P.Srujan B19CSE063 Lab Report

Question 1:

1. The data as loaded using panda.read_csv() function

2. We plotted the count of each target using value_counts() function and then plotted is using .plot(kind='barh')

3. The unique keywords were printed using the keys of the dictionary formed from the Counter() function

4. It was plotted the same way the count of each target was as in step 2

5. The length of the tweet was calculated using for loop.

6. The correlation between the length of the tweet and target was plotted using sns.heatmap() function.

7. The null values were printed using .isna().sum()

8. They were removed using .dropna(axis = 0, how ='any') and the index was made to reset

9. The Double Spaces, Hyphens and arrows, Emojis, URL, another Non-English or special symbol were removed using re.sub() function.

10. The wrong spellings were replaced with correct ones using pyspellchecker using the SpellChecker() function and were inserted into the dataset.

11. We made a set of texts whose target is of class 0 and 1 respectively using .loc function

12. Then we imported the WordCloud() function and generated the wordcloud using the generate function and plotted it using matplotlib.pyplot

13. A new dataset was created with only the rectified Text and Target using .loc function

14. Dataset was split into train and validation using .iloc() function

15. To count the number of unique words the Counter function was used

16. We created the Term Document Matrix using CountVectorizer() which was trained and scaled using fit_transform() whose array representation was done using .toarray() function

17. The same was done for the text corresponding to classes 0 and 1

18. The frequency was calculated for classes 0 and 1 using the Counter function

19. The total frequency was calculated by adding all the frequencies of the words.

20. The probability for each word in a given class was calculated as :

P[i] = (frequency of the word I in that class)/(total frequency of that class)

21. Class wise probability was calculated by forming the dataset with target value 0 and 1 from training dataset and then divivding their individual dataset size by the total size of training dataset.

22. For Laplacian smoothing we used the following formulae:

If a word from the new sentence does not occur in the class within the training set, the equation becomes zero.
Else the normal posterior was calculated viz likelihood*prior

23. Confusion matrix was made using the confusion_matrix() function

24. Precision recall and f1 score were calculated using precision_recall_fscore_support() function

25. The accuracy was calculated using accuracy_score() function and the ROC was plotted using roc_curve() function and matplotlib

Analysis:

1. The accuracy score was 0.690748031496063. (80% train size)
2. As asked in the question

Explanation:
A word which is present in both Class 0 and Class 1 text is considered unique in both so its counted twice but when considering the whole dataset the word is counted once.
Let the Green part be 0 and the Blue part be 1 so a word that let C(i) be the count of unique words in them (i=0,1)
Sum of unique words in target 0 and 1 = C(0)+C(1)
But the Total count of unique words = C(0)+C(1)-C(0 intersection 1)
This term C(0 intersection 1) makes all the difference