NLP Assignment 2
CSC74040

Name: Sagar Ghimire

1. [Movie review classification using NaÅNıve Bayes - 10 points]
Assume that you have trained a NaÅNıve Bayes classifier for the task of sentiment classification (please refer to Chapter 4 in the J&M book). The classifier uses only bag-of-word features. Assume the following parameters for each word being part of a positive or negative movie review, and the prior probabilities are 0.4 for the positive class and 0.6 for the negative class.

	pos	neg
I	0.09	0.16
always	0.07	0.06
like	0.29	0.06
foreign	0.04	0.15
films	0.08	0.11

Question: What class will Naive Bayes assign to the sentence “I always like foreign films”? Show your work.

Answer//

The prior probabilities are.

P(pos class) = 0.4
P(neg class) = 0.6

P(“I always like foreign films” | +) = P(pos class) x {P(word_1 | +) x….x P(word_n | +)}
(multiply probabilities of positive words)

= 0.4 x 0.09 x 0.07 x 0.29 x 0.04 x 0.08
= 0.00000234 (lower)

P(“I always like foreign films” | -) = P(neg class) x {P(word | -) x………x P(word_n | -)}
(multiply probabilities of negative words)

= 0.6 x 0.16 x 0.06 x 0.06 x 0.15 x 0.11
= 0.0000057 (higher)

Therefore, the sentence will be assigned to a negative class.

Question Number 2

a) Implement in Python a NaÅNıve Bayes classifier with bag-of-word (BOW) features and Add-one smoothing. Note: Do not use smoothing for the prior parameters. You should implement the algorithm from scratch and should not use off-the-shelf software.

Answer//

The code is in notebook which is ***NB. Ipynb***. NB codes for both large and small corpus are same besides some small changes as per required. All the steps are commented in the notebook. Class NB and methods are introduced to preprocessed data obtained from ***pre-process.ipynb.***

For small corpus I have manually assigned the training data and test data and vocabulary in the notebook instead of making a file in local computer and calling it. For the larger corpus all train test and vocab data are imported using the path in local computer.

b) Use the following small corpus of movie reviews to train your classifier. Save the parameters of your model in a file called movie-review-small.NB (you can manually convert this small corpus into the vector format, so that you can run NB.py on it). [10 points]

i. fun, couple, love, love comedy
ii. fast, furious, shoot action
iii. couple, fly, fast, fun, fun comedy
iv. furious, shoot, shoot, fun action
v. fly, fast, shoot, love action

Answer//

In this part I have taken same NB codes but slightly edited for this small corpus and I have added the code noetbook as small_corpus.ipynb.

The feature vector has been saved as movie_review_small.NB

```
{"comedy": {"fun": 1, "couple": 1, "love": 2}}
{"action": {"fast": 1, "furious": 1, "shoot": 1}}
{"comedy": {"couple": 1, "fly": 1, "fast": 1, "fun": 2}}
{"action": {"furious": 1, "shoot": 2, "fun": 1}}
{"action": {"fly": 1, "fast": 1, "shoot": 1, "love": 1}}
```

Class counts: {'comedy': 2, 'action': 3}

Total samples: 5
Prior_prob_comedy= 2/5=0.4
Prior_prob_action= 3/5= 0.6

Log prior probabilities: {'comedy': -0.916290731874155,
'action': -0.5108256237659907}

c) Test you classifier on the new document below: {fast, couple, shoot, fly}. Compute the most likely class. Report the probabilities for each class. [5 Points

Answer//
test_features = {'fast': 1, 'couple': 1, 'shoot': 1, 'fly': 1}

Log probabilities for test data:
Probability of class comedy : -9.52173897104528
Probability of class action : -8.705062601975643

Predicted class for test data: action

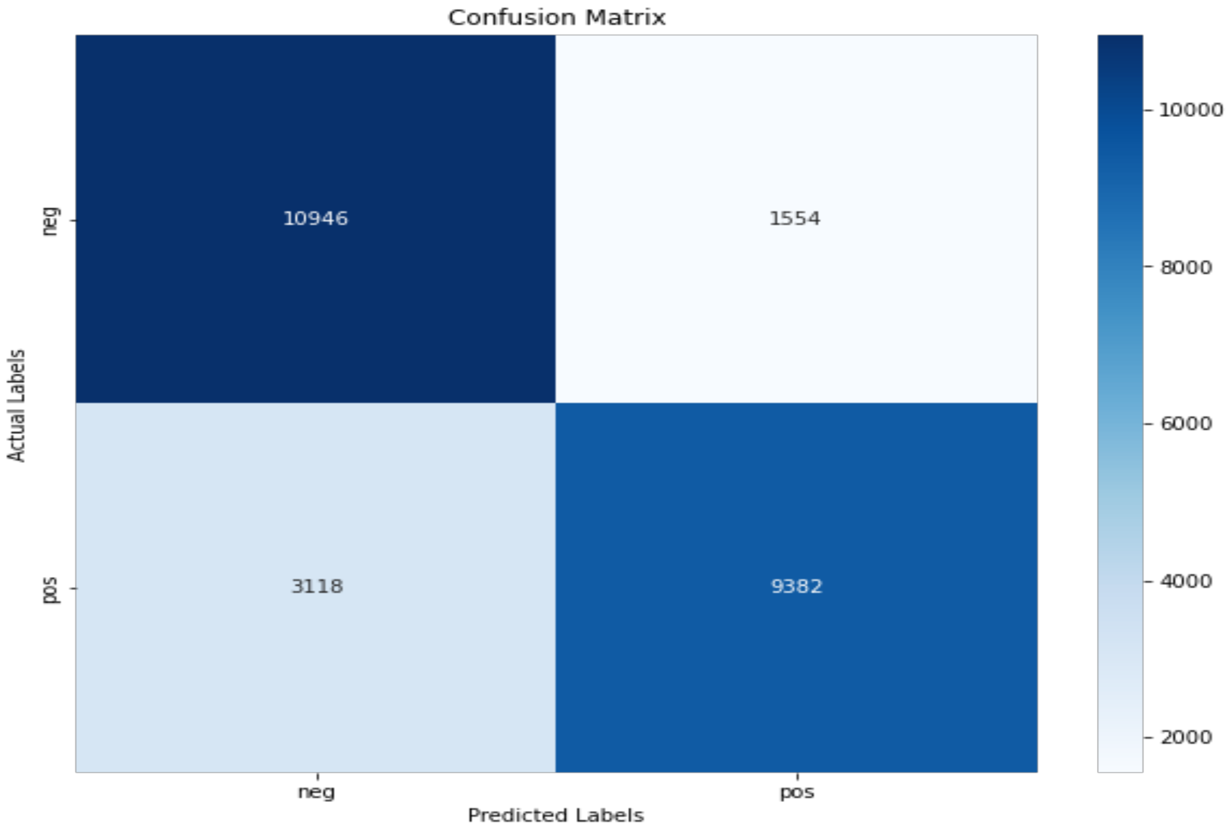
Accuracy: 1.0
The log probability for class comedy is -9.52
The log probability for class action is -8.7
Therefore, the test document is of *action class* as it has the highest values.

d) Now use the movie review dataset provided with this homework to train a Naive Bayes classifier for the real task. You will train your classifier on the training data and will test it on the test data. The dataset contains movie reviews; each review is saved as a separate file in the folder “neg” or “pos” (which are located in “train” and “test” folders, respectively). You should use these raw files and represent each review using a vector of bag-of-word features, where each feature corresponds to a word from the vocabulary file (also provided), and the value of the feature is the count of that word in the review file. Pre-processing: prior to building feature vectors, you should separate punctuation from words and lowercase the words in the reviews. You will train NB classifier on the training partition using the BOW features (use add-one

smoothing, as we did in class). You will evaluate your classifier on the test partition. In addition to BOW features, you should experiment with additional features. In that case, please provide a description of the features in your report. Save the parameters of your BOW model in a file called moviereview-BOW.NB. Report the accuracy of your program on the test data with BOW features.
Investigate your results. For the reviews for which your program made incorrect predictions, were there any trends that you observed? That is, can you explain why these incorrect predictions were made?

Answer//
The result of this test was decent as we are able to achieve 81.3 % of accuracy for the test data. This is the confusion matrix for the predicted neg and pos documents. Negative classes are predicted more than positive class in this test. Negative are predicted more than 14,064 and positive are predicted below 10,936. The accuracy is better as it is over 80%. Reason behind incorrect predictions may be because of any step missed in preprocessing . Both the classes have same prior probability of 0.5. When I tried different preprocessing methods like removing stop words, single characters and removing words with low frequency. The accuracy got lower than 55.8%. Since, the vocabulary has many punctuations like – which is attached to the words like well-beings. So, applied preprocessing according to the vocabulary.

In future the plan is to check the model again and increase the performance without any bias and get the accuracy above 90%.



```
Classification Report:
              precision    recall  f1-score   support
```

```
         neg       0.78      0.88      0.82     12500
         pos       0.86      0.75      0.80     12500

    accuracy                           0.81     25000
   macro avg       0.82      0.81      0.81     25000
weighted avg       0.82      0.81      0.81     25000


############################# THE END##################################
```