# Yelp Project Analysis

*April* $30^{th}$, 2020

*Nathan Kolbow*

*Peer Rating: 1.5*

## 1 Introduction

The Yelp dataset consists of millions of reviews and hundreds of thousands of restaurants; a star rating of either 1, 2, 3, 4, or 5 accompanies each review. For this project, we were given a subset of some 92,236 reviews of 1,361 businesses in the Madison area. The data were partitioned into three subsets: 60% reserved for training, 20% for testing, and 20% for final validation. Each review consists of the full review text along with other miscellaneous categories[1]. The goal of this project was to create the best possible model[2] for predicting the review's rating given only these parameters. Models were scored based on the root mean square error (RMSE) of their predictions of the validation data subset ($\sqrt{\frac{1}{n}\sum[(prediction - actual)^2]}$). We found that the best such model was linear and utilized cross-validation to protect against overfitting.

## 2 Predictor Generation and Selection

First, the entire subset of training reviews was sifted through to create a dictionary of every word mentioned[3]. With this, a contingency table was created consisting of each review and the number of times each word appeared in that review; it is worth noting that this table was *extremely* sparse. To whittle down this list of words, Fisher's exact test was run on every word in the contingency table; the 7,000 most significant words were kept. This test can be unreliable on sparse data like ours[4], so it was only used for this initial screening.

Next, the collinearity and interaction effects of the provided predictors were analyzed. As shown in Figure 1, there is moderate collinearity between useful, funny, and cool, but it is weak enough to be ignored. The number of characters in a review (nchar) and the number of words in a review (nword) have strong collinearity, though. To avoid any intricacies, nchar was used as a potential predictor, but nword was not. Interaction effects between word counts in reviews were not analyzed because the number of possible combinations makes the task computationally infeasible in our setting. Figure 2 shows the only interaction plots that showed evidence of potential interaction effects[5]; non-parallel lines in interaction plots indicate possible interaction effects. None of the other plots showed evidence of any interaction effects.

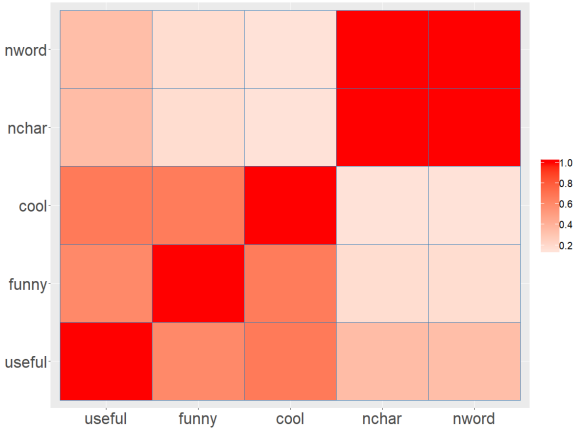


Figure 1: Correlation Heatmap

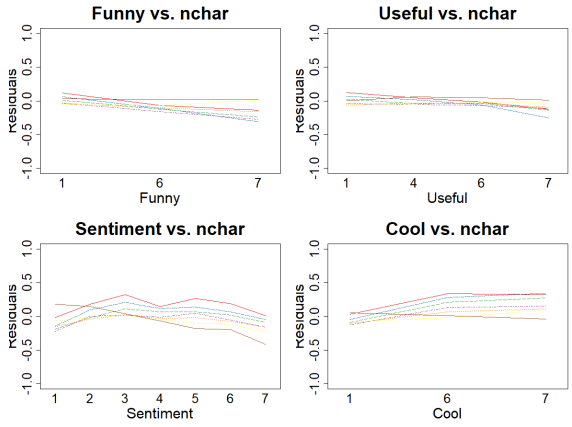


Figure 2: Interaction Charts

One of the most common negations used in English is the word 'not'. Our model only uses counts of individual words, not phrases, so we decided to include interaction effects between 'not' and every other word. There were thousands of these combinations, so they weren't analyzed individually. Instead, all of these interactions were included in our initial model and we allowed our predictor selection method to determine which, if any, of the effects were significant[6].

