Movie Review Sentiment Analysis Report

11/27/2022

- Deli Yang (deliy2); Tianhong Yin (tyin7); Rachel Banoff (banoff2)

Contributions: Tianhong Yin and Deli Yang wrote and checked the code together. Tianhong Yin wrote the report.

1. Overview

In this project, we deal with a data set consisting of 50,000 IMDB movie reviews, with each being labelled as positive or negative. Specifically, the data set contains review ID, sentiment (a binary variable indicating a positive or negative review), score (the 10-point score assigned by the reviewer, where scores 1-4 mean negative sentiment and scores 7-10 mean positive sentiment), and review content. Reviews with score 5 or 6 are not included in the data set. The goal of this project is to build a binary classification model to predict the sentiment of a movie review.

With 50000 data observations in total, we create 5 sets of training/test splits in 5 subfolders, each of which has (1) training data with 25000 IDs, sentiments and reviews, (2) a test data with 25000 IDs and reviews, and (3) another test data that contains IDs, sentiment and score. The training data do not contain the "score" column, which is to avoid mistakenly using "score" as an input feature. The first test data with IDs and reviews is used to predict the probability of the sentiment of a movie review. The second test data with IDs, sentiment and score is used to be compared with the predicted outcome so as to understand whether we make a good prediction.

The evaluation metric used in this project is Area Under the Curve (AUC) on the test data, which measures how well a specific classifier distinguishes between different classes, and we always prefer a bigger AUC. The ROC (Receiver Operator Characteristic) curve plots the true positive rate (TPR) against false positive rate (FPR), which is the measurement curve for AUC in this project. TPR equals to TP/(TP+FN) and FPR equals to FP/(FP+FN), where TP is the number of true positive cases, FP is the number of false positive cases and FN is the number of false negative cases. The goal is to produce AUC score bigger than 0.96, and we produce AUC over all five test data by using LASSO with logistic regression. The smallest AUC score is 0.9659 for the fifth train/test split.

2. Technical Details

Before building models and making a prediction, we first pre-process the data and construct our customized vocabulary.

First, following the Campuswire post, we stem, tokenize, and vectorize each review to create a Document Term matrix (maximum 4-grams), and we use the default vocabulary size (namely

the number of columns of dtm_train) more than 30000, which is bigger than the sample size 25000. Specifically, we remove the common words, such as “i”, “me”, “my”, “myself”, which do not contain substantive meanings from the reviews. We also remove special symbols such as brackets and punctuations from the reviews. Then we use the itoken() function in the text2vec library to convert all review texts into lower cases and conduct tokenization by specifying tokenizer to be word tokenizer. Then we filter and only include terms with at least 1 word and at most 4 words. Those terms should appear at least 10 times across the movie reviews, and they also should account for at least 0.1 percent and at most 50 percent of the reviews. Then we vectorize texts and use it to create a DT matrix.

Second, we use Lasso (with logistic regression) to trim the vocabulary size to 1000. Using the glmnet() function, we get an output that contains 100 sets of estimated beta values corresponding to 100 different lambda values. Specifically, tmpfit$df tells us the number of non-zero beta values (namely the df) for each of the 100 estimates. We choose the largest df among those less than 1000, which shows the 36th column with 976 non-zero term coefficients and 976 terms, and store the corresponding words in myvocab. A word here actually means a term, which could be a phrase involving multiple words, such as “do_not_watch_this” or “lost_interest”.

3. Model

The model we use is cross-validated LASSO (alpha=1) with the logistic regression, which is useful when the dependent variable is a binary variable. Before training the model, we repeated part of the previous data pre-processing process for each of the five training and test data sets by removing special symbols, converting all texts to lower cases, and conducting tokenization by specifying tokenizer to be word tokenizer. Then we use the vocabulary list already created to create the document term matrix.

In this case, the 976 terms in the document term matrix serve as the features used to predict the sentiment of each movie review. Each of these terms has a coefficient, which can be influenced by a penalty term in LASSO with a shrinkage parameter lambda chosen by cross validation. In this project, the lambda is the one that achieves minimum cross-validation error in making predictions and classifying movie reviews in the test sets.

4. Results

The table below shows the AUC scores and the running time for each train/test split. All splits have an RUC score over 0.96. The running time includes both training and prediction time, which excludes the process of vocabulary construction. The work was done on a MacBook Pro with 2.9 GHz, Intel Core i5 and 8GB memory, and the system is MacOS Mojave (Version 10.14.6)

Split No.	Split 1	Split 2	Split 3	Split 4	Split 5
AUC Score	0.9664	0.9667	0.9664	0.9672	0.9659
Running Time	66.030s	60.439s	60.827s	70.353s	55.271s

