Project 2   Movie Review Sentiment Analysis

Wenjing Du (wenjing4)   *Xinyi Song (xinyis8)*

In this project, we are provided with a dataset consists of IMDB movie reviews, where each review is labelled as positive or negative. The goal of this project is to build a binary classification model to predict the sentiment of a movie review. This report is divided into three parts. The first part is about data pre-processing. In the second part, we provided technical detail of the three models we used. Finally, we evaluated the performance of the three models based on the value of AUC (Area Under the Curve) of ROC (Receiver Operating Characteristics）.

Data Pre-processing

The dataset in this project has 50,000 rows (i.e., reviews) and 3 columns. Column 1 "new_id" is the ID for each review (same as the row number), Column 2 "sentiment" is the binary response, and Column 3 is the review.

First, we cleaned html tags by using R regular expression and split data into training and testing datasets. Here we divided the original dataset into three splits and used two splits as train data and the remaining one as test data. Then we used R package 'text2vec' to build vocabulary and construct document term matrices (DTMs). Here we considered all words and phrases consist of up to 4 words. We ended up having a large number of features, close to 30,000, which is bigger than the sample size n = 25,000. We tried logistic regression with these features and then checked some of the selected variables, i.e., terms/words with non-zero logistic regression coefficients. We found that it was difficult to interpret, since some words might be informative, e.g., "stupid" or "excellent", but they are mixed with a large amount of hard-to-explain words such as “as the”, which could be meaningless in our model and prediction.

To improve the efficiency and accuracy, we tried a screening method with the help of two-sample t test: Suppose that we have two groups of one-dimensional observations $X_1, X_2, \ldots, X_m$ and $Y_1, Y_2, \ldots Y_n$. To test whether the X population and the Y population have the same mean, we compute the following two-sample t-statistic:

$$\frac{\bar{X} - \bar{Y}}{\sqrt{\frac{s_X^2}{m} + \frac{s_Y^2}{n}}}$$

where S_X^2 and S_Y^2 denote the sample variance of X and Y respectively.

After calculating the two-sample t-statistics of words, we ordered the words by the magnitude of their t-statistics and then picked the top 2,800 words. By doing this, we could choose the positive or negative features as informative as possible. Then we used the ordering index to select the positive features whose t statistics are greater than zero, and negative features whose t statistics are less than zero based on the fact that in this dataset, the label of positive movie reviews is one while the label of negative movie reviews is zero. These selected 2,800 words were used in the three model we fitted.

Implementation

In this project, the dependent variable here is limited to two categories, positive or negative, which can be simplified to a binary classification problem, where $Y \epsilon \{0,1\}$. Therefore, we used three kinds of classification models: Logistic Regression Model with Elastic Net Penalty, Naïve Bayes Classifier, and Linear Discriminant Analysis, to analyze movie reviews, predicted their sentiment and evaluated their performance based on AUC of ROC.

Logistic Regression Model with Elastic Net Penalty

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables, and the elastic net is a regularized regression method that linearly combines the L1 and L2 penalties of the lasso and the ridge methods. We used the ten-fold cross validation to choose the optimal λ by calling the function *cv.glmnet*, and then fitted the logistic regression model with sentiment versus key features by calling the *glmnet* function in package "glmnet" and setting "family=binomial" for canonical link function of logistic regression.

The reason that we chose to use Elastic Net penalty here for better model performance is as follows: Lasso regression could help us do model selection by shrinking coefficients to zero. However, the lasso tends to select one variable from a group, ignore the others and fails to do group selection. Besides, the number of selected features is bounded by the number of samples. On the contrary, Ridge Regression only shrinks the coefficients of the variable depends on its importance to the model accuracy rather than providing 0. Therefore, by applying elastic net model, we could obtain both variable selection features of lasso penalty and effective regularization characteristics of ridge regression, remove the limitation on the number of selected variables and obtain grouping effect. Since in this project, the design matrix of predictors (features) could be extremely sparse, to avoid the bound of number of variables and do group selection, we chose Elastic Net Penalty rather than Lasso. Here we set the alpha in the *glmnet* function to be 0.1 for better results.

Naïve Bayes Classifier

The Naïve Bayes classifier is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting[1]. It is based on Bayes' theorem with the independence assumptions between predictors.

