Stats 101C

Analysis and Prediction of Business Star Rating

By:

Katrina Rivera

UID 004886996, Statistics and Economics Dept.

Casey Truong

UID 905785411, Statistics Dept.

Celeste Vargas Vera

UID 305395537, Statistics Dept.

1. Abstract

Founded in 2004, Yelp is an online website and mobile app that provides users the ability to review businesses.[1] Today, Yelp is one of the most famous review sources with an incredibly large dataset across the U.S. This report focuses on analyzing significant factors that affect business star ratings. To analyze this, we go through some exploratory data analysis, variable selection methods, and supervised models, specifically classification models. The dataset used in this report is originally from the Yelp Dataset, and it is within California and contains 16 analyzed fields and almost 54,000 records.

2. Introduction

Word-of-mouth is a powerful purveyor of new customers, and with the rising popularity of Yelp, the open review website is now considered reliable for individuals to decide which businesses they should visit. How do customers intuitively conclude which businesses are the best? The main key factor is the Star rating. If one business has a 3 or above star rating on Yelp, it is likely to gain trust among new customers. However, under 3 stars rating has been known to ruin a business.[2] The goal of this paper is to find the best model to predict whether a business has a good star rating (4-5 stars) or a bad star rating (1-3 stars), based on other factors present in the data. We will first preprocess the data to contain a balanced dataset with only usable variables. We will then explore the cleaned data set through data charts, correlation charts, and text mining. After, we will perform variable subset selection to determine the most important variables for models. We will finally create a range of supervised models to find the best model to predict a good or bad star rating.

2.1 DATA DESCRIPTION AND DATA WRANGLING

We begin our report by observing the structure of our data. We found that the original data table consists of 18 fields, with 16 fields being analyzable. Some variables in the data set are below:

- Star: the number of stars that users give to a business
- Useful: reviews from users that are rated as “Useful” by the online community
- Cool: reviews from users that are rated as “Cool” by the online community
- Funny: reviews from users that are rated as “Funny” by the online community
- Elite: whether a user belong to the Elite subscription[3]
- Bus_Ave_Star: a business average star from all time
- User_Review_count: the number of reviews a business receives
- User_Useful_count: the number of “useful” reviews from users

Upon first inspection, we observed 11427 missing values under the Elite variable. We also noticed the Elite column consisted of different dates in several rows. To ease our understanding of the Elite column, we decided to make it binary, replacing NA’s with 0 and non-missing entries with 1. We also changed the Star feature into binary, where 0 refers to star ratings from 1-3, and 1 refers to star ratings from 4-5. The City predictor was transformed into a factor. We removed the variables Unnamed, User_id, and Bus_id since they are not relevant for data analysis. Moreover, we also removed the State column because the entries were all uniformly “California”.

After the general pre-processing, we created a new dataframe containing only the variables that can be analyzed. These are the variables Useful, Cool, Funny, Bus_Ave_Star, User_Review_count, Elite, User_Useful_count, User_Funny_count,User_Cool_count, User_Fans, and Users_Ave_Star.

2.2 DATA BALANCING AND SPLITTING

While inspecting the data, we noticed our dataset is imbalanced: there are more high star ratings than low star ratings. To prevent the issue of overfitting or creating biases, we balanced our data by equating the amount of 0’s and 1’s in the Star column, using the bootstrap method with the function createDataPartition in R. The training dataset contains 70% of the original dataset, and the test data set consists of the remaining 30% of the original dataset. The data set that we analyzed contained 27802 records and 13 fields.

---

[1] “Fast Facts,” Yelp (Yelp), accessed December 1, 2022. https://www.yelp-press.com/company/fast-facts/default.aspx.

[2] "Restaurant Manager Says Yelp Is Killing His Business." 25 Apr. 2013. Accessed 1 Dec. 2022. https://www.businessinsider.com/owner-yelp-is-bad-for-small-business-2013-4.

[3] Taylor Lyles, “How to Become a Yelp Elite Member and Get Exclusive Perks from the Service,” Business Insider (Business Insider), accessed December 1, 2022. https://www.businessinsider.com/guides/tech/how-to-become-yelp-elite.

3. Exploratory Data Analysis

For data exploration, we are going to create exploratory data charts of the balanced train data set. We will then explore the response variable via correlation analysis, and lastly, we will explore the text column “Review” through text mining and sentiment analysis.

3.1. Exploratory Data Charts

We decided to begin by creating a couple of pie graphs and bar charts. The first pie chart (left) shows the percentage of businesses that received a good star rating (1) or a bad star rating (0), where we can see that 49.64% of businesses received a bad star rating, and 50.36% of businesses received a good star rating. We confirm a balance in our dataset through the near equal split of 0’s and 1’s. The second pie chart (middle) consists of the proportion of users that have had elite status (78%) or have not had elite status (22%). The third pie chart (right) compares the proportion of Useful, Cool, and Funny rated reviews. Here we observed 43.6% of reviews were rated useful, 23.4% were rated funny, and 23.4% were rated cool. Moving onto our bar charts, the first bar plot (right) shows the number of businesses across different cities in California, with Santa Barbara, Goleta and Carpinteria as the leading cities. The second bar plot (right) shows the number of mean star ratings received by city. West Hill, Sparks and South Lake Tahoe are the top three cities with the highest star rating means.

