Natural Language Processing in Yelp Reviews

Victor Kong

Abstract

This study identifies common English language cues associated with human sentiment and assesses these connections quantitatively via an investigation into Yelp reviews of North American restaurants. Even when given solely knowledge of the word content in a written passage, without context, syntax, or other background information, an accurate judgment can still be made of an individual's opinion regarding

1 Introduction

Natural language processing is a branch of computer science that deals with the study of human language, its interpretation, and its application to interfacing with artificial intelligence. In any language, diction is generally indicative of the true sentiments of a speaker or writer, and a minor deviation in an account's choice of words can drastically change the perceived intent behind it as a whole. It is typical to strategically use certain words and phrases to most potently emanate a particular tone or to best portray a topic in a specific light.

This study investigates English language cues that demonstrate either a positive or negative attitude towards the topic of discussion, specifically how accurately a positive or negative sentiment can be predicted merely by a set of these cues. The researched data set consists of restaurant reviews written by users of the web-based social networking site Yelp.com. These reviews were inspected for the presence of certain words and strings, each of which was then examined for correlation with whether the review corresponded to a positive or negative experience. In the context of this study, a positive experience was defined to be one that earned a rating of four or more stars whereas a negative experience was defined to be one that earned a rating of three or less stars. This study involved developing multiple models to predict whether an experience was positive or negative by these standards, given only knowledge of the presence of certain language cues. Close scrutiny into these models can then also allow assessment of the effectiveness of individual cues in predicting the sentiment associated with a user's experience.

2 Methodology

2.1 Data Cleaning and Feature Selection

The raw data set was packaged within the Challenge Dataset distributed throughout the fifth round of the Yelp Dataset Challenge. Specifically, this data set included approximately 1.6 million reviews for 61 thousand businesses in ten cities across Europe and North America. In a preliminary cleaning, a longitude constraint was imposed upon businesses to eliminate all reviews made in French-speaking Montreal and other territories further east, as well as all reviews made for businesses that did not fall under the category of "Restaurants." At this stage, unsupported characters as well as most reviews written in languages other than English were also removed.

A total of 206 variables were created from English language cues hand-selected from a randomly-generated subset of the reviews. Of these, 203 are binary-categorical, each observation being assigned `TRUE` or `FALSE` according to whether the corresponding review contained the cue. One variable was generated from cues representing contrast (i.e., but, however, though), and observations were assigned values simply by the number of times the cue was detected. The last two variables quantify respectively the use of uppercase letters and the use of exclamation points. Specifically, the former contains ratios of the number of uppercase letters used over the total number of letters used, and the latter contains ratios of the number of exclamation points used over the combined number of exclamation points and periods used.

In a secondary cleaning, the 203 categorical variables were evaluated for how much preference each had for one level of the response. This was quantified using the formula

$$\theta_{cue} = \frac{\max\left(p_{high,cue}, p_{low,cue}\right)}{\min\left(p_{high,cue}, p_{low,cue}\right)},$$

where

$$p_{high,cue} = \frac{\text{number of ``high'' reviews with cue}}{\text{total numbers of ``high'' reviews}} \text{ and } p_{low,cue} = \frac{\text{number of ``low'' reviews with cue}}{\text{total number of ``low'' reviews}}.$$

In words, θ_{cue} is the number of times more likely it is to encounter a cue when given one level than when given the other, so a variable that sorts well would theoretically have θ_{cue} far greater than 1. After elimination of all categorical variables with $\theta_{cue} < 2$, a revised data set included a sample of 921615 reviews and 102 predictor variables.

The final cleaning involved generating models with different sized subsets of the predictors by means of forward stepwise, backward stepwise, and sequential replacement selection. A comparison of BIC, C_p, RSS, and adjusted R^2 values among these models showed that the best was the one with 93 predictors created via backward stepwise selection. Only the predictors associated with this model were kept, the reasoning being that given any prediction model, the variables cut out were likely highly dependent on or simply redundant given the remaining ones. The final data set therefore contained 921615 observations over 93 predictor variables.

