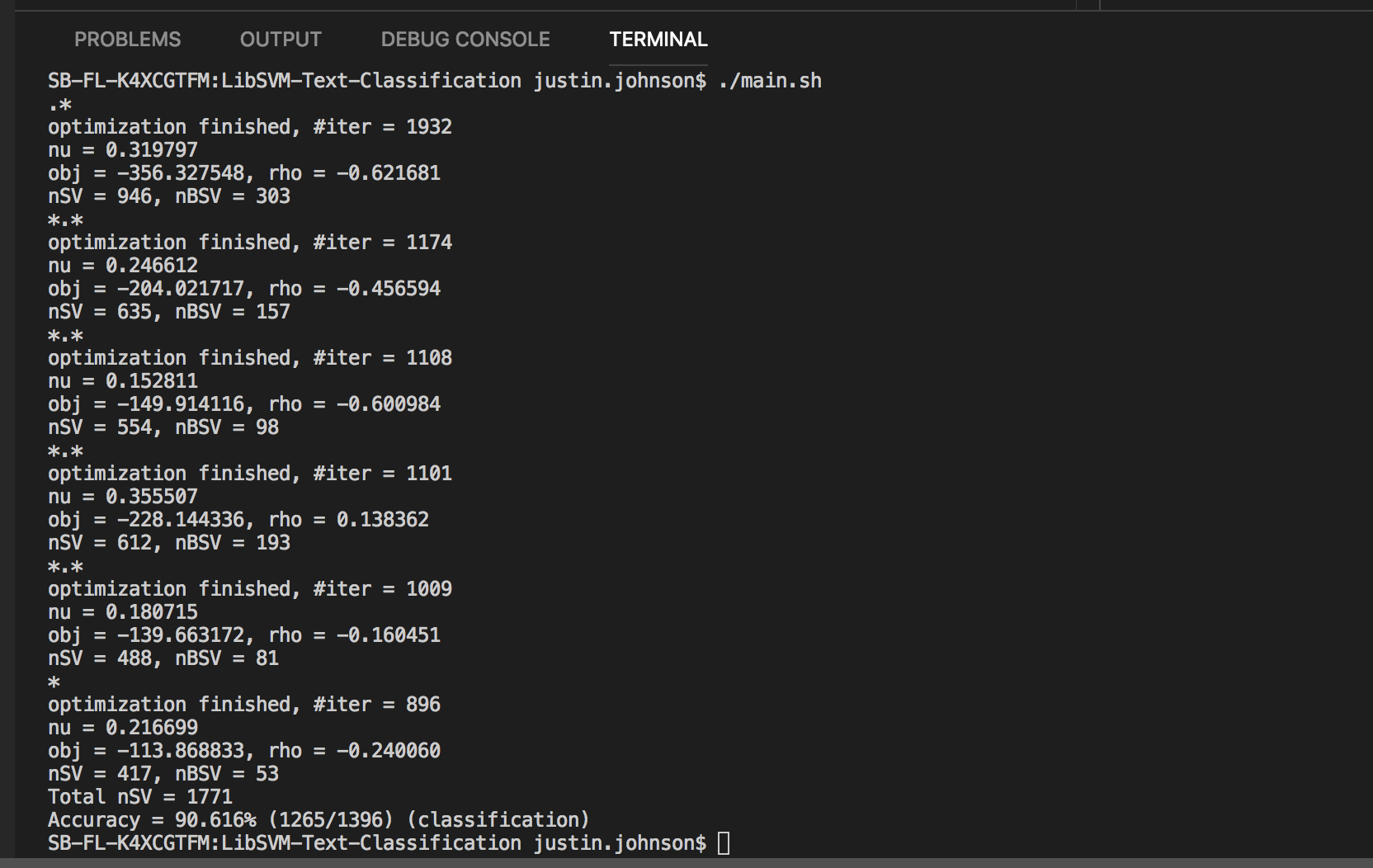
2. Execute python data formatting script, responsible for converting WebKB data into LibSVM format

3. Train Linear SVM model using LibSVM (-t 0 is required option for linear kernel)

4. Use model from 3 to make predictions on test data, writing results to output file and printing accuracy to console

**Results**

Execution of main.sh produces the following terminal output:



The LibSVM classifier with linear kernel correctly classified 90.616% of the test data instances. Additionally, the svm-predict executable wrote the predicted class values to the output file that was defined in config.sh. The labels are still in their integer format but can be easily converted back into the original string labels using the mapping defined earlier. A preview of first 16 predictions below:

