CSEE5590/490:

Python and Deep Learning Programming

(2018 Fall)

*Python Lab Assignment 2*

Submitted On:

10 October, 2018

Submitted By:

Name: Farid Uddin Ahmed

Class ID: 02

Name: Zarin Tasnim Sandhie

Class ID: 26



LIST OF THE DOCUMENTATIONS:

1. Author
2. Objective
3. Features
4. Configuration
5. Input/ Output Screenshots
6. Implementation & Code Snippet
7. Evaluation & Explanation
8. Conclusion
9. Links
10. References

AUTHORS

This report contains all the documents for the of Lab Assignment #1. The assignment was done by Farid Uddin Ahmed (Class ID 2) and Zarin Tasnim Sandhie (Class ID 26), both are graduate student majoring in Electrical Engineering Department at University of Missouri Kansas City (UMKC).

OBJECTIVE

In the last three weeks of Python/deep learning class, we got to work with the python and machine learning functions. The definition and working principle of the following topics were taught in the class:

* Machine learning
* Regression
* Clustering
* Scientific Packages
* Classification
* Support Vector Machines (SVM)
* Natural Language Processing (NLP)
* Natural Language Toolkit (NLTK)
* Corpora, WordNet, Tokenization, Stemming, Lemmatization.
* Named Entity Recognition (NER)
* N-gram and finding the frequency of n-gram.

This assignment includes all of the things mentioned above.

* The dataset chosen for problem 1, 2 and 4 are “digits” dataset. This dataset contains 0-9 digits in the target data. And around 1700 data which have to be categorized in the target data.
* The first problem deals with the use of built-in dataset in the seaborn library and the creation of prediction model based on Naïve Bayes Classification.
* The second problem deals with the use of Support Vector Machine (SVM) classification using different kernel.
* For solving the third problem, lemmatization, bigram, Counter etc. were used.
* The fourth problem works on the basis of k nearest neighbor algorithm for classification. And also observes the effect of changing K-value on the accuracy
* The evaluation section contains all the explanation asked in the problem.

FEATURES

The features of all the problems are discussed below:

Problem 1:

Pick any dataset from the dataset sheet in the class sheet

* Plot how many of each category is available in your dataset
* Create one prediction model based on Naïve Bayes Classification and evaluate your model

Problem 2:

Implement Support Vector Machine classification:

* Apply SVC with kernel “poly” degree =4
* Apply SVC with “rbf” kernel.
* Change gamma and C parameters in the model to see how the result may change.
* Report the accuracy of the model on both models separately and with which parameters you got better result.

Problem 3:

Write a program in which take an Input file. Use the simple approach below to summarize a text file:

* Read a file
* Apply lemmatization on the words
* Apply the bigram on the text
* Calculate the word frequency (bi-gram frequency) of the words (bi-grams)
* Choose top five bi-grams that have been repeated most
* Go through the original text that you had in the file
* Find all the sentences with those most repeated bi-grams
* Extract those sentences and concatenate

Problem 4:

Report your views on the k nearest neighbor algorithm when we change the K how it will affect the accuracy. Provide a good justification for the changes of the accuracy when we change the amount of K. For example: compare the accuracy when K=1 and K is a big number like 50, why the accuracy will change.

CONFIGURATION

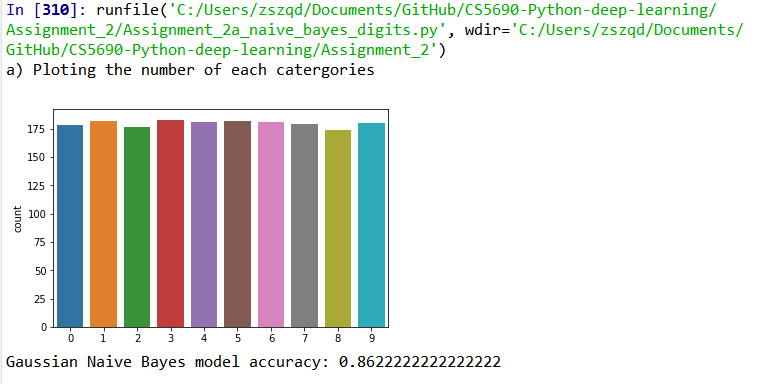
For executing the assignment, coding was done with Python software version 3.6. The simulation was in software: Anaconda (Spider).

INPUT/OUTPUT SCREENSHOTS

Problem 1:

* The “digits” datasets in the seaborn library is taken as input dataset.
* When simulated, the output plot shows the number of each digits in the dataset.
* It also shows the accuracy of the Naïve-Bayes classification.

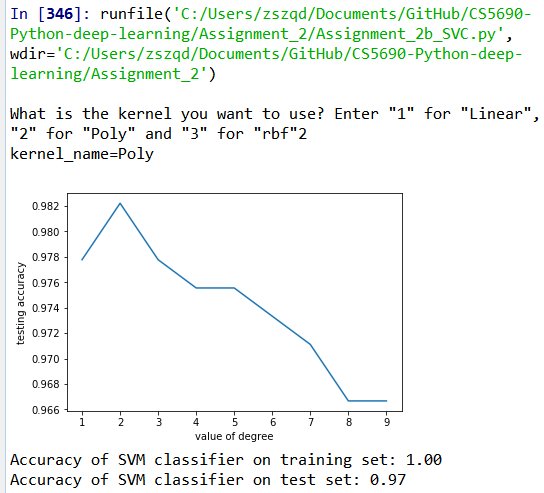
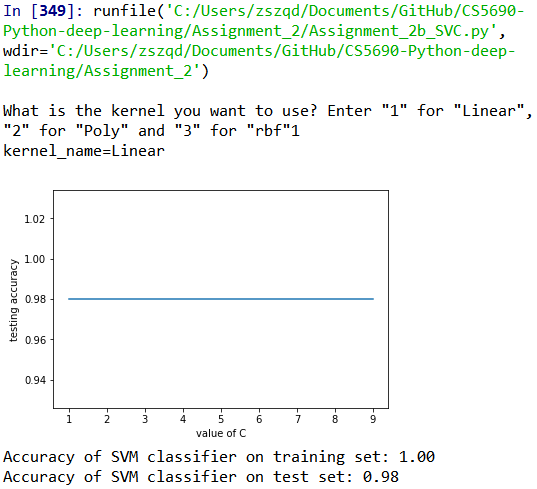
Output:

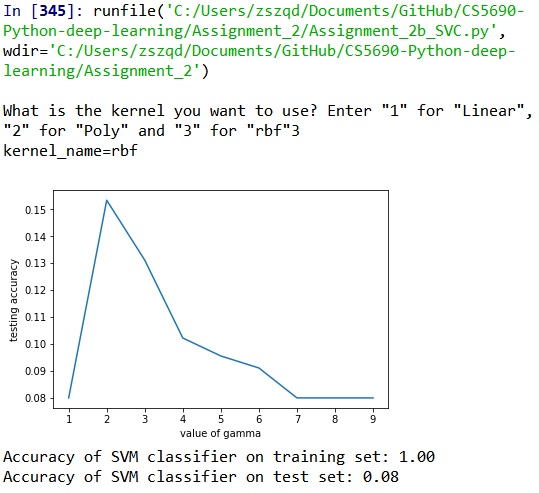


Problem 2:

* It works on “digits” datasets.
* And applies SVC (Support Vector Classification) on the given dataset.
* The type of kernel on the SVC can be specified by giving input.
* Upon compilation, the code asks for the type of kernel that the user wants to apply.
* For linear kernel, it shows the accuracy of the model for C = 1, and it also varies the value of C within the range 0 to 10 and shows the accuracy.
* For Poly kernel, it shows the accuracy of the model for degree = 4, and it also varies the value of degree within the range 0 to 10 and shows the accuracy.
* For rbf kernel, it shows the accuracy of the model for gamma = 1, and it also varies the value of gamma within the range 0 to 10 and shows the accuracy.

Output:

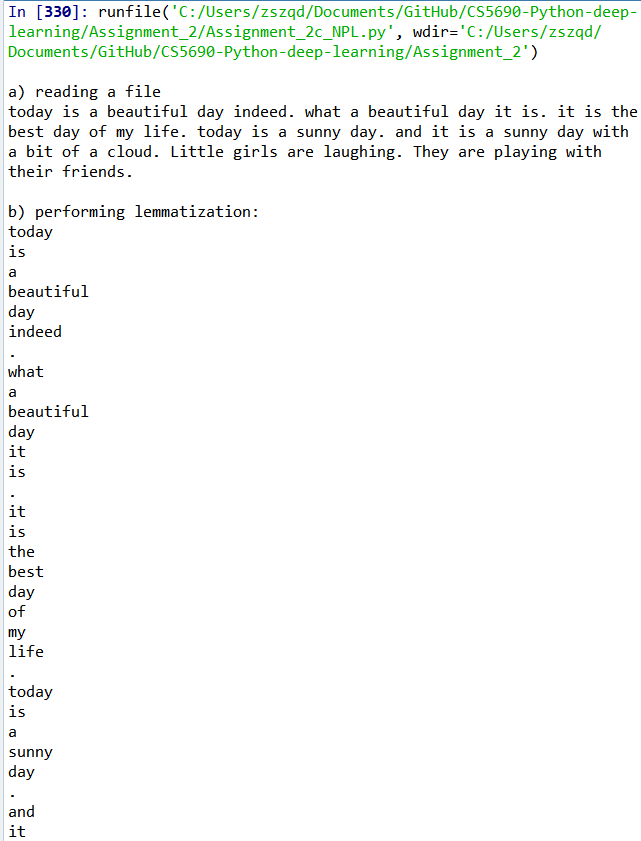
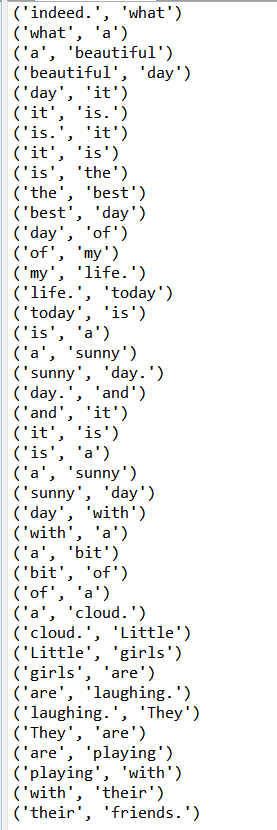
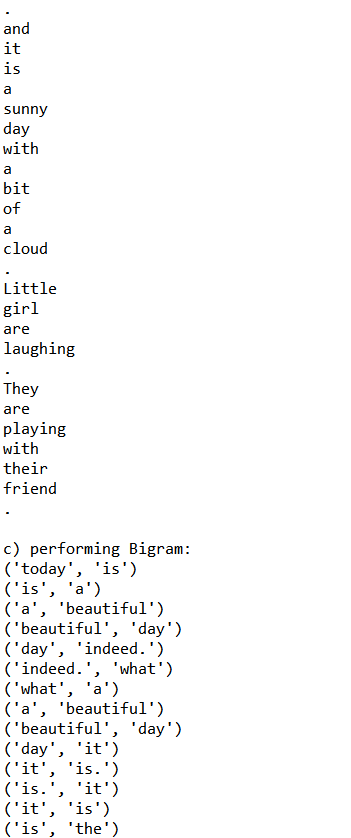


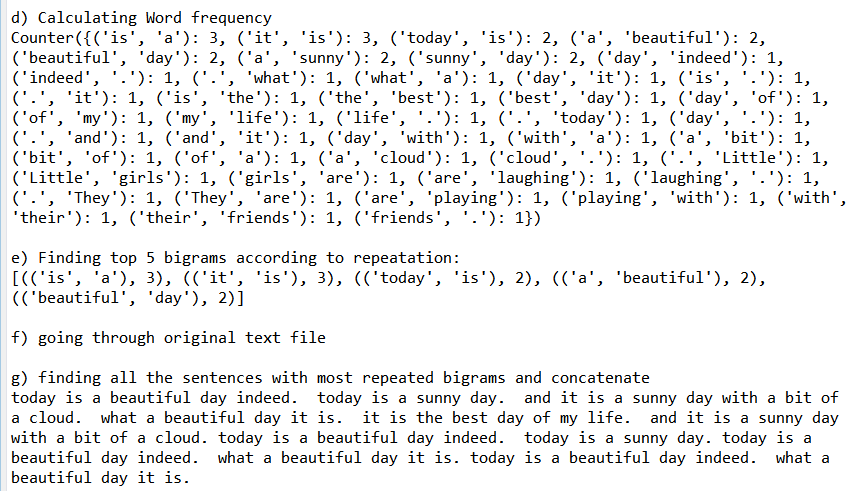


Problem 3:

* An input file “big.txt” is given to read which contains some repetitive word to observe the different operations of the problem.
* After reading the file, lemmatization, bigram, word frequency calculation of the bigram is done.
* Later on, the top five most repetitive bigrams are found and the sentences containing those bigrams in the original files are concatenated.

Output:

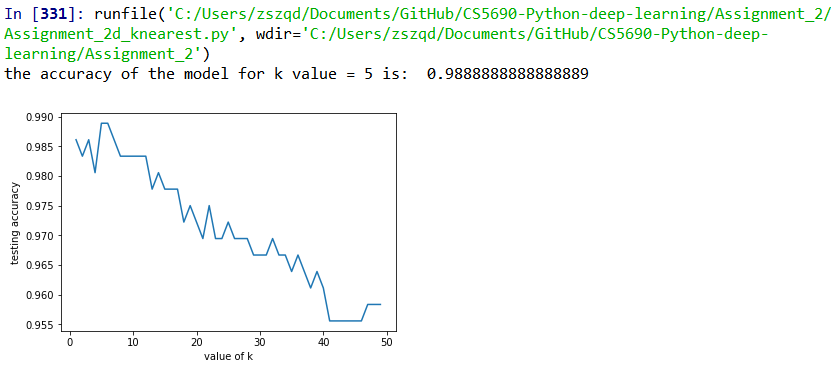
 



Problem 4:

* “digits” dataset is taken as input.
* K-neighbor classification is done on the dataset for the neighbor value = 5.
* The accuracy is observed.
* The accuracy is also observed with the range of neighbor value from 0 to 50.

Output:

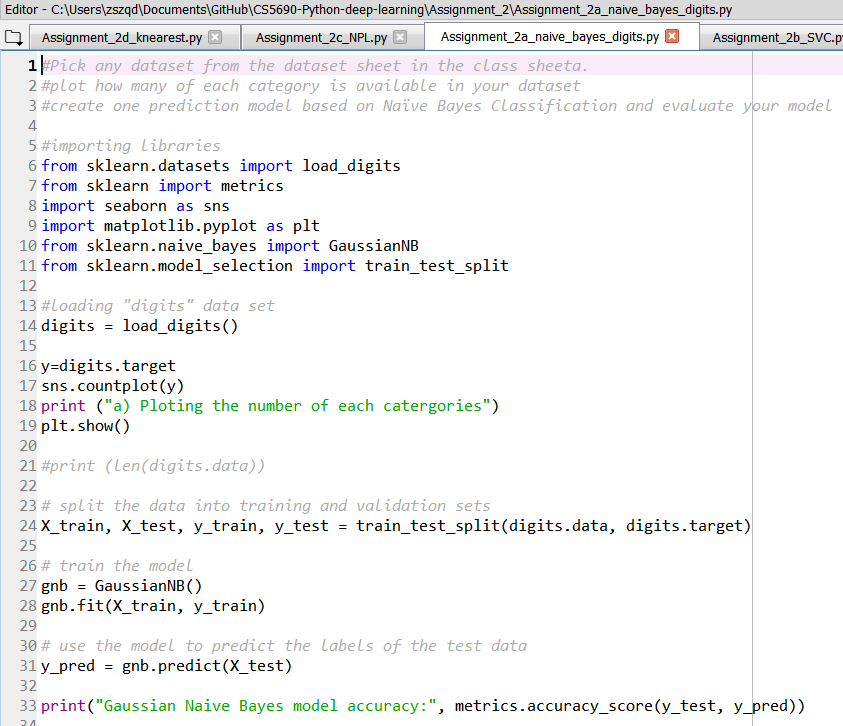


IMPLEMENTATION & CODE SNIPPET

Problem 1:

* The code takes “digits” dataset as input dataset.
* sns.countplot is used to show the number of elements in each category.
* The dataset is split into testing and training dataset.
* Naïve Bayes classification is used on the training dataset.
* Then based on the training dataset, prediction is done on testing dataset.
* Accuracy is calculated based on the testing dataset.

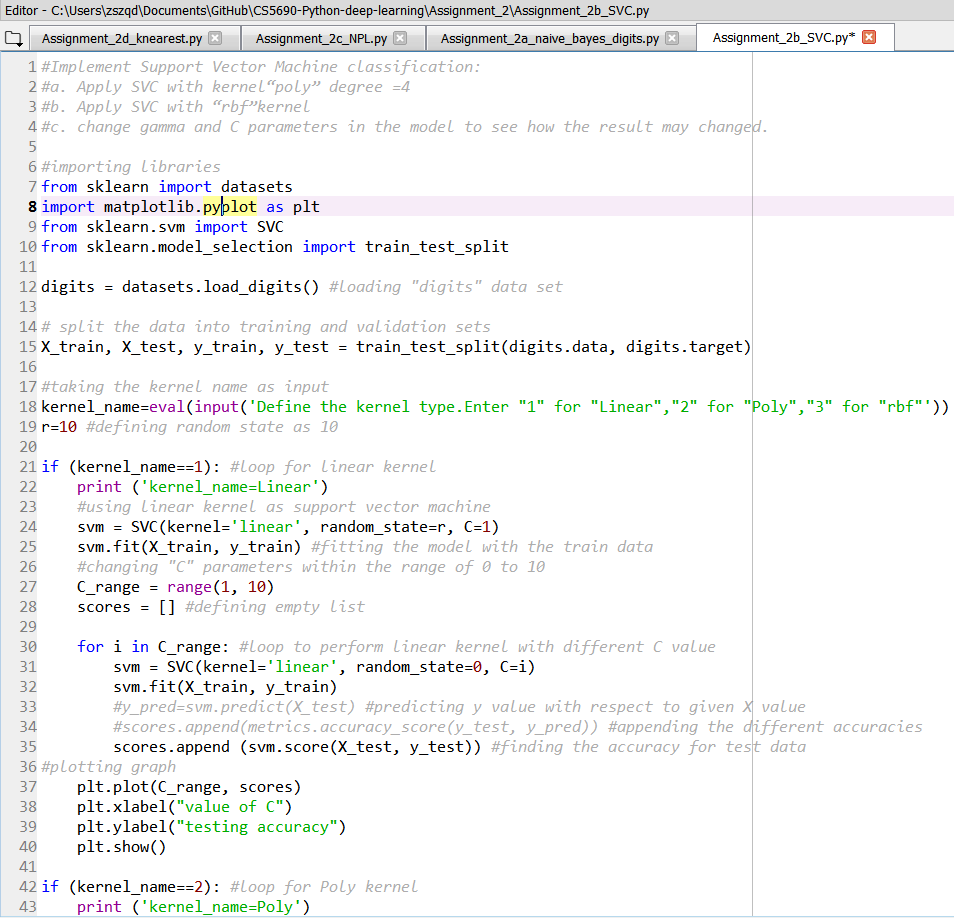
Code:

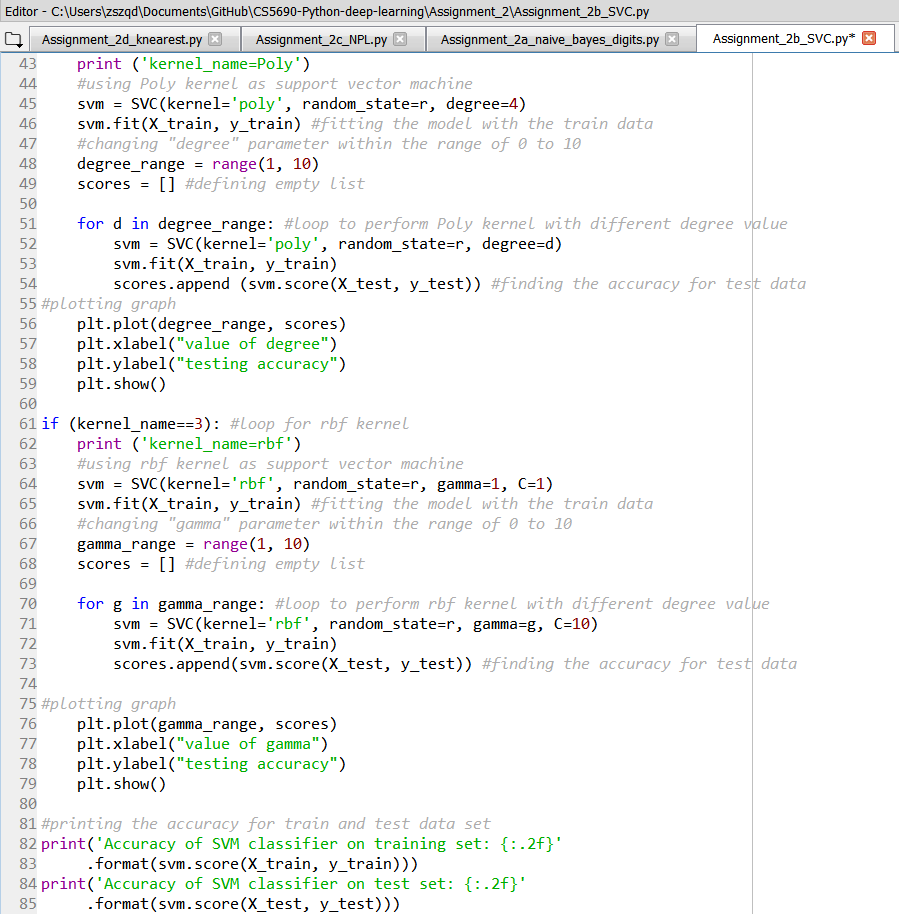


Problem 2:

* The code takes “digits” dataset as input dataset.
* The dataset is split into testing and training dataset.
* Kernel name is taken as input
* For input = 1, SVM with linear kernel is done and accuracy is found out for C = 1.
* Then inside a “for loop”, the value of C is changed from 0 to 10 and the accuracy is found out and it is plotted in a graph.
* For input = 2, SVM with poly kernel is done and accuracy is found out for degree = 4.
* Then inside a “for loop”, the value of degree is changed from 0 to 10 and the accuracy is found out and it is plotted in a graph.
* For input = 3, SVM with rbf kernel is done and accuracy is found out for gamma = 1.
* Then inside a “for loop”, the value of gamma is changed from 0 to 10 and the accuracy is found out and it is plotted in a graph.

Code:

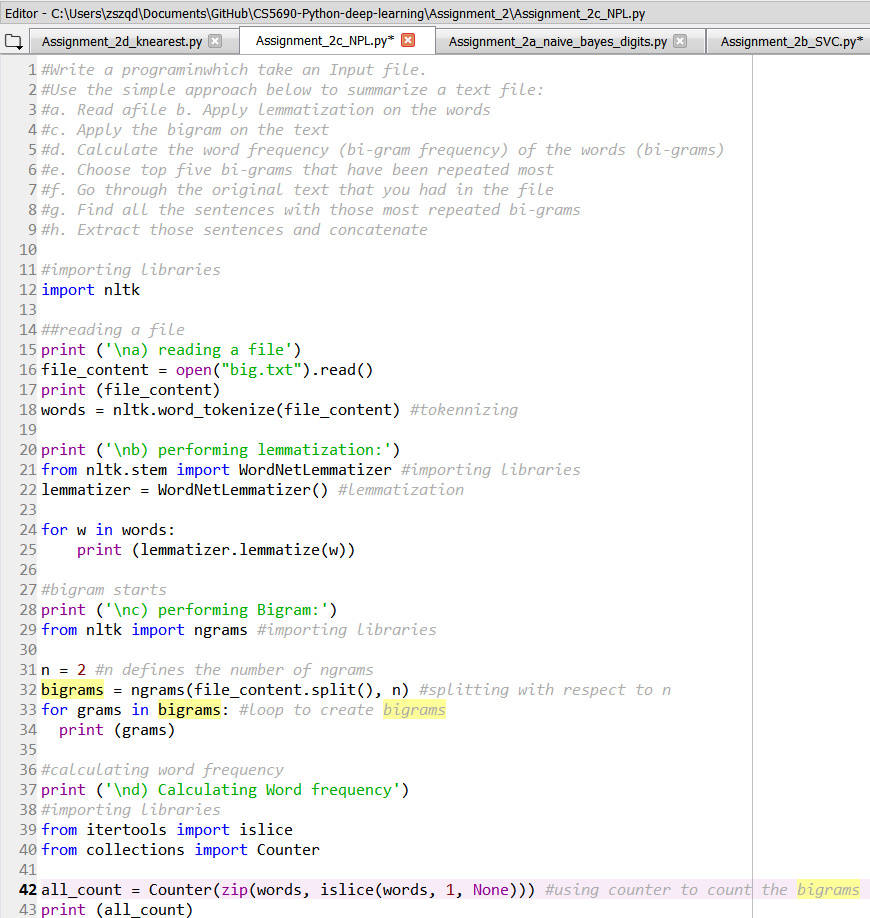


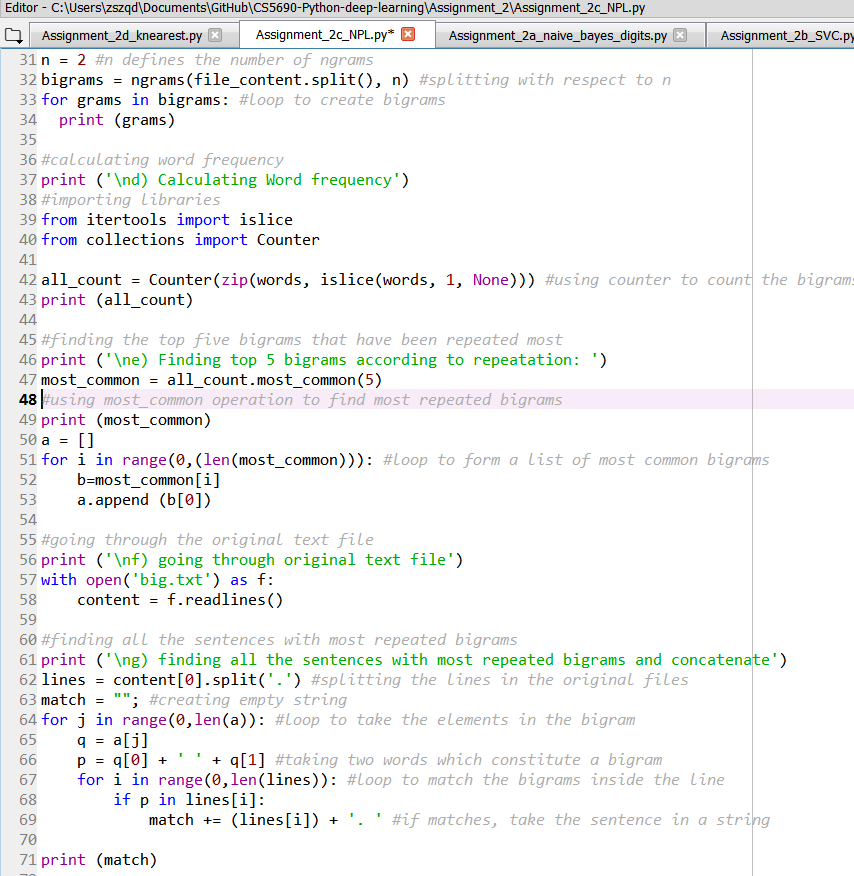


Problem 3:

* A file named “big.txt” is taken as input file.
* The words in the file is tokenized and then lemmatized accordingly.
* Then bigram is done on the words of the file.
* After that, the frequency of the bigrams is found out using the built in function “Counter”.
* Then the 5 most repeated bigrams are found out by the function “most\_common”.
* Then the 5 most common bigrams are taken in a list.
* Then inside a loop, the bigrams are taken as a set separately.
* After that, another loop is used to match the bigrams with the words of the original file.
* If it found any matches, the sentences containing the matched bigrams are taken and concatenated.

Code:

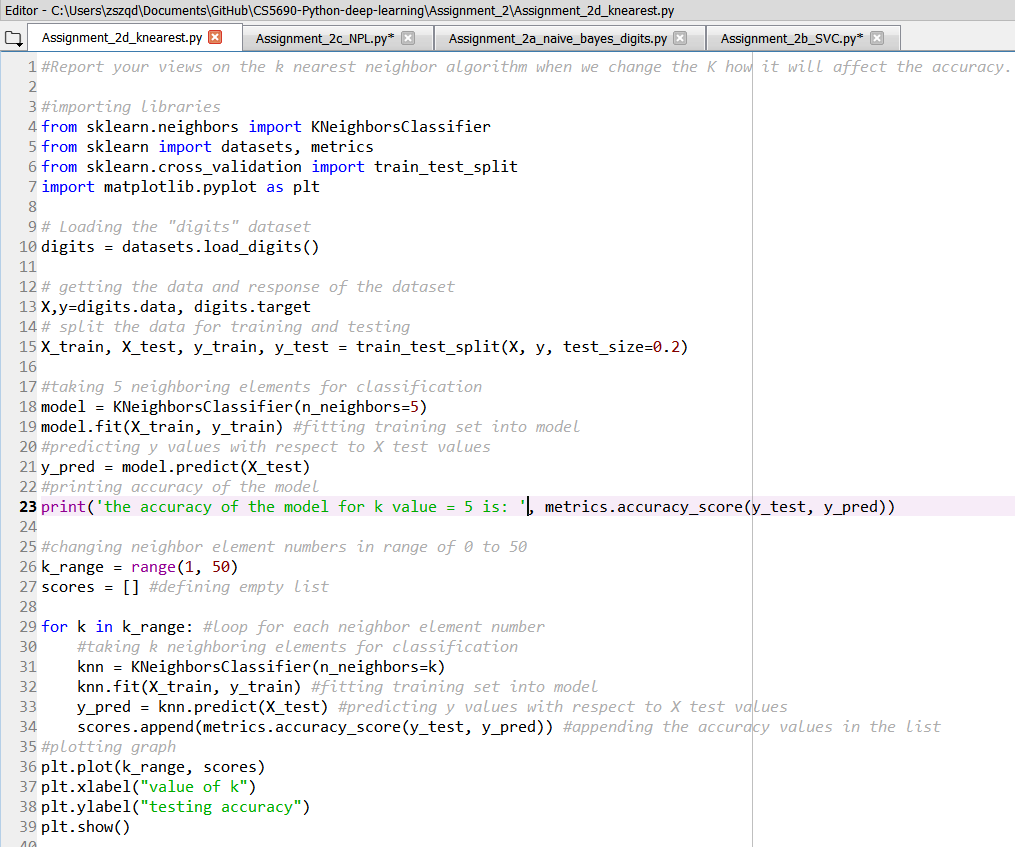




Problem 4:

* The code takes “digits” dataset as input dataset.
* The dataset is split into testing and training dataset.
* Then K-neighbor classification is done with the neighbor number = 5.
* The training set is fitted into the classification.
* Prediction is done on the test set.
* The accuracy of the model is found out.
* Then inside a loop, the value of k (neighbor number) is varied from 0 to 10 and accuracy is found out.
* The results are shown in a plot.

Code:



EVALUATION & EXPLANATION:

Problem 1:

The accuracy found in the Naïve Bayes classification with digits dataset is around 80%. And it shows the number of each digits in the dataset. The total number of data is around 1700. And the number of each digit is around 170.

Problem 2:

This problem works on different types of kernel.

For linear kernel, the accuracy on the test set is around 98%. Though it vary slightly for different case. The value of C parameter was changed from 0 to 10. But the results were quite similar.

For poly kernel, the accuracy is in between 97%-99%. By changing the value of degree parameter, the change in accuracy can be observed. The accuracy rises at first with the increase of degree value and then starts to decrease. This decrement continues for a large value of degree (20).

The accuracy on test set is lowest with rbf kernel. The accuracy in the range of 10% for rbf kernel. By changing the value of gamma parameter within the range of 0 to 10, the accuracy increases slightly and then decreases.

Problem 3:

The operation of the code is as intended in the problem specification. All the operation mentioned in the problem was done sequentially. Any file can be taken as input. For the sake of easy understanding of all the operations happening in the code, a simple file with lots of repetition was chose.

Problem 4:

The objective of this problem was to review the operation of k-neighbor classification and observe the effects of changing the neighbor number within the range of 0 to 50. For K = 5, the accuracy is around 98%. With the increase of K, the accuracy has a downward slope in general. The optimum range of accuracy is for the value of K = 1~10. If the K value is too big compared to the data, all the data falls under the same classification, which can result into an underfit and accuracy can decrease.

CONCLUSION

All the required problems were solved successfully during this assignment. During this assignment, we learned the use and advantages of different kinds of classification techniques. We also learned about the NLP (Natural Language Processing) and the built-in functions of NLTK (Natural Language ToolKit).

LINKS

Video Link:

<https://www.youtube.com/watch?v=oy09gVxFOKo&feature=youtu.be>

Github Link:

<https://github.com/Sandhie177/CS5690-Python-deep-learning/tree/master/Assignment_2>

REFERENCES

1. <https://github.com/wade12/WikiScraper/blob/master/>
2. <https://www.w3resource.com/python-exercises/>
3. <https://www.learnpython.org/>
4. <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>