[1] This includes the number of users who rated the review cool, useful, or funny, as well as the text's sentiment as determined by the AFINN lexicon; these predictors will now be referred to as "cool", "useful", "funny" and "sentiment"

[2] With the restriction that the model must not exceed 3,000 predictors

[3] 45,911 words in total

[4] Renter, D. G.; Higgins, J. J.; and Sargeant, J. M. (2000). "PERFORMANCE OF THE EXACT AND CHI-SQUARE TESTS ON SPARSE CONTINGENCY TABLES," *Conference on Applied Statistics in Agriculture.*

[5] In these plots, data were binned before graphing so that trends would be identifiable

[6] 737 such interaction effects ended up in the final model, confirming our intuition

At this point, we had ~14,000 predictors. The best way to choose predictors would have been an exhaustive search with RMSE as the criterion, but this search would have to go through $2^{14000} = 2.6e4214$ possible models, which is infeasible. We did try two different sets of forward and backward stepwise regression, though, one with AIC as the criterion and one with BIC, but this also took too long to compute. Due to these computational restrictions, we used Lasso regression cross-validation with 5 folds for predictor selection[7]. This method has the strength of selecting predictors based on their out-of-sample significance which helps protect our final model from overfitting. For this reason, given the sparsity of our data, Lasso cross-validation may be the best of the aforementioned choices.

## 3 Model Selection and Diagnostics

Star ratings are categorical variables, so a multinomial regression model seemed most appropriate. We naively fit a multinomial model and were immediately met with a terrible score (RMSE $> 0.95$)[8]. After some analysis we determined that this was due to our choice of a multinomial model; Figures 3 and 4 illustrate this issue. Multinomial regression treats the star rating as a categorical variable, so if two ratings, say 1 star and 4 stars, have similar probability, the model just chooses one. The consequence of an incorrect choice in this case is a very large residual and, based on the size of clusters in Figure 3, this is a problem that our multinomial model ran into often enough to be troublesome.

This issue can be remedied with a more complex multinomial model, but a more intuitive fix is to use a model that treats each review's star rating as a continuous variable. This view of the rating comes with the complication of potential predictions less than 1 or greater than 5; these predictions were simply rounded up and down to 1 and 5 respectively. Figure 4 shows how switching to a linear model, in this case using multinomial linear regression (MLR), significantly improves the model's RMSE. The squared errors aren't clustered as they were before and are now concentrated much more heavily between 0 and 2.5. The MLR model also has very few squared residuals at or above 4, whereas the multinomial model had significant clusters at 4, 9, and 16. These observations led us to the conclusion that a linear model would be most appropriate for this task. Lasso regression with cross-validation has the aforementioned protection against overfitting, as well as the potential benefit of penalized regression, so we decided to use it as our final model.

