Assignment 3

Due on 27.04.2025 (23:00)
Programming Language: Python 3.9

Introduction

People are always in need of the others experiences as two heads are better than one, you can learn from other people's faults and gain the wisdom about it without suffering from the pain of the fault itself via experiences of the others, one of the experience transfer method is giving/reading reviews about the experiences itself, for example, people can leave reviews about the products that they bought online for recommending it or criticizing it so that other people can have knowledge about the product and moreover it forces seller to provide good quality as s/he cannot sell his/her products otherwise with help of reviews. Amazon is one of the important e-commerce website that has plenty of customers worldwide that reviews the products that they bought, moreover, as Amazon cares about reviews and gives penalty to sellers in case of any bad experience; customers also get encouraged to do reviews about the goods that they bought. Note that data must be split data into train (80%) and test (20%) sets randomly.

Dataset

The dataset contains product reviews that contains title of the review, content of the review and given star for the review which is as an integer that is ranging from one to five that given during review. It contains 72,500 reviews and each star has equal number of samples. It can be assumed that reviews that have 1 or 2 stars are negative, and the ones that have 4 or 5 stars as positive, and the ones with 3 stars either assumed as neutral or dismissed with explaining the reason between your selection. Also note that the reviews with 1 star are more negative than the ones with two stars, same applies for the positive reviews. The negativity and positivity level of a review has to be also used as a weight factor. Title and content either processed together or separately by giving appropriate weight according to their importance with explaining the reason between your selection. The approach that you followed during the project will also be graded, so, please be neat while explaining reasoning for your approach.

Natural Language Processing Terminology

- **N-Gram:** An n-gram is a contiguous sequence of n items, items can be word, letter, or any of them, in Natural Language Processing domain it is mostly referred to words, and

n can be any positive integer. For example every word can be considered 1-gram (which is usually called as unigram) and every string that contains two words can be considered as 2-gram (which is usually called as bigram). Considering the sentence "Lorem ipsum dolor sit amet, consectetur adipiscing elit.", unigrams are "Lorem", "ipsum", "dolor", "sit", "amet", "consectetur", "adipiscing", and "elit". Also punctuation marks can be considered as word for better understanding of data, in this manner "," and "." are also unigrams for this sentence. The bigrams of it is (including punctuation marks) "Lorem ipsum", "ipsum dolor", "dolor sit", "sit amet", "amet ,", ", consectetur", "consectetur adipiscing", "adipiscing elit", "elit .".

- **Stopwords:** Common words like "the," "is," and "and" that are frequently eliminated during text processing jobs since they usually don't have any relevance for analysis are known as stopwords.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** A numerical metric called TF-IDF is used to assess a word's significance in a document in relation to a corpus, or group of documents. It considers a word's inverse frequency across all papers in the corpus as well as its frequency within a document (term frequency).

Steps to follow

1. Import and visualize the data in any aspects that you think it is beneficial for the reader's better understanding of the data.
2. Use BoW (Bag of Words) methodology to extract relevant information for your very own Naïve Bayes Algorithm (you must implement it from scratch) from the train data, you must use both unigram and bigram, you can also use trigram etc. for further implementation. Note that you may be in need of following to not fail with your implementation:
   - Use logarithmic probabilities instead of the raw format as it may cause numerical underflow. **Keep in mind that multiplication is addition in logarithmic domain according to the log(a*b)=log(a)+log(b) equation.**
   - Deal with the words that are not seen during training stage. (You may use Laplace Smoothing and unknown word handling via assuming that words that occur very rare are unknowns.)
   - You must implement a dictionary for your BoW approach, you must implement it from scratch.
3. Finally compute performance of your model to measure the success of your Naïve Bayes based classification algorithm for each setting you have used (unigram, bigram etc.):

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \textbf{Precision} = \frac{TP}{TP+FP}$$

$$\textbf{Recall} = \frac{TP}{TP+FN} \qquad F_1\textbf{-Score} = \frac{2}{recall^{-1}+precision^{-1}}$$

Bonus

How does the performance of the model get effected if "Word Embedding" with Logistic Regression is used instead of Bag of Words and n-gram approach at the previous part. You can use any extra library for this purpose, Word2Vec and Glove Techniques are the methodologies you may use. Feel free to play with data and generate new results, the points that you will get from bonus will be determined according to your effort, the better you do, the higher you get. Moreover, you can also use extra libraries, such as NLTK, for this part to achieve better accuracies for the BoW approach to compare three of them: Your approach from strach, your approach with NLTK, your approach with Word Embedding. **Keep in mind that you cannot use NLTK for the assignment itself, you can only use it for bonus part to compare and justify your results at the assignment.**

*** The score of the bonus part will be multiplied by your overall score (excluding the bonus part) and divided by the maximum score that can be taken from these parts. Say that you got 80 from all parts excluding bonus part and 5 from bonus part, your score for bonus is going to be 5*(80/100) which is 4 and your overall score will be 80+4=84.**

What to Hand In

You are required to submit all your code in a Jupyter notebook, along with a report in ipynb format, which should also be prepared using Jupyter notebook. **The code you submit should be thoroughly commented and your notebook must be ran and have outputs for each cell in the order of the cells before submission.** Your report should be self-contained and include a concise overview of the problem and the details of your implemented solution. Note that your report also has to contain necessary libraries to be installed with the versions that are used (!pip install commands are preferred). Feel free to include pseudo-code or figures to highlight or clarify specific aspects of your solution. Submission hierarchy must be as follows:

```
- <GroupID>.zip
    - assignment3.ipynb
    - *.(jpg|jpeg|png|gif|tif|tiff|bmp|svg|webp) (optional)
```

Do not send the dataset.

Preparing a good report is important as well as the correctness of your solutions! You should explain your choices and their effects on the results. You can create a table (or any content you believe that it is beneficial to show your all work) to report your results.

Note that submission format is crucial and submit system is set to give you score as one if you follow the submission hierarchy, which is really easy (there might be some issues for the MacOS users but it can be overcomed via the mini guide that is shared at the Piazza). If you do not score one from the submit system **you will penalized by 20% even if your submission hierarchy is correct**.

The restrictions continues at the next page!

P.S.: You can use libraries for holding (such as numpy, pandas etc.) and representing (such as matplotlib, seaborn etc. for visualization and explanation) data, but you must implement Naïve Bayes, BoW, and things related to their mechanics from scratch, which means any library usage is forbidden except for the visualization and explanation purposes. **Usage of the NLTK is also forbidden!** Note that you can use train_test_split from scikit-learn, it is an exception. Moreover, you can use any libraries **only for the bonus part as written at bonus section, it is also an exception, but do not forget to separate the project itself and bonus part at your Jupyter Notebook, bonus part (including new imports that you did such as NLTK etc.) must start after the project itself, anything related to non-bonus part must not be in after the bonus part**.

Academic Integrity

All work on assignments must be done on your own group unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudo-code) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.