[1] https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54

A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Besides, decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution, which helps alleviate problems stemming from the curse of dimensionality, such as the demand for data sets that increase exponentially with the number of features. If the Naive Bayes conditional independence assumption actually holds, a Naive Bayes classifier will converge quicker than discriminative models like logistic regression, thus less training data is needed. And even if the NB assumption doesn't hold, a Naive Bayes classifier might still perform surprisingly well in practice.

By calling the function *naiveBayes* in the package of "e1071", we fitted the naïve Bayes classification model based on our train data and make predictions on our test data.

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a method evaluating how well a group of variables supports an a priori grouping of objects. It is based on work by Fisher (1936) and is closely related to other linear methods such as multiple linear regression, principal components analysis (PCA), and factor analysis (FA)[2].

In LDA, a grouping variable is treated as the response variable and is expected to be categorical. Groupings should reflect ecologically relevant knowledge, such as sampling environment or method, or reflect the results of an exploratory method such as cluster analysis or non-metric dimensional scaling. LDA assumes that the observations conform to Gaussian distribution and each classifier shares the same covariance matrix.

Linear Discriminant Analysis models the distribution of predictors separately in each of the response classes, and then it uses Bayes' theorem to estimate the probability. Comparing with other classification algorithms such as random forests, it is much more interpretable, and the prediction process is easier and more efficient.

Here we fitted the Linear Discriminant Analysis model by calling the function *lda* in the R package "MASS" based on our train data and used test data to do prediction.

Performance Evaluation

We use AUC of ROC to evaluate the performance of our models. A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied[3]. The AUC of ROC curve measures performance for classification model at various thresholds settings. In other words, it tells how much model is able to distinguish between different classes. Other thing being equal, the higher the AUC is, the better does the model do prediction

[2] https://mb3is.megx.net/gustame/discrimination/linear-discriminant-analysis

[3] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

(classify the 0s as 0s and 1s as 1s). In our project, higher the AUC, better the model is at distinguishing between positive movie reviews and negative movie reviews.

We used the *roc* function to calculate the AUC with help of R package “pROC”. Below is the table containing the AUC (Area Under the Curve) for each model with respect to three splits:

Table I. AUC for Each Model among Three Splits

Model	Split 1	Split 2	Split 3
Logistic with Elastic Net	0.9610	0.9618	0.9622
Naïve Bayes	0.8670	0.8602	0.8616
Linear Discriminant Analysis	0.8793	0.8817	0.8809

Based on the table above, we can see that for all three splits, Logistic Regression with Elastic Net Penalty performs better than Naïve Bayes classifier and Linear Discriminant Analysis Model in terms of AUC, which was contrary to our intuition since our data is sparse and extremely high dimensional. Following are possible reasons to explain this scenario:

Logistic Regression VS Naïve Bayes

It is true that for a small training dataset, high bias/low variance classifiers such as Naïve Bayes Classifier perform better than low bias/high variance classifiers, since the later one might overfit. But our dataset is extremely large with 50,000 movie reviews. In this case, low bias/high variance classifiers start to win out with lower asymptotic error and high bias classifiers aren't powerful enough to provide accurate models. Therefore, here Logistic Regression Model with Elastic Net Penalty performs better than Naïve Bayes Classifier.

Logistic Regression VS Linear Discriminant Analysis

The only difference between Logistic Regression and Linear Discriminant Analysis is estimation method: Logistic Regression uses Maximum Likelihood Estimation while Linear Discriminant Analysis assumes that observations conform to Gaussian distribution and then estimates the mean and covariance matrix. And in Linear Discriminant Analysis, every classifier shares the same covariance matrix. Therefore, if the training data set satisfies the Gaussian distribution assumption, Linear Discriminant Analysis performs better than Logistic Regression Model. Since our data do not satisfy this assumption, here Linear Discriminant Analysis does not perform as well as Logistic Regression.

Naïve Bayes VS Linear Discriminant Analysis

Both LDA and Naïve Bayes are linear classifiers and come under the category of Generative Models which estimates the posterior P(class|x). LDA assumes Gaussian class-conditional density models and equal covariances while NB assumes variables to be independent. LDA is closely related to NB in that both classifiers assume Gaussian within-class distributions. However, NB relies on a less flexible distributional model in that it no correlations between variables within a class. An LDA classifier is Bayes-optimal (ignoring estimation error) if the distributions corresponding to the two classes are Gaussian and have equal covariance. Here, in our data, LDA performs a little better than Naïve Bayes.