**Bayesian Learning for Text Classification:**

**How to Run:**

1. Download the Zip file and unzip all the contents.
2. Change the path of the dataset according to the new folder directory.
3. Run the ‘textclassifier.py’.
4. Find the results in the console, i.e. Accuracy rate.

**Implementation:**

Steps performed to reach the accuracy rate of 86.6 with the training set of 500.

1. Loading the Data: After the data is unzipped from the .Tar format, it is loaded using the ‘os’ API in Python.
2. There are various methods like:

* retrieveFile()
* cleanData()
* Calculate\_probab()

1. In the CleanData(), function, the data is cleaned using the Remove\_List and the Replace\_list lists. All the unwanted characters are removed or replaced. Also, the data is lowercased using the data.lower().
2. In the retrieveFile(), function, using the random.randint() library, we can get the random index of the folders and the files in the dataset and after pre-processing can remove that file from consideration to prevent re-processing of the same file. Read the file, using data.read().
3. Calculateprobab(), function, needs the parameters 🡪 words and wordcount, and calculate the probability using,

math.log(float(value)/float(sum\_)).

Now, the main execution of the program,

We will take the training set = 500 and start loading the training data.

In the ‘myFile’, clean the data after opening the file which was selected using random.randint.

Using the dictionaries, word\_count and total\_word\_count and file\_name, we will get the word count of the file and word count of the folders and keys of the folders respectively.

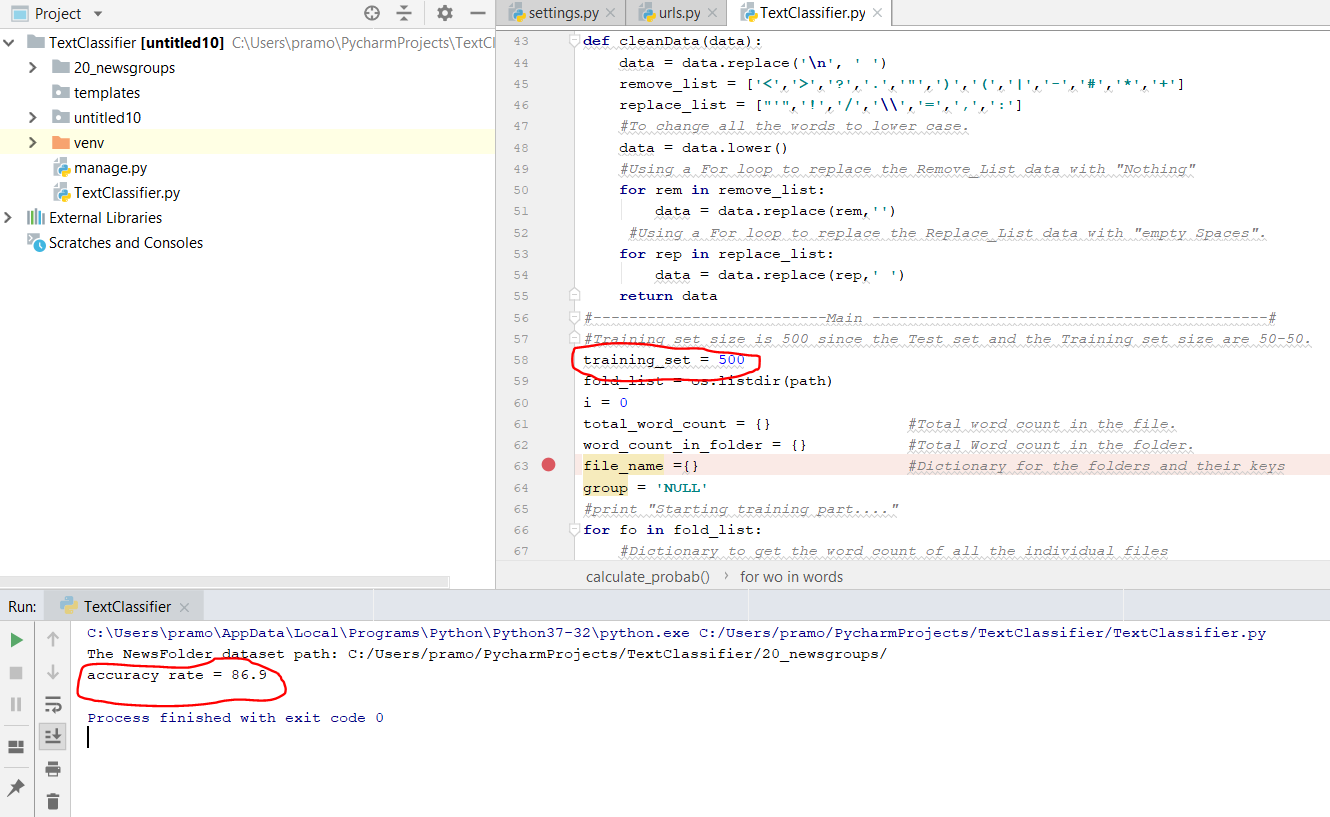
On every iteration, we will check if the word is existing or not in the dictionary and accordingly mark the new entry or increment to the existing data.

Now, we will move to the Test data, where we will have accuracy set to Zero and iterate the data just like the training set.

In the end, we will calculate the probability of the individual files and folders and append them comparing the maximum probability obtained and consider it for the accuracy of the prediction.

**Results:**

Training set = 500; Accuracy Rate = 86.9



Training set: 100; Accuracy: 74.2