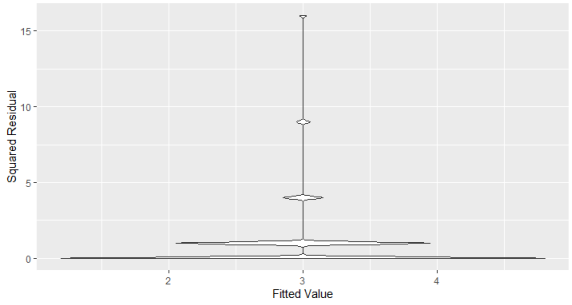


Figure 3: Multinomial Square Residuals Plot

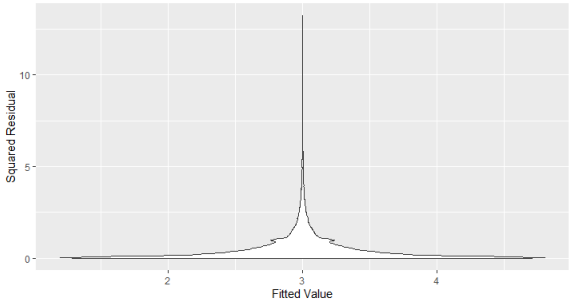


Figure 4: MLR Square Residuals Plot

Three main assumptions need to be met for a linear model: the data must follow a linear model, the variance of residuals must be constant (homoskedasticity), and residuals must be normally distributed. Lasso residuals are not very good for doing diagnostics, so we instead refit the data using an MLR model to perform diagnostics. The linearity assumption is violated; our outcome variable is categorical, so the data follows a multinomial model, not a linear one. As shown earlier, though, we care little about properly categorizing the ratings and more about getting an estimate that comes close to the proper categorization, meaning our desired model representation of the data *is* linear. We will have to keep the categorical nature of the outcome in mind, though, when we evaluate the homoskedasticity of the model's residuals. Figure 5 shows the scale-location plot of the data. The striping seen in the plot is an effect of our estimating a categorical variable as a continuous variable, so it can be ignored. More importantly, the horizontal red line is mostly flat throughout the plot; this means we can be confident that the homoskedasticity assumption

[7]Our final model consisted of 2,725 predictors

[8]A baseline model provided by Professor Shi had an RMSE of 0.897

is met. Finally, Figure 6 shows that the normal QQ plot is very near-linear, which means that we can be confident that our residuals are normally distributed.

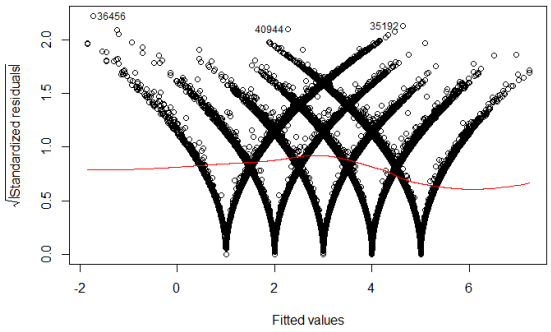


Figure 5: Scale-Location Plot

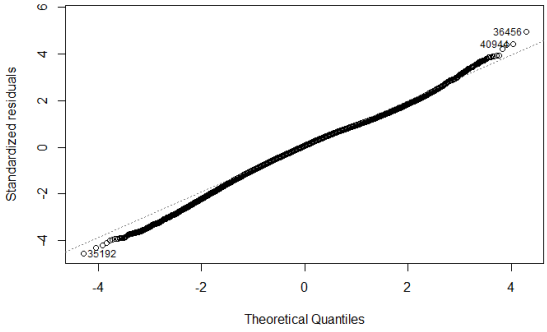


Figure 6: Normal QQ Plot

The final step in finalizing the model was identifying and removing outliers. Figure 7 shows the Cook's distance of each observation, all of which are about 0 except for observations 39,127 and 5,550. With this many data points, an observation with Cook's distance of 0.1 or greater can safely be considered an outlier or overly influential. Observations 5,500 and 39,127 have Cook's distance *significantly* greater than 0.1, so we removed them from the model and reran the Lasso cross-validation; this was our final model[9]. After refitting the data, the optimal penalty ($\lambda$) found by the Lasso regression was 0.00371 with a standard error of 0.00538. This $\lambda$ is nearly 0 and is less than one standard error away from 0, so we cannot say with any significant level of confidence that we benefitted from penalized regression.

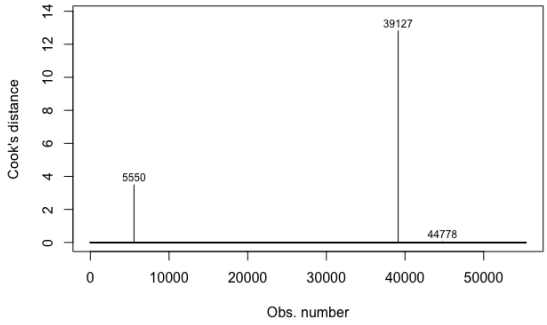


Figure 7: Cook's Distance Plot

All of the predictors whose presence caused the largest impact on a review's rating were words, and interestingly most of these predictors were correlated with a lower star rating. Of the 10 highest impact predictors, 9 of them were correlated with a lower review rating, and of the top 100 such predictors, 72 of them were correlated with a lower review rating, while only 47% of the total predictors were correlated with a lower review rating. This is an observational setup, not an experimental one, so it would be inappropriate to try and connect these observations to a cause. The 5 largest impact predictors were the following, in order: boooo, grass, pants, not:thermostat[10] and surprising. Their corresponding coefficients were $-2.3715$, $-1.360$, $-1.233$, $-1.223$ and $-1.081$, respectively. This means that, for example, all else held constant, each appearance of the word "boooo" in a review decreases that review's predicted star rating by 2.3715 stars. Another interesting coefficient is the intercept: 3.114. This means that if our model was not given any information about a review it would predict that review's rating to be 3.114 stars. This interpretation is somewhat disingenuous, because our model relies so heavily on the text of a review, but it is still interesting to note that the intercept strayed more than half a star from the mean rating[11].