2.2 Rating Prediction

This study examined classification prediction models generated by a total of six methods: logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), random forests of decision trees, and support vector machines with linear and radial kernels (SVM-L and SVM-R, respectively).

Models generated via logistic regression, LDA, and QDA were all assessed using 10-fold cross-validation over the same ten partitions and the means for each model were computed. In addition, fifty evenly-spaced values between 0.01 and 0.99 were used as threshold probabilities for when to predict a review's corresponding rating to be "high."

In the two cases of decision trees, the necessary calculations were far too computationally expensive to use large training and test sets. Instead twenty disjoint random samples were selected, ten training sets of size 5000 and ten corresponding test sets of size 1000. Each of the training sets were used to train a single model, which would then be evaluated using the corresponding test set. The ten test error rates for each method were eventually averaged. The study also considered ten different values for the number of predictors considered in each iteration of growing a random forest.

Likewise, high computational expense once again prevented the use of the entire data set in training and testing support vector machines, so the same method as that which was used for decision trees was also employed here.

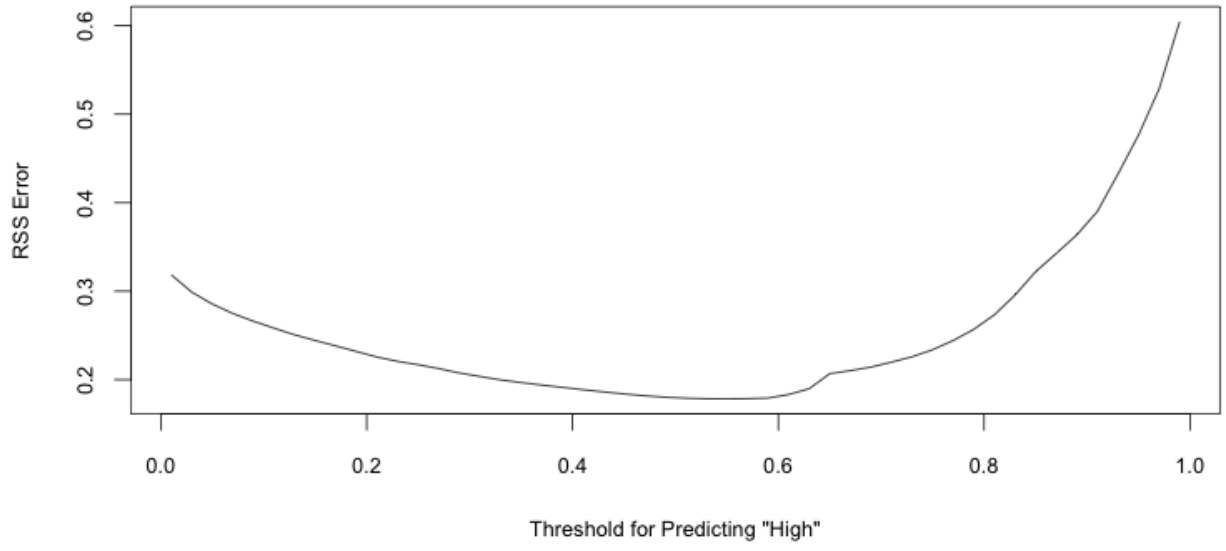
3 Results

3.1 Logistic Regression

Logistic regression models saw that the same coefficients were consistently either very positive or very negative. When these coefficients were placed in ascending order, the following result was observed:

```
> logVars
       (Intercept)       deliciousTRUE     affordableTRUE            bestTRUE 
         0.5148676          1.1981090          0.7492228          0.9055221 
     awesomeTRUE       wonderfulTRUE     refreshingTRUE       homestyleTRUE 
         1.0621003          0.9893590          0.8035371          0.6835192 
          loveTRUE            gladTRUE           relaxTRUE      excellentTRUE 
         0.7808976          0.9198585          0.7093844          1.1500977 
      exclamation       topnotchTRUE          uppercase     handsdownTRUE 
         1.9043048          1.1530934          0.7882216          1.1570087 
       highlyTRUE      favoriteTRUE            yumTRUE          a_mustTRUE 
         1.0748676          1.0252801          0.8371525          1.3580901 
    three_starTRUE      two_starTRUE      one_starTRUE     cant_waitTRUE 
        -1.7705840         -1.8562798         -0.6026853          1.3736184 
unpretentiousTRUE    outstandingTRUE      fabulousTRUE       perfectTRUE 
