STAT 542 Project 3: Movie Review Sentiment Analysis

Wenbo Fu (679744457), Bingyan Liu (668046518)

1 Data and Objective

The IMDB movie reviews comprises 50,000 entries, each representing a movie review and a sentiment where 1 represents positive and 0 represents negative. In this project, we are given 5 sets of training/test data. 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. The goal is to predict the sentiment for the testing data while ensuring a vocabulary size that is less than or equal to 1000. The goal is to achieve an AUC score equal to or greater than 0.96 across all five test data sets.

Details can be found in `https://liangfgithub.github.io/Proj/F23_Proj3.pdf`.

2 Method

Our algorithm contains two parts: vocabulary generation and model fit.

In the vocabulary generation part, we combine the training and testing reviews and use Tf-idf vector to represent each entry. We include 1-4 word grams and use nltk stopping words. Then we use Lasso logistic regression to reduce the feature size with the training data from the first split. Setting the regularization coefficient $C = 1.01$ gives a vocabulary of size 996.

In the model fit part, we use two models and combine the two as a bagging algorithm. The first model is a logistic regression with l2 normalization, the second model is an XGBoost classifier. The regularization in the first model is determined by a 5 fold CV. The hyper parameters in the second model is searched by a 5 fold CV in a greedy way. The hyper parameters are max depth = 4, estimators = 500, learning rate = 0.2, min child weight = 6. We find that the first model has a better performance on AUC thus we put a greater weight(0.7) on the first model and a smaller weight(0.3) on the second model.

3 Result and Discussion

The AUC on the 5 splits:

$$0.960, 0.962, 0.962, 0.963, 0.963 \tag{1}$$

The processing time on the 5 splits:

$$36.56, 32.73, 35.31, 34.97, 38.76 \tag{2}$$

All the time are in the unit of seconds. The vocabulary generation step processing time is 80.2 seconds. The code is excuted on a Macbook Air, 8GB memory, Apple M1 chip.

To summarize, we use methods described in the instructor's posts: `https://campuswire.com/c/G06C55090/feed/626`. We don't do more on the vocabulary generation process but use a bagging algorithm combining both a logistic regression model and an xgboost classifier.