CMPS142-Fall 2018
Project
Sentiment Analysis on Movie Reviews

Handed out: November 6, 2018
Evaluation deadline: Dec 3, 2018
Report and code due on Dec 7, 2018

- You are allowed to do this project in groups of 3 (in some cases 2).
- 10% of the points are for the group's diversity.
- We recommend implementing all the codes in Python3.
- You are allowed to use any external library including Machine Learning and Natural Language Processing libraries.
- One (and only one) member of the group has to submit the project report using his/her account on Canvas. All group members will get points for that submission.
- How to submit your solutions: Your project report must be typed up separately (in at least an 11-point font) and submitted on the Canvas website as a PDF file.
- Your report should clearly mention the names, email addresses and student ids of all group members.
- You are very strongly encouraged to format your report in LATEX. You can use other software but handwritten reports are not acceptable.
- The Computer Science and Engineering Department of UCSC has a zero tolerance policy for any incident of academic dishonesty. If cheating occurs, consequences within the context of the course may range from getting zero on a particular assignment, to failing the course. In addition, every case of academic dishonesty will be referred to the student's college Provost, who sets in motion an official disciplinary process. Cheating in any part of the course may lead to failing the course and suspension or dismissal from the university.

---

1 Course Project [100 points]

With the growing impact of online platforms like social media, such as Twitter, and review websites like Yelp and Rotten Tomatoes, it has become important to gather insights from the huge amounts of subjective data. In this project, you will learn how to derive insights from a corpus of movie reviews. To this end, you will be asked to automatically predicts sentiment of textual phrases from movie reviews. Specifically the Machine Learning task is as follows: given an input phrase, you have to classify it into one of the following sentiment categories: negative, somewhat negative, neutral, somewhat positive, positive. Solving this task allows us to analyze the intricacies of sentiment and to capture complex linguistic phenomena.

1.1 Dataset

The training dataset provided to you consists of phrases and their human-annotated sentiment labels from the Rotten Tomatoes dataset– a corpus of movie reviews originally collected by Pang and Lee [2]. These phrases were extracted from sentences which have been shuffled from their original order and parsed into phrases by the Stanford parser [1]. Each phrase has a PhraseId and SentenceId indicating which sentence it is associated with. Phrases that are repeated (such as short/common words) are only included once in the data. Each phrase is also associated with one of the following sentiment labels:
0 - negative
1 - somewhat negative
2 - neutral
3 - somewhat positive
4 - positive

You will be provided:

- train.csv which contains the phrases and their associated sentiment labels.
- test.csv which contains the phrases (and not labels). The test set will be provided shortly before the project evaluation deadline.

```
PhraseId, SentenceId, Phrase
77679,3994,should come with the warning " For serious film buffs only !
58875,2969,enjoyed it
110618,5863,George W. Bush , Henry Kissinger , Larry King
100868,5298,Lang 's Metropolis
75188,3856,'70s exploitation picture
37645,1790,Nights feels more like a quickie TV special than a feature film ...
```

Figure 1: Sample showing the format of the predictions file the test (test.csv) that your code will take as input

```
PhraseId, Sentiment
77679,1
58875,4
110618,4
100868,3
75188,2
37645,2
```

Figure 2: Sample showing the format of the predictions file (.csv) that your code will output

1.2 Evaluation

Your trained model will be evaluated on a held-out and hidden test set. As mentioned above, the goal at test time is to assign a sentiment label to each phrase in test set. The test set will be provided to you on the day of evaluation only. Your code should take as input the test file (with no labels) and output predictions in a *.csv* file. We will evaluate your predictions against the ground truth (hidden from you at all times) using the following performance measures: Accuracy, Precision, Recall and F1-score. Figure 1 shows a sample of file format for the test set (input).

Your system should be able to accept such a file as input. Note that the file is comma separated and contains a header. Also, the last column represents the phrase and it can contain punctuation marks including commas and quotation marks. Figure 2 shows a sample of file format for the predictions file (output). It should contain only two columns as shown in the figure and should also a header. The file should be comma separated (it should not contain any other punctuation marks like quotation marks). Outputs that do not conform to this format will not be evaluated.

1.3 Report

You are also expected to write a short report of your findings. The report will describe the details of your approach like the data cleaning/pre-processing, feature extraction, model details, and experiments done to build your model. The first section of the report should be titled 'Tools used' and should list all the tools/libraries that you use for the project. In the report, indicate whether you wrote code for a particular step or used a library. For example, if you try Logistic Regression, when describing your approach indicate if you used a library or coded the algorithm. The first page should also contain a small paragraph on diversity. Diversity of the group can be based on a variety of factors and as mentioned in class you don't have to limit yourself to race/gender. Talk to your groupmates and find out how you might be different from them. You will be evaluated on your description of diversity. A typical report would have the following components/sections, but feel free to customize the suggested components according to your project.
REQUIRED COMPONENTS:

1. Title
2. Group details (full names, email addresses and student ids of all group members)
3. Tools Used (including a short 1-2 sentences description of what they were used for)
4. Diversity

SUGGESTED COMPONENTS:

1. Abstract (1 paragraph summary of your approach and key findings)
2. Data Pre-processing
3. Feature Extraction
4. Approach(es)
5. Experimental Set-up
6. Results
7. Conclusion
8. Ideas for future work

1.4 What to Submit:

1. Report (.pdf file) to be submitted on Canvas.
2. ⟨Names⟩_code.zip: This file should contain any code that you write for the project. It should contain a ReadMe and the code should be properly documented. This file should be submitted on Canvas with your report.
3. ⟨Names⟩_predictions.csv containing the predictions of your model on the provided test set. Check Figure 2 for expected file format. Please note that we will not be able to evaluate your predictions if your predictions file is not in the correct format.

In the above description, ⟨Names⟩ should be replaced by last names of all group members in alphabetical order. For example, if there were two members in the group named: Joe Smith and Mary Johnson, then the zip file would be named as JohnsonSmith_code.zip.

References

[1] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60, 2014.

[2] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 115–124, 2005.