         1.1154608          0.8183777          0.7671748          1.2496445 
          gemTRUE          juicyTRUE         amazingTRUE       fantasticTRUE 
         1.6851988          0.8881287          1.0939266          1.1386669 
         cozyTRUE       foreverTRUE            hitTRUE           hateTRUE 
         0.6574323         -0.8076682         -0.5061157         -0.3294368 
         lackTRUE          worstTRUE     disappointTRUE    understaffTRUE 
        -1.0524977         -1.9646730         -1.4510997         -0.8962116 
ridiculousTRUE         dirtyTRUE       terribleTRUE        averageTRUE 
        -0.5443417         -0.9512480         -1.8305109         -0.8511299 
      cannedTRUE       greaseTRUE    disgustingTRUE            badTRUE 
        -0.7503179         -0.4107917         -1.9056738         -0.6474825 
another_tryTRUE           clueTRUE           tinyTRUE           slowTRUE 
        -1.5798807         -0.6305397         -0.4398014         -0.6668508 
         however      drawbackTRUE      undercookTRUE     unfortunateTRUE 
        -0.1328901          0.7133008         -1.3589601         -1.0247585 
      not_worthTRUE        grimyTRUE          plainTRUE           rudeTRUE 
        -2.2110517         -0.9800766         -0.3420331         -1.9018766 
     decentTRUE nothing_specialTRUE          boringTRUE          avoidTRUE 
        -0.5884920         -1.1085712         -0.7186167         -0.3201062 
       filthyTRUE      confusingTRUE            noTRUE         at_bestTRUE 
        -2.0056312         -0.4950818         -0.4978326         -1.9757411 
     horribleTRUE       sadfaceTRUE          wasteTRUE      tastelessTRUE 
        -2.0896199         -0.6410675         -1.1455281         -2.0538887 
         coldTRUE          emptyTRUE          soggyTRUE            dryTRUE 
        -0.6198743         -0.5734976         -0.8886643         -0.9604826 
         awfulTRUE          at_allTRUE            oilyTRUE             okTRUE 
        -1.7782245         -0.4549470         -0.4106633         -1.3434469 
    stay_awayTRUE      potentialTRUE        misleadTRUE          yuckTRUE 
        -0.7263507         -1.0026206         -0.9440940         -1.7914916 
     oh_wellTRUE        barelyTRUE          crapTRUE        refuseTRUE 
        -0.3898355         -0.8964151         -0.5325656         -1.2603446 
not_be_backTRUE  cant_go_wrongTRUE          messTRUE     not_evenTRUE 
        -3.4602707          1.1430367         -0.5309087         -0.7477630 
     smileyTRUE       not_failTRUE 
         0.8699589          0.9112141 
```

The cross-validation error associated with logistic regression models also seemed to decrease when the threshold was increased from 0.01 until reaching a minimum at 0.55, after which it rose sharply.