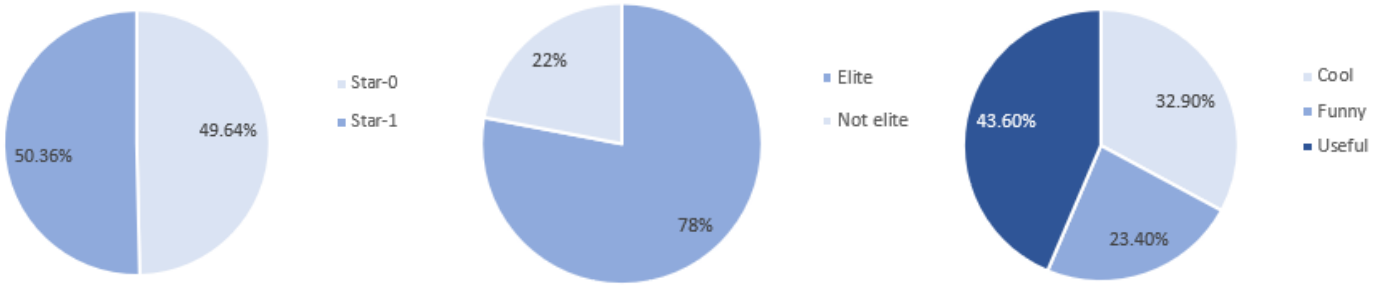

Figure 1. Descriptive Stats Pie Charts: Star Rating (far left), Elite (middle), Rated Reviews (far right)

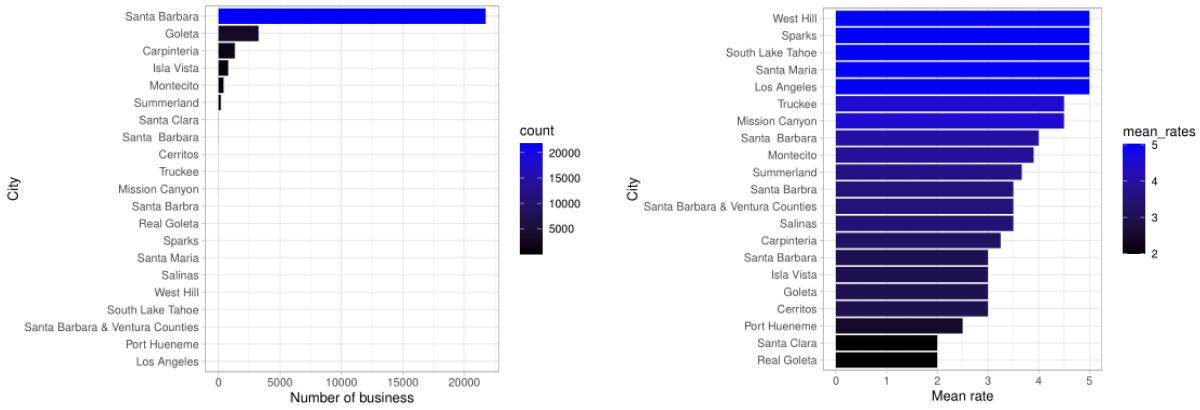

Figure 2. Descriptive Stats Bar Plots: Yelp Businesses and Cities (left) and Star Rating and Cities (right)

3.2. Exploration of Response Variable

3.2.1. CORRELATION

To learn more about our response variable, we observed the correlation between the predictors and the response variable Star could be used to learn about their interactions.

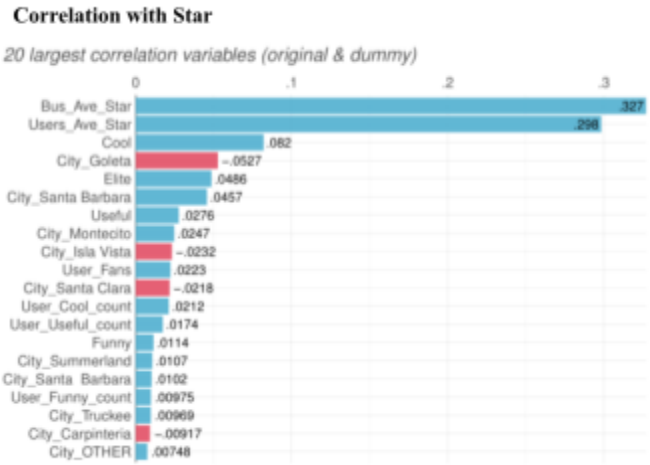

Figure 3. Correlation Plot with Response Variable (Star)

Based on the correlation bar graph, we observed that Bus_Ave_Star, Users_Ave_Star, Cool, Elite, City_Santