# STAT 542 Project2

## Movie Review Sentiment Analysis

Author: Lingyi Xu (lingyix2)

Date: 04/16/2019

## 1. Introduction

For IMDB movie reviews, we are interested in building models that has a good prediction performance on predicting positive or negative label. The rawdata has 50000 observations, 3 variables. This study will show works of training lasso, lda and xgboost models including technical details such as data preprocessing and evaluation performances.

## 2. The technical details

### 1) Preprocessing

For data preprocessing, I did the following to the given text data:

* Removed the punctuations in the text data. Punctuation can provide. grammatical context which supports understanding, however, when we are doing bag of words based sentiment analysis, punctuation does not have any value.

* Convert text to lowercase. I define the prep_fun to make every letter in the text. data to lower case to avoid distinguish between words simply on case. So the same words will be considered to be identical no matter it is at the beginning of

the sentence or not. It can also help us to shrink the vocabulary size and save space in our database.

* Tokenize the sentences into words. Since we are doing bag of words based sentiment analysis, we need words rather than sentences.

* Ignore words which appear too often. These words are so common that they can barely provide any information of whether a review is positive or not, so we don't want them to mislead our algorithm or to take up space in our database.

2) Model Selection and application

- LASSO

  The family="binomial" argument is appropriate for a classification problem. In this setting, it allows me to estimate the parameters of the binomial GLM by optimising the binomial likelihood whilst imposing the lasso penalty on the parameter estimates. The dichotomous response is perfectly fine here. This is useful because it allows feature selection or parameter shrinkage to avoid overfitting.

  After data preprocessing, I used new training and testing dgCMatrix and cv.glmnet function to build the lasso model which can automatically tune the

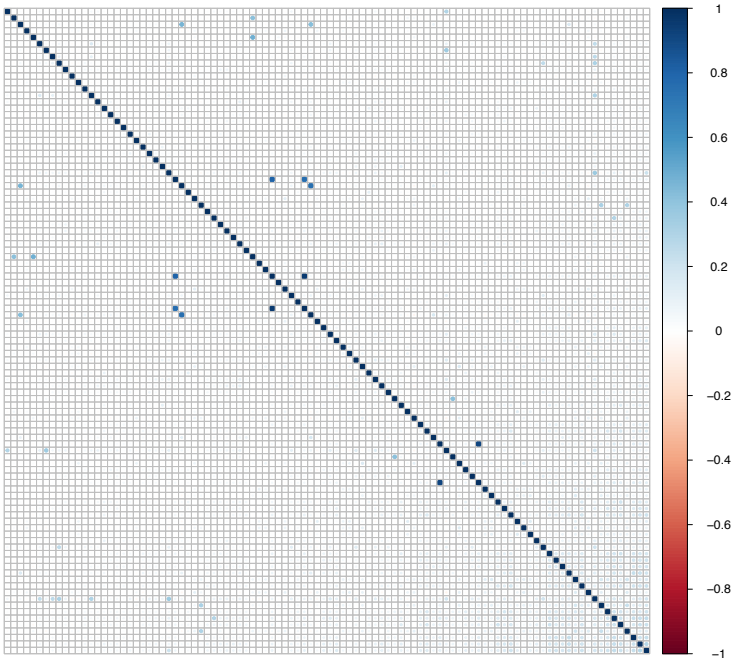
parameter and choose the best lambda. Then I made predictions by estimating the probability that a new set of inputs belongs to which class.

- LDA

Linear Discriminant Analysis is the preferred linear classification technique if. we have more than two classes. In this case, the LDA model estimates the mean and variance from my data for two class.

In this model, I also dropped some high collinear variables according to the collinearity test.

Correlation Matrix



- XGBOOST

Random Forest Model is also good for binary classification.

The parameter I used in this case:

max.depth = 18, nthread = 10, nrounds = 10, eta =0.03, seed = 3,
colsample_bytree = 0.2, subsample = 0.9

After plugging training data sets into the function, we could construct the tree model and make predictions with the new data sets.

## 3. Performance Summary and Conlusion

AUC Evaluation metric

| | Split1 | Split2 | Split3 |
|---|---|---|---|
| auc_lasso | 0.9604519 | 0.9607527 | 0.9617584 |
| auc_lda | 0.9486831 | 0.9491577 | 0.9499553 |
| auc_xgboost | 0.9607887 | 0.9636960 | 0.9649920 |

Here we have the AUC for Lasso, LDA and XGboost models.

From the table above, we could see that AUC of LASSO and XGboost has the best performance which have all three auc velue bigger than 0.96. The benchmark has been reached. The auc values of LDA model are aroud 0.95 which has also a very good performance.

4. The computer system and the running time of our code

1) computer system:

macbook air 10.14.3 (18D109) 1.6 GHz Intel Core i5 8 GB 2133 MHz LPDDR3

2) running time:

| user | system | elapsed |
|---|---|---|
| 927.811 | 255.413 | 1354.529 |