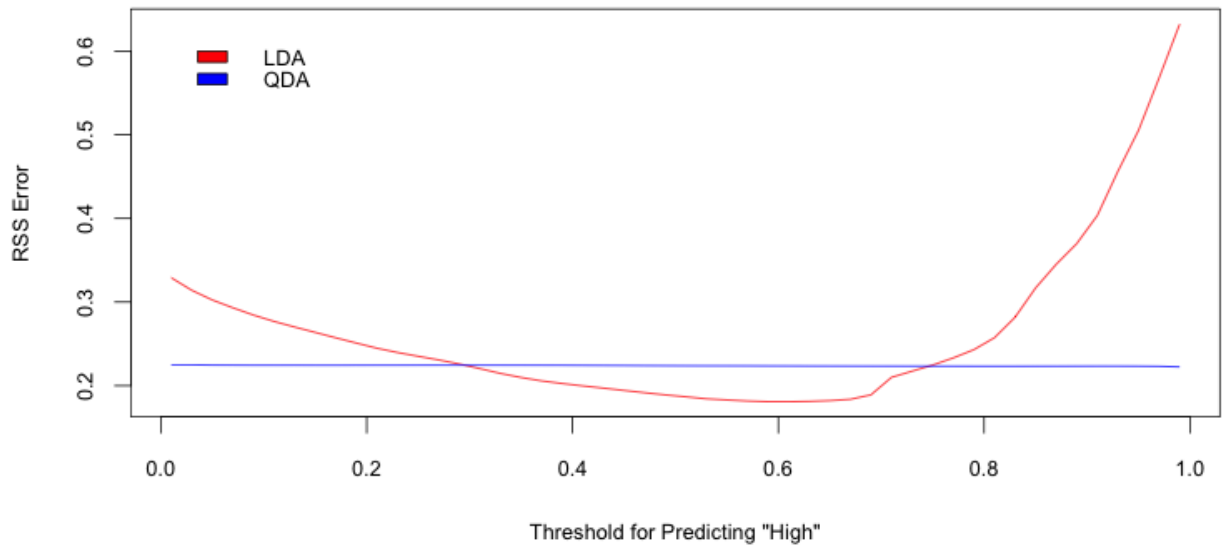


3.2 Linear and Quadratic Discriminant Analysis

Scalings attributed to each variable as a result of model-fitting by LDA provided the following quantitive analysis of predictor strength:

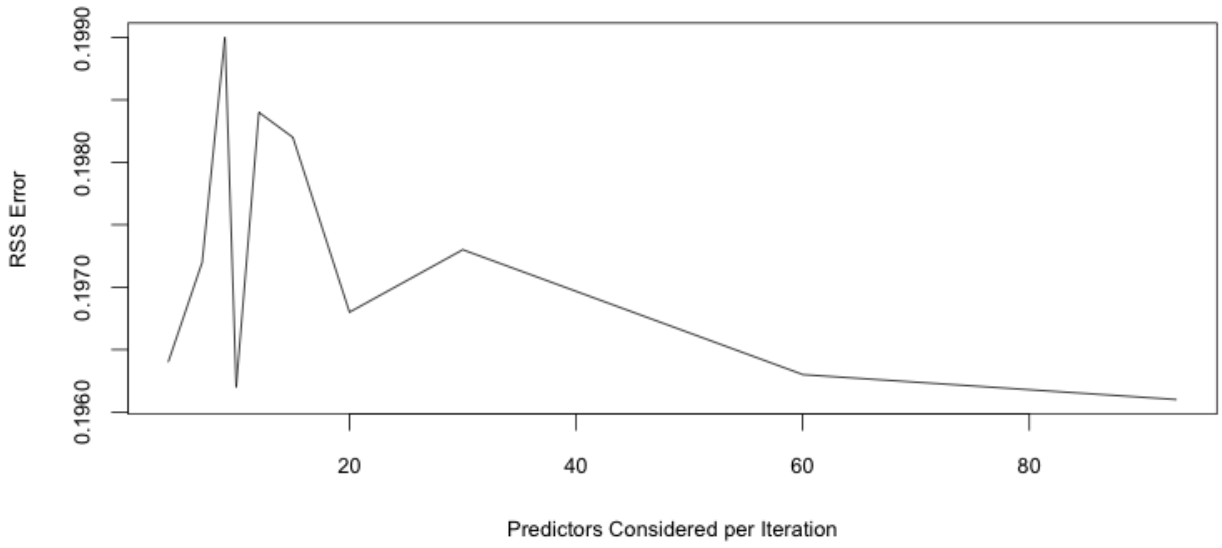
```
> ldaVars
    not_be_back       not_worth        horrible      three_star           rude           worst        terrible
     -1.38164394     -1.11728604     -1.07026146     -1.06105066     -1.05317743     -0.98473613     -0.96620956
         at_best        two_star              ok     another_try          filthy            yuck      tasteless
     -0.95729543     -0.93104849     -0.91014153     -0.89156012     -0.88218118     -0.87707581     -0.86237930
            awful      disgusting      disappoint          refuse       undercook nothing_special           waste
     -0.85099347     -0.82791217     -0.80567845     -0.69752077     -0.67068485     -0.66259778     -0.64647391
             lack         average       understaff     unfortunate           dirty       potential             dry
     -0.62870217     -0.59084210     -0.57993432     -0.57032501     -0.55619229     -0.55000843     -0.54628528
         mislead           soggy          barely          forever            slow           grimy       stay_away
     -0.53722425     -0.49637440     -0.48170130     -0.47725966     -0.45714743     -0.45671603     -0.45394787
              bad          decent        not_even          boring          canned            cold         sadface
     -0.42341168     -0.41042710     -0.39194412     -0.39160002     -0.39032815     -0.37283966     -0.37280475
            empty             hit            clue              no            mess        one_star       ridiculous
     -0.32441818     -0.32394520     -0.32057755     -0.32011566     -0.30849980     -0.30400794     -0.29479726
             crap       confusing           at_all          grease         oh_well            oily            tiny
     -0.29334049     -0.27352959     -0.25257781     -0.25248675     -0.23788243     -0.21997403     -0.21136179
            avoid           plain            hate         however        fabulous           cozy       homestyle
     -0.20443006     -0.18288822     -0.17179546     -0.06378314      0.28923452      0.30234656      0.30304940
       refreshing      handsdown           relax          smiley       cant_wait     outstanding      affordable
      0.32107623      0.32477280      0.36363701      0.37455643      0.37772312      0.38331878      0.38510230
             yum       wonderful        not_fail           juicy            love            glad          highly
      0.39008633      0.39897224      0.39942674      0.39950107      0.42411730      0.42855916      0.43229499
        drawback    unpretentious          a_must   cant_go_wrong        topnotch            best         amazing
      0.45841094      0.46035222      0.47263077      0.47356232      0.47375727      0.47653367      0.48098735
        favorite       fantastic         awesome             gem         perfect       delicious       excellent
      0.49075188      0.50312125      0.51650330      0.53126028      0.54804286      0.58558345      0.59491163
       uppercase     exclamation
      0.74401523      1.02702459
```

Additionally, the analysis shows that the LDA cross-validation error dropped as the threshold was increased from 0.01 until reaching a minimum at 0.61 and sharply increased afterwards. QDA on the other hand appeared to be relatively consistent despite changes in this threshold. In fact, it appears to see its the lowest RSS error when using a threshold of 0.99.



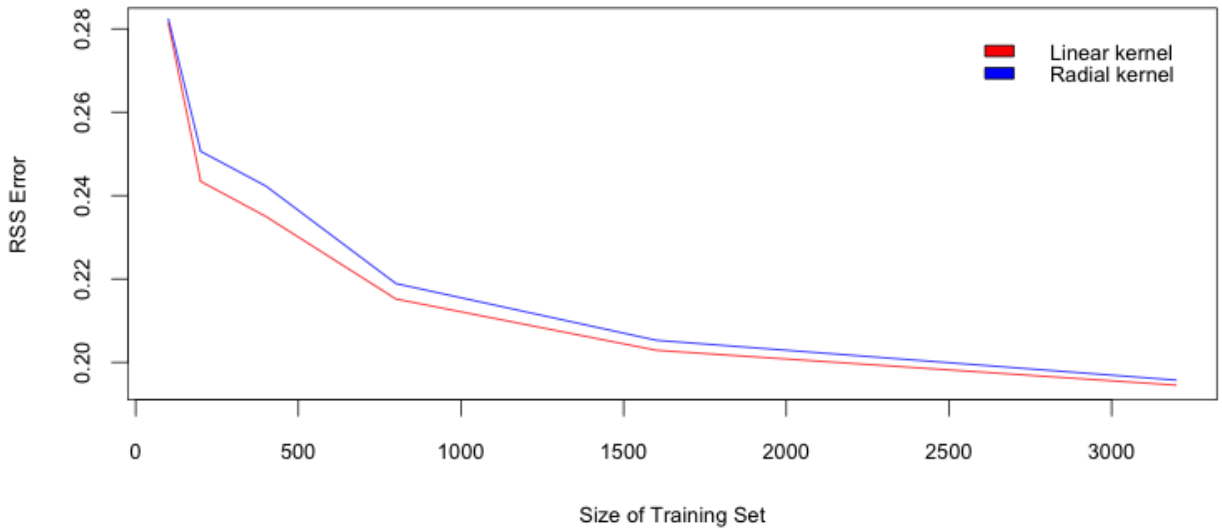
3.3 Random Forests

In typical applications of random forests, it is generally believed that the optimal number of predictors considered per iteration to minimize variance is around $\sqrt{p}$, where p is the total number of predictors. Interestingly enough, the lowest RSS error was seen in forests that considered exactly 10 predictors per iteration, but the errors of generated random forests overall did not seem to correlate much with this quantity.



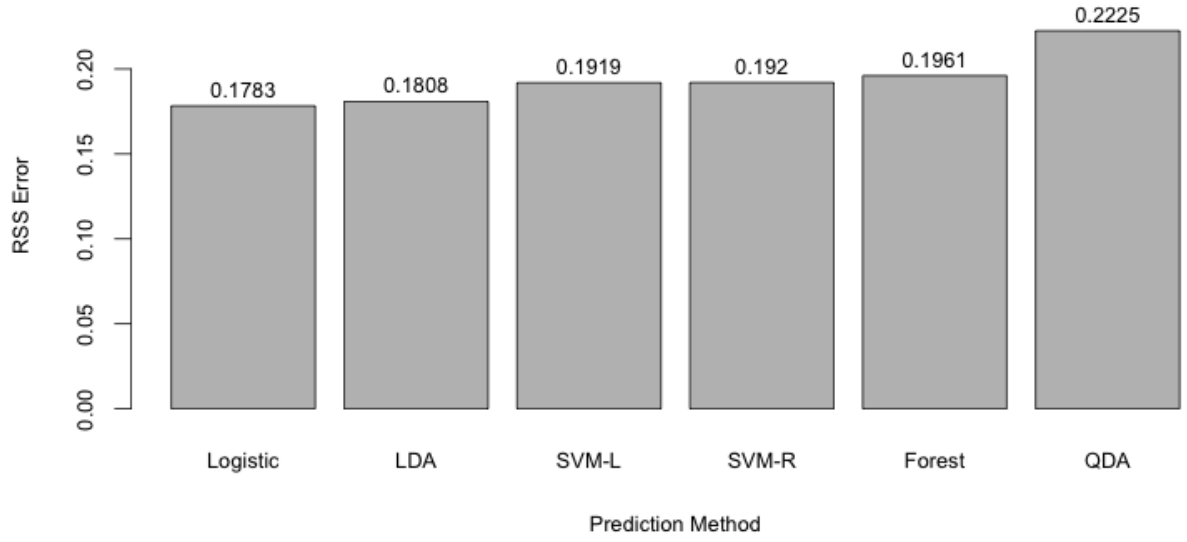
3.4 Support Vector Machines

As it was not computationally feasible to employ support vector machines for training sets larger than 5000 observations, a lot was left to be wanted as this method has often been seen to outperform most others. Below are the results of an investigation into SVM performance given sample size. It appears that SVMs still have room for improvement with larger training sets.



3.5 Comparison of Prediction Models

When comparing all the RSS errors of all six prediction methods, we see that logistic regression with a 0.55 threshold for predicting "high" performed best. LDA with a 0.61 threshold was a close second, while QDA with a 0.99 threshold saw the worst performance. Interestingly, support vector machines with a linear kernel and a radial kernel performed similarly. Random forests that considered 10 predictors per iteration performed worse, but not by much.



4 Discussion

4.1 What Was Learned

It did not come as a surprise that in the second data cleaning, variables corresponding to cues that would unmistakably imply a certain sentiment or suggested a very particular feeling about one's experience saw a value of θ_{cue} far greater than 1. For example, it would be rare to see the word "tasteless" used in any good review, as the word itself implies that the restaurant's very craft is sorely lacking. Variants of the string "not be back" also suggest that the user likely had a very poor experience; otherwise, he or she would not make it a point to place emphasis on the fact that he or she will not be giving the business a second chance.

Both the strings "love" and "like" were also included in the preliminary data set, and they saw values of θ_{cue} of 2.07 and 1.37, respectively. This is interesting as the two words are often thought to be very similar in meaning, with the former simply being a more emphatic version of the latter. As "love" was deemed a good enough predictor whereas "like" was not, this stage of the study lended quantifiable evidence to support this belief, showing well the true power of statistical analysis.

Another very interesting find was that variants of the strings "one stars," "two stars," and "three stars" saw values of θ_{cue} high enough to consider them good predictors but variants of the strings "four stars" and "five stars" did not. It should be noted that users will occasionally rate certain parts of a restaurant experience separately (e.g., food, service, atmosphere, etc.). This may suggest that simply because a restaurant excels in one area does not guarantee happy customers. Another meaning this can take is that one weakness among a restaurant's virtues could very easily be noticed by a critical patron. Fitting a models by logistic regression

LDA also produced a list of strong and weak predictors, signified by relative scalings of coefficients. One of the best things about these list is that the scalings can be positive or negative, which indicate whether they lean towards a higher or lower rating, respectively. It is clear that many of the predictors still left over after the final cleaning suggest a negative sentiment. This makes some sense as Yelp ratings are positive overall, meaning that users tend to rate their experiences four or more stars more often than three or less. It may be a lot easier to identify a bad experience out of a set of good ones than the other way around.

4.2 Final Thoughts

The one method that was not used in this study but was considered in the original plans was boosting applied to decision trees. It was thought to have a lot of potential, as decision trees are great for interpretation and the added power of boosting improves its often subpar performance greatly. However, this would not have been feasible without a great deal of extra computing power.

It was very surprising to see that QDA produced similar error rates despite varying "high" rating threshold values. This could either mean that these models are often dead set on either "high" or "low," or that an error occured at some point of the analysis.

Lastly, it was very disheartening to find that larger sample sizes could not have been used to fit models via decision trees or support vector machines. In a separate analysis, it was found that logistic regression, LDA, and QDA failed to improve much more in performance when the size of the training set was increased past 3000. However, it does seem that support vector machines are still capable of reducing RSS error beyond 3200 training observations, so it is unclear at this point whether these methods could actually produce the best prediction model. There were also very high hopes for decision trees, as it is often argued that they mirror human decision making more accurately than other prediction methods. If, in the future, more resources become available, it would definitely be worthwhile to perform a follow-up investigation of this matter.