Project 3: Movie Review Sentiment Analysis

tianni2@illinois.edu

December 3, 2021

1.Introduction

1.1Background

I was provided with a dataset consisting of 50,000 movie reviews of IMDB. Each row of the data has four variables: id--the identification number, sentiment--0 means negative review and 1 for positive review, score--the 10-point score assigned by the revieIr, where scores 1-4 correspond to negative sentiment and scores 7-10 correspond to positive sentiment(score 5 and 6 Ire removed from the data), review--the text message of review.

1.2 Objective

The goal of this analysis is to build a binary classification model to predict the sentiment of a movie review with a vocabulary size less than or equal to 1000. I was required to use the same vocabulary for all five training/test datasets.

I also need to produce AUC (on the test data), which is the evaluation metric, equal to or bigger than 0.96 over all five test data.

2.Technical Details

2.1 Preprocessing

I am provided with a file called "project3_splits.csv", which contains 25,000 rows and 5 columns, each column containing the 25,000 row-numbers of a test data. And I need to use these numbers to produce the 5 sets of training/test splits with each set--train.tsv, test.tsv and test_y.tsv --stored in a subfolder. More than that, the training data did not contain the "score" column to avoid I mistakenly using "score" as an input feature.

Also, I need to shrunk the vocabulary size to less or equal to 1000. To do this, I use the whole dataset to construct the DT(DocumentTerm) matrix(maximum 4-grams). After remove some stop words, I got the document_term_matrix. To filter out a vocab words that I can interpret, I decided to calculated the t statistics for each variables and pick the top 2,000 words with the highest t stat values. I use commands from the R package slam to efficiently compute the mean and variance for each column of the dtm matrix. And after some algorithms I get the t statistics for all the variables. Since my target is a vocabulary size less than or equal to 1000, I use lasso (with logistic regression) to trim the vocabulary to 976.

2.2 Models

I choose Lasso (with logistic regression) model to fit the data.

2.3 Training process

Since the split data could not be applied to the model directly, I need first transform the data into pure words and calculate the document_term_matrix of words in the vocab we calculated at the beginning for the given data. For each split set, both training and test data need transformation and re-calculation.

2.4 Tunning parameters

The only tunning parameter was lambda used in the lasso cross-validation model. And I choose the value of lambda to be its minimum when process prediction on test data.

3.Model Validation

3.1 Model performance

The lasso model performs well in predicting the the sentiment. For the five split sets, the AUC on the test data are greater than 0.96. Below is the summary table.

Split set	AUC
Split I	0.9659
Split II	0.9664
Split III	0.966
Split IV	0.9667
Split V	0.9654

3.2 Model limitations

Though we achieve the 0.96 AUC bound, there are some limitations existing in my process. My vocabulary is built based on the whole data, instead of only the training set, which is kind of cheating. Because when we are dealing with some issues in reality, we could only get the training set. More than that, the vocabulary needs to be updated when there are new reviews set coming in.

3.3 Error explanation

The vocabulary list could be split into positive words and negative words, where positive words has positive coefficients in the lasso model because they could increase the probability of the review being classify as positive. Below is the summary table contains 10 largest and 10 smallest beta coefficients in the lasso model of split set 1.

Top Negative words		Top positive words	
words	beta	words	beta
4_10	-2.846151954	7_10	3.734288557
3_10	-2.588735308	or_hate	2.717506907
2_10	-2.41077505	7_out	2.515534702
not_recommend	-2.321199504	marvellous	2.311558284
1_out	-2.08273504	8_10	2.2966891
yawn	-1.961458989	this_fun	2.027477597
only_saving	-1.896325851	7.5	1.961062109
this_dull	-1.783772033	negative_comments	1.873240046
had_high	-1.716606531	glued	1.846663411
waste	-1.696696589	little_slow	1.725657946

In the analysis of split set 1, we got a misclassification when id is 3, the review is actually negative while the predicted probability is 0.964584061, which means it's highly possible to classify this review as positive. Below in the box is the review text:

> "All in all, this is a movie for kids. We saw it tonight and my child loved it. At one point my kid's excitement was so great that sitting was impossible. However, I am a great fan of A.A. Milne's books which are very subtle and hide a wry intelligence behind the childlike quality of its leading characters. This film was not subtle. It seems a shame that Disney cannot see the benefit of making movies from more of the stories contained in those pages, although perhaps, it doesn't have the permission to use them. I found myself wishing the theater was replaying \"Winnie-the-Pooh and Tigger too\", instead. The characters voices were very good. I was only really bothered by Kanga. The music, however, was twice as loud in parts than the dialog, and incongruous to the film.<br /><br />As for the story, it was a bit preachy and militant in tone. Overall, I was disappointed, but I would go again just to see the same excitement on my child's face.<br /><br />I liked Lumpy's laugh...."

The words “great”, “loved”, “good”, “excitement” and “laugh” definitely increase the the probability that it would be regarded as a positive review. While there are some negative words like “disappointed” and “preachy”, the possibility still is pretty large.

3.4 Future step

I want further shrunk the vocabulary size, try use lasso or elasticnet to select a smaller value for vocab size.

4. Running time

With 2019 MacBook Pro, 2.4 GHz Intel Core i5 CPU with 16GB memory. The running time for each split is about 30 seconds.

5. Conclusion

When we are trying to classify the test review, the vocabulary selection is very important. After removing some stop words, I apply the combination of DocumentTerm matrix and lasso(with logistic regression analysis) to shrunk the vocabulary size to 2000. Then I calculate the t-stat value for each words and select the top 976 words with largest t value. That how I get the vocabulary set. Then use lasso to fit the model and make prediction. I finally get the AUCs for all five splits are greater than 0.96, which is exactly what I need.

6.Acknowledgment

Professor Liang’s Campuswire posting for the Project 3(Fall 2021)
R document about the packages test2vec, slam