Project 3: Movie Review Sentiment Analysis

Fall 2023

Contents

Movie Review Data

The IMDB movie reviews comprises 50,000 entries, each representing a movie review. It contains four columns:

- "**id**": An identification number assigned to each review.
- "**sentiment**": A binary label where 0 indicates a negative sentiment, and 1 indicates a positive sentiment.
- "**score**": This column represents the reviewer's assigned score on a scale of 1 to 10. Ratings from 1 to 4 correspond to **negative** sentiments, while ratings from 7 to 10 correspond to **positive** sentiments. Notably, this dataset does not include reviews with scores of 5 or 6.
- "**review**": The actual text content of the movie review.

Project Datasets

In this project, you are given 5 sets of training/test data. (Download link is provided on Coursera/Canvas.) Each set represents a random split of the original movie review data into two halves, with 25,000 samples allocated for training and 25,000 samples for testing. Each set consists of three files:

- `train.tsv`: The training data with columns "id", "sentiment", and "review"
- `test.tsv`: The test data with columns "id" and "review"
- `test_y.tsv`: The test data with columns "id", "sentiment", and "score"

Notably, the training data does not include the "score" column to prevent the unintended utilization of "score" as an input feature. You can download these five sets of training/test splits from the provided source.

A portion of this dataset was originally used in a Kaggle competition titled Bag of Words Meets Bags of Popcorn. For further insights and sample code, you can refer to the Kaggle competition discussion.

Objective

The objective is to construct a binary classification model capable of predicting the sentiment of a review while ensuring a **vocabulary size** that is less than or equal to **1000**. It's important to note that in this context, a phrase like "worst_movie" is treated as a signle word.

You must utilize the **same vocabulary** for all five training/test data sets to ensure consistency.

The evaluation metric for this project is the **Area Under the Curve (AUC)** on the test data. Your goal is to achieve an AUC score equal to or greater than 0.96 across all five test data sets.

Submission Guidelines

Please provide the following **four** items on Coursera/Canvas for your project submission:

- **myvocab.txt**: This text file should contain your word list, with each word or phrase on a separate line. Ensure that the total number of rows is less than 1000. Please note that partial credit may be granted for vocabulary sizes that surpass 1000 but remain below 3000. For specific details, refer to the rubric provided on Campuswire.
- **R/Python Markdown/Notebook File (in HTML)**: This file should comprehensively explain how you constructed your vocabulary. Include all the necessary code within this file to ensure that your results can be easily reproduced. This document should serve as a detailed guide to your vocabulary construction process.
- **Report (PDF)**: Submit a concise report (maximum of 2 pages, in PDF format) which contains two sections:
  - **Section 1**: Technical Details: Discuss details such as data pre-processing and other non-trivial implementation aspects of your models. Your description should be comprehensive enough for your fellow PSL classmates to replicate your results.
  - **Section 2**: Evaluation Metric: Report the accuracy (AUC) of your prediction on each of the 5 test datasets (refer to the evaluation metric described above), the execution time of your code, and details of the computer system you used (e.g., Macbook Pro, 2.53 GHz, 4GB memory or AWS t2.large) for each of the 5 fold.
- **Code**: Your R/Python script should be in a singular file named either mymain.R or mymain.py. This script should:
  - Accept `myvocab.txt`, `train.tsv` and `test.tsv` as inputs.
  - Generate one file named `mysubmission.csv` based on the specified format (described below).
  - Important: Do not submit ZIP files or markdown/notebook files. Do not try to access `test_y.tsv` in your code.

Code Evaluation

Execution:

- For R: We'll run `source(mymain.R)` in RStudio from a clean environment (meaning, no pre-loaded libraries).
- For Python: We'll execute python `mymain.py` from the command line.

We will execute the command in a directory that contains only **three** files:

- `myvocab.txt`
- `train.tsv`
- `test.tsv`

After executing your code, we should find a CSV file named `mysubmission.csv` in the same directory. This CSV file should include a header, and its entries should be separated by commas, structured as follows:

```
id,     prob
47604,  0.940001011154441
36450,  0.584891891011812
```

```
30088,  0.499236341444505
18416,  0.00687786009135037
```