STAT 154: Tweeter Sentiment Project

Yiyao (Tato) Lu, SID: 23366943

Minki (Min) Seo, SID: 23974602

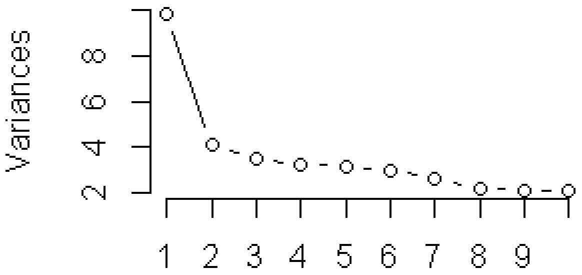
April 28, 2016

The objective of the TweeterSentiment Kaggle competition is to use machine learning to predict whether a given tweet sentiment is positive or negative, a two-class classification problem. For example, a tweet such as "*wow i do not get anything in algebra. its so freakin hard and i forgot everything we learned this year*" might be categorized as having negative sentiment. We competed under the group name "Tato and Min", with our best accuracy score on the Kaggle public leaderboards as 0.76208, which we obtained through the random forest model. In section one, we describe the data; in section two, we explain the models we use; finally, in section three, we conclude with the model performances and results. Credit for the data goes to the Stanford Computer Science department.

I. DATA

We work with the "TrainTest.RData" file, which contains `X`,`y` , and `Xtest`. The matrix `X` contains 50,000 observations of 1,000 predictors. Each of the observations represents one tweet, and each of the predictors represent a count for the most common words that show in the data. For instance, the i'th predictor may represent the word "algebra", which would take a value of one in the earlier example tweet, since "algebra" appears once. The vector `y` is the response vector corresponding to `X`, and it takes a value of one if the tweet sentiment is positive and zero if the sentiment is negative. Finally, the matrix `Xtest` is the test data containing 50,000 new observations of the 1,000 predictors. We use `Xtest` to construct the predictions we submit to the Kaggle competition.

Figure 1: Scree Plot for `X`



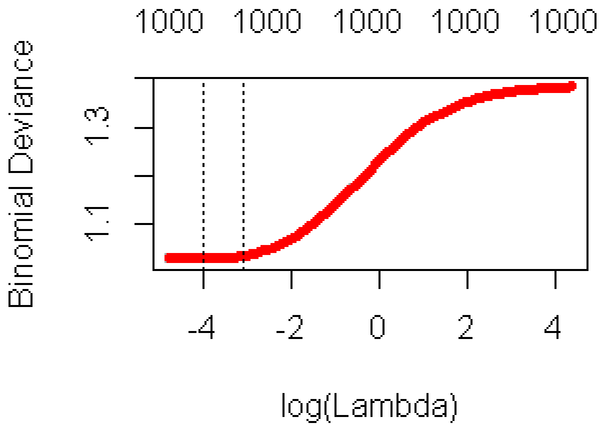
Given the large dimension of `X`, we attempt to run PCA for the purpose of dimension reduction. However, it seems that the predictors do not have a strong covariance with each other, as PCA fails to reduce the dimension of the data significantly. PCA still requires 899 components to explain around 95 percent of the variation in the data. Runnning kernel-PCA produces similar results. As such, we opt not to apply PCA when analyzing the data.

II. MODEL SELECTION

Since the question we want to answer is a two-class classification problem, we use methods such as LDA, logistic regression, L2 logistic regression, and random forest. We expect either logistic regression or the random forest model to perform the best, and we include LDA mainly for the sake of comparison. L2 logistic regression may be useful in identifying the most important words, given the large number of predictors. Due to the large size of the dataset, we opt to work with single-fold cross-validation, randomly assigning about 70 percent of the observations in X as training data and using the remainder as test data. The same allocation of training and test data is used across all methods.

For performance evaluation, we calculate three scores: accuracy, ROC, and F1. The ROC score essentially measures how well the model does in balancing sensitivity and specificity, where sensitivty relates to how often the model detects a positive tweet, and specificity relates to how well the model does in correctly classifying a tweet. Sensitivity can also be thought of as the true positive rate, and specificity can be thought of as the false positive rate. The F1 score is a weighted average of the precision and recall of the model. Precision is defined as the number of correctly predicted positive tweets divided by the number of all predicted positive tweets, and recall is defined as the number of correctly predicted positive tweets divided by the number of actual positive tweets. The scores for each model will be compared to a baseline. The baseline model sets all predictions to the larger of the two classes in the response vector. In this case, the majority of tweets had positive sentiment, so the baseline model classifies all observations as positive.

Figure 2: Parameter Selection for L2 Logistic Regression



To select the parameter λ for the L2 logistic regression, we use cross-validation via the `cv.glmnet` function in R. This produces the plot in Figure 2. In order to avoid overfitting, we choose the largest value of λ such that error is within one standard error of the minimum. This gives $\lambda = 0.04545$.

Table 1: Parameter Selection for Random Forest

`num_parallel_tree`	Accuracy Score
10	0.76087
20	0.76207
40	0.76007

We manually set the number of trees for the random forest model. Changing the parameter `num_parallel_tree` in the `xgboost` function does not drastically change accuracy score of the model, as illustrated in Table 1. As such, we simply choose the number of trees to be 20.

III. RESULTS

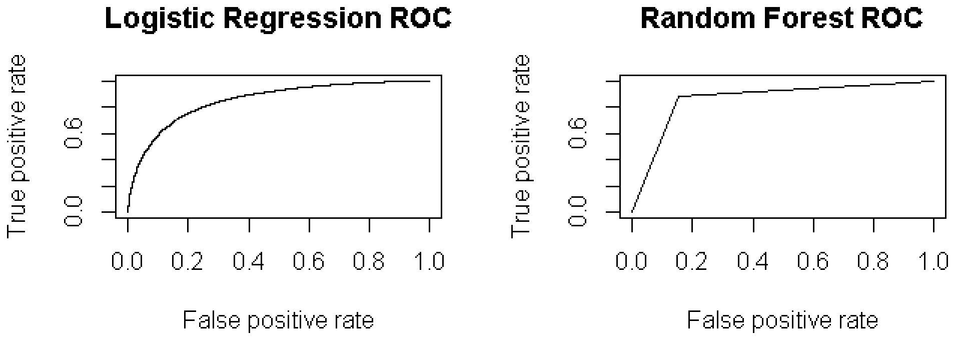
After running LDA, logistic regression, L2 logistic regression, and random forest analyses on the data, we obtain the results in Table 2.

Table 2: Model Performance

Score	LDA	Logit	L2 Logit	RF	Baseline
Parameter	-	-	$\lambda = 0.04545$	ntrees=20	-
Accuracy	0.50287	0.74287	0.73187	0.76207	0.50287
ROC	0.74858	0.83074	0.83159	0.76195	0.49917
F1	0.66921	0.71495	0.68700	0.76781	0.66921
Kaggle	0.10508	0.74384	0.73044	0.76208	0.49780

Of all the methods, the logistic regression and random forest models performed the best, with the highest accuracy scores. The random forest model has higher accuracy and F1 scores than the logistic regression, but the logistic regression has a higher ROC score.

Figure 3: ROC Plot Comparison



To understand why logistic regression might generally have a higher ROC score than the random forest model, we compare the ROC plots for the two methods in Figure 3. Intuitively, one can see how the area under the ROC for logistic regression might be larger than that for the random forest model. In light of this, based off the other two measures of accuracy score and F1 score, we believe the random forest model performs better than logistic regression.

Using the logistic regression and random forest models to construct predictions based off the `Xtest` data, we see that the random forest indeed does better than logistic regression, with an accuracy score of 0.76208 compared to the logistic regression's accuracy score of 0.73044. Logistic regression does not perform any better than the baseline, i.e. labeling all tweets as having positive sentiment. Note that these Kaggle scores are calculated on approximately 50% of the test data, so logistic regression may perform better on the other 50%.

A. Source Code

https://github.com/tatolu/STAT154_FinalProject_TatoandMin.git