HW1 - Movie Review Classification

Published Date:
12 Sep 2024, 11:35 a.m.

Deadline Date:
14 Oct 2024, midnight

Description:
Description:
****

This is an individual assignment
****

Overview and Assignment Goals:
****

The objective of this assignment are the following:

1. Implement the Nearest Neighbor Classification Algorithm
2. Handle Text Data (Reviews of movies)
3. Design and Engineer Features from Text Data.
4. Choose the Best Model i.e., Parameters of a Nearest Neighbor Selection, Features and Similarity Functions

Detailed Description:
For this assignment, your task is to infer sentiment (or polarity) from free form review text submitted for movies.

For the purposes of this assignment, you are required to implement a k-Nearest Neighbor Classifier to predict the sentiment for 25000 reviews for movies provided in the test file (test.dat). Positive sentiment is represented by a +1 review rating, and Negative Sentiment is represented by a review rating of -1. In test.dat you are only provided the reviews but no ground truth rating, which will be used to compare with your predictions.

Training data consists of 25000 reviews as well and exists in file train.dat. Each row begins with the sentiment score followed with a text of the rating.

For both training and test data, each review ends with #EOF to denote the end of review.

For Evaluation purposes (Leaderboard Ranking), we will use the Accuracy Metric comparing the Predictions submitted by you on the test set with the ground truth (hidden from you). Some things to note:

- The public leaderboard shows results for 50% of randomly chosen test instances only. This is a standard practice in data mining challenge to avoid gaming of the system.
- The private leaderboard will be used to rank your entry.
- In any 24-hour cycle, you are allowed to submit a prediction file 10 times only. Thereforeyou're your cross validation diligently before making a submission.
- The final ranking will always be based on the submission with the highest score.
- format.dat shows an example file containing 25000 rows alternating with +1 and -1. Your test.dat should look similar to format.dat with the same number of rows i.e., 25000 but of course with the sentiment score generated by your developed model.

Rules:

- This is an individual assignment. Discussion of broad level strategies are allowed but any copying of prediction files and source codes will result in honor code violation.
- Feel free to use the programming language of your choice for this assignment, but Python is preferred and recommended.
- While you can use libraries and templates for dealing with text data, you are required to implement your own nearest neighbor classifier and cross validation.
- Cross validation is required to determine the best parameter choices, and should be done prior to submitting your predictions. In other words, do not use the accuracy score to decide on the parameter choices. This includes: the number of neighbors (k), the bag-of-words representation (binary vs. raw frequency count vs. TF*IDF), and distance/similarity measure (at least two). In your study, also include experiments using different features (e.g. full feature set vs. reduced feature sets using various dimensionality reduction or feature selection techniques).
- The TA should be able to run your code and recreate the same accuracy as your Miner submission. If for some reason the same accuracy cannot be recreated, you will be asked to explain the discrepancy. So if you do any random sampling in your algorithm, make sure you save the samples in a file (and mention it on your report) so we can recreate the results.

- Each student is allowed to use only one Miner account throughout the semester.

Deliverables:

- Valid Submissions to the Miner2.vsnet.gmu.edu website
- Submission of source code on Gradescope and report on Canvas. Be sure to include the following in the report:
    1. User Name Registered on miner website.
    2. Rank & Accuracy score for your submission (at the time of writing the report).
    3. Instruction on how to run your program.
    4. Your Approach. Describe your parameter choices and how you choose your best parameters and features via cross validation (see above). Report the results on all experiments. Use tables or plots to make your presentation of results more clear.
    5. Efficiency of your algorithm in terms of run time. Did you do anything to improve the run time (e.g. dimensionality reduction)? If so, describe them and report run times with their respective accuracy before and after the improvement.

Grading:

Grading for the Assignment will be split on your implementation (70%), report (10%) and ranking results (20%).

Files:

- *Train Data:* Download File (/files/uploaded_files/1726155335_0552218_1706639070_1872668_1694136687_852449_train_new.txt)
- *Test Data:* Download File (/files/uploaded_files/1726155335_1536613_1706639070_3843262_1694136687_9313295_test_new.txt)
- *Format File:* Download File (/files/uploaded_files/1726155335_2495596_1706639070_492611_1694136688_007641_train_new_ground_truth.txt)

rangwala at cs.gmu.edu (mailto:rangwala@cs.gmu.edu)