Code report

Part 1:

We start by reading the CSV file, using open won’t open the file since it is an excel file, so I tried using pandas.read\_excel but it also didn’t work since it does not recognise the excel format , so we have to change the excel to csv for the file to read . Then we implement the instances into training and testing set making sure we put the rest of the code in the middle.

Part 2:

Second part is text processing , like removing the non-alphabetical words , stop words html and square brackets , creating the functions for them then adding all the functions intone to keep a bit tidier instead of applying function by function , remove char checks using a regular expression and if the regex matches it will remove the character , the regex will basically find punctuation and numbers then remove them , another function uses the same method but removes brackets in that case , stop words on the other hand uses a pre-determined list where of the word is not in the list of stop words it adds it to a list and then after that returns the joined list as one text , and the final function uses a package called BeautifulSoup and it strips the html links from the file.

Result for part 3 shown on a different page

Part 4, 5 and 6:

This part is for building , training and testing the model I decided to use the sklearn.svm model using the linear kernel SVC , this model is also known as support vector machines some of the advantages of this model is being effective in high dimension spaces , effective when samples are less than dimensions and in general is very versatile , the disadvantages however is when there is too many features and not directly giving probability estimates , support vector machine is an algorithm used in machine Learning "to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points” [1] , in the code we used the SGDClassifier in order to build the model and call it SVM then we train the model with svm.fit , running the code without this command shows this error “This SGDClassifier instance is not fitted yet. Call 'fit' with appropriate arguments before using this estimator” so even though after calling the command we do not use it will have an effect on the testing so it doesn’t work without the training part , after the model has been trained with vectorised train list and the training sentiments we are ready to use prediction (PS: I have used TFIDF vectorization in the previous task and can be seen in task 3 part in the second page ) after our predictions has been set I use a code in order to construct a report on precision , recall and f1-score and it gives us the accuracy as well (this report can be seen in the second page as well for a more details ) , but focusing on the accuracy the overall accuracy has turned to 51% , why so low ? I think it has to do with the model I used , which is why I tried to use a different model but failed to do so that model was LTSM , using count vectorization dropped the accuracy down by 1% and shuffling the test and train set also dropped the accuracy by 2% , how can improve the accuracy though ? well I though about doing stemming however stemming would lower the accuracy “Overstemming lowers precision and understemming lowers recall. So, since no stemming at all means no over- but max understemming errors, you have a low recall there and a high precision” [2] which is why lemmatization would be a better option since it fixes the grammatical errors that come with stemming , the only other way to put up the accuracy is using a better model than SVM.

part 3:

I have coded the program to find some features in the dataset, here is the result:

1 average length: before text processing: 231.15694

after text processing: 122.79582

punctuation:

[('.', 70), (',', 58), ("'", 35), ('<', 22), ('/', 22), ('>', 22), ('"', 16), ('(', 8), (')', 8), ('-', 8), (':', 2), ('!', 1), ('?', 1), ('&', 1)]

manual testing for punctuation shows the punctuation is not accurate so keeping that in mind I am unsure the result or how to fix it

review: count: 50000

unique: 49582

top: Loved today's show!!! It was a variety and not...

freq: 5

Name: review, dtype: object

Sentiment: positive: 25000

negative: 25000

Name: sentiment, dtype: int6

TFIDF: train TFIDF: (40000, 6629081)

test TFIDF: (10000, 6629081)

--------------------------------------------------------------------------------------------------------------------------------------

Part 5:

PR = precision, Re = Recall and F1 = f1-score

PR RE F1 support

positive 1.00 0.02 0.04 4993

negative 0.51 1.00 0.67 5007

accuracy NA NA 0.51 10000

macro avg 0.75 0.51 0.36 10000

weighted avg 0.75 0.51 0.36 1000

references :

[1]"Support Vector Machine — Introduction to Machine Learning Algorithms", *Medium*, 2022. [Online]. [Accessed: 01- Jan- 2022].

[2]D. classification?, "Does stemming harm precision in text classification?", *Stack Overflow*, 2022. [Online]. [Accessed: 01- Jan- 2022].

[3]"1.4. Support Vector Machines", *scikit-learn*, 2022. [Online]. [Accessed: 01- Jan- 2022].