Figure 1: Accuracy and Running Time

5. Interpretability of the Algorithm

With LASSO with the logistics model, it is not difficult to understand why the algorithm assigns different particular scores (i.e. the probability of being positive here) to particular reviews. The table below shows the top 30 positive words and top 30 negative words from the first train data set. We can find that they are clearly different from each other in terms of their sentiment. For the positive words, a word that in an earlier place has a larger prediction power in predicting a review being positive. Similarly, for the negative words, a word that in an earlier place has a larger prediction power in predicting a review being non-positive (i.e. negative).

```
 [1] "great"       "excellent"   "wonderful"   "best"        "of_best"     "one_of_best"
 [7] "love"        "perfect"     "loved"       "amazing"     "beautiful"   "superb"
[13] "well"        "favorite"    "brilliant"   "highly"      "life"        "must_see"
[19] "also"        "fantastic"   "very"        "beautifully" "one_of"      "both"
[25] "today"       "always"      "very_well"   "performance" "enjoyed"     "performances"
```

Figure 2: Positive Terms Examples

```
 [1] "bad"         "worst"       "waste"       "awful"       "terrible"    "worse"
 [7] "boring"      "no"          "stupid"      "nothing"     "waste_of"    "poor"
[13] "horrible"    "of_worst"    "minutes"     "even"        "so_bad"      "just"
[19] "crap"        "supposed"    "one_of_worst" "acting"     "poorly"      "supposed_to"
[25] "at_all"      "why"         "plot"        "script"      "ridiculous"  "waste_of_time"
```

Figure 3: Negative Terms Examples

Here are two examples of review information with their ID, sentiment, score and the probability of being positive. One of them has a positive sentiment with a high score and the other one has negative sentiment with a low score. We choose five relevant words from the two reviews as examples to see how their probabilities might lead to different sentiment predictions.

ID	Sentiment	Score	Probability
14833	1	10	0.999999
28256	0	3	4.876391e-11

Figure 4: Review Examples

The text for the first review with ID 14833 is “I do not have much to say than this is a great finish to the story. Most people have said that there is not enough plot and its just eye candy.But think about it, everything was explained in FFVII you cannot add more plot to such a grand story it would ruin it. They did the best that they could do and I think that this should be taken more as A Final FMV.. the last fight. Graphics - 10/10, Absolutely amazing . . . I loved every second of this movie. It was a pleasure to visit the world of FFVII just one last time. . . ”

We can see that there are many positive terms in this review, and five examples include “great”, “best”, “amazing”, “loved” and “pleasure”.

The text for the second review with ID 28256 is “. . . "Hellborn" or "Asylum of the Damned" as is known in the U.S., is a bad movie simply because it is just not involving, and irremediably boring

and tiresome. While it has a very good premise, it is just poorly developed and the mediocre acting doesn't make things better... While the premise is quite interesting, the execution of the film leaves a lot to be desired. In an attempt of making a supernatural psychological thriller, Jones goes for the easy way out and makes a movie filled with every cliché of the genre. Of course, there are lots of great movies that are also filled with clichés; but in "Hellborn" every single one is wasted and turned into a cheap jump scare to keep things moving, resulting in a boring and predictable storyline ..."

We can see that there are many negative terms in this review, and five examples include "bad movie", "poorly", "cliché", "wasted" and "boring".

Below is the table showing the coefficients of the select positive and negative terms, and we can find that positive terms have positive coefficients and negative ones have negative coefficients. Thus we can infer that a review with more positive words is more likely to be predicted as having positive sentiment while a review with more negative terms is more likely to have a negative sentiment.

Positive Terms	Coefficient	Negative Terms	Coefficient
great	0.519419176	bad movie	-0.346360265
best	0.453296283	poorly	-1.321779008
amazing	0.682739728	cliché	-0.473030697
loved	0.505338128	wasted	-0.608396201
pleasure	0.395185847	boring	-0.687570123

Figure 5: Example Term Coefficients

6. Discussion

The table above shows that all of the training/test data splits achieve AUC scores larger than 0.96, which shows that LASSO with logistic regression model generally works in this case. An interesting finding is that reviews with a score closer to 5 and 6, namely the middle point of the score range, are more likely to be misclassied. For example, for the first train/test split, the numbers of misclassfied reviews with the score 1, 2, 3 and 4 are respectively 269, 181, 310 and 507; the numbers of misclassfied reviews with a score 7, 8, 9 and 10 are 375, 264, 140, 289. We can find that generally the closer a review score is to 5 and 6, the more likely they are to be misclassified.

Although the logistic model looks good in making prediction and achieving large accuracy scores, it is limited in its rigid decision boundary, which takes 0.5 as a rigid cutoff value. Any value larger than 0.5 is labeled as one classification and any value below 0.5 is labeled as the other category. However, the substantive meanings of 0.1 and 0.49 might be different a lot in different cases, and we might try different models to make predictions.

To make improvements on models, in addition to trying different models, we might also try different ways to construct the document term matrix, such as using the Term Frequency Inverse Document Frequency (TFIDF) method, so as to compare which approach produces better accuracy scores. We can also try different vocabulary size because the 1000 word size might not be able to capture enough key terms, which might lead to larger errors.