[9]The final model was created with 20 cross-validation folds

[10]Here, the colon indicates an interaction effect, so this predictor is equal to the number of appearances of "not" multiplied by the number of appearances of "thermostat" in the given review

[11]The mean star rating of the training data was 3.713

## 4 Conclusion

Our final model scored an RMSE of 0.7934 on the testing data subset and 0.8016 on the validation data subset[12], so overall the model performs very well. This performance can be attributed in large part to the use of Lasso regression cross-validation. The out-of-sample model building attribute of cross-validation protected us against overfitting our model, and the use of a linear model helped to significantly decrease our model's RMSE. Our model does have multiple weaknesses, though. nchar was found to be a significant variable by our initial Lasso regression, but, as shown in Figure 8, nchar has a very wide range and is heavily skewed. After performing a log transformation on nchar, though, the distribution becomes much nicer. It may have been beneficial to use the log transformation of nchar as opposed to nchar directly.

Another glaring weakness of our model is that it is very simple. Natural language processing is a very difficult task, and counting the appearance of each word in each review barely scrapes the surface of what is possible. A more sophisticated model could make predictions based on more complex patterns in reviews such as phrases, root words, punctuation, grammatical correctness, and so on.

Finally, our model suffers from an inability to give actual categorical estimations, if such behavior were to ever be desired, and is largely a black box. Take a prediction of 3.00 stars, for example. With a linear model, this number is all that you have access to. That same review in a multinomial model with identical predictors could be the consequence of any of the 3 scenarios: 99% probability of being 3 stars, 49% of being 2 stars and 49% of being 4 stars, or 34% chance of being 1 star and 34% chance of being 5 stars. If our model were multinomial, we would be able to extract and analyze the individual predicted probabilities for each level, but with a linear model we do not have access to such insights. Despite these weaknesses, when all that we care about is getting the best possible RMSE, a linear model simply outperforms a multinomial model.

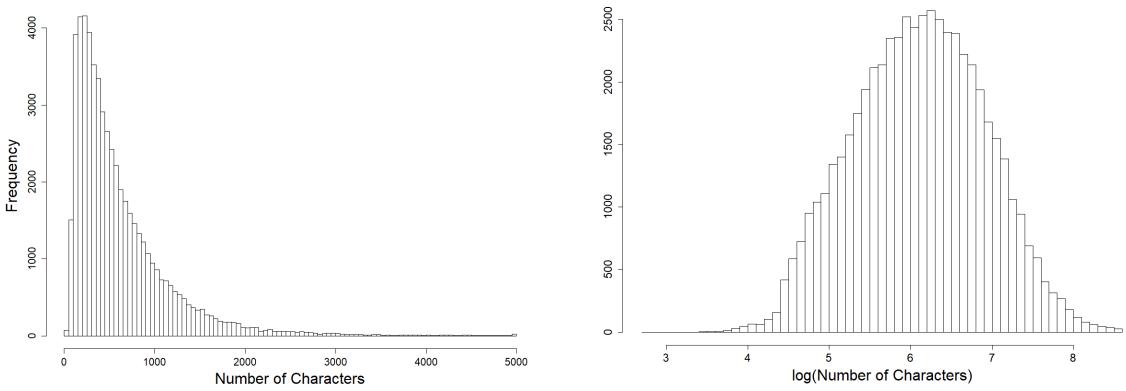


Figure 8: Distributions of nchar and log(nchar)

[12] RMSE $< 0.85$ on both was required for full credit; Professor Shi's benchmark scored 0.8972 and 0.9119 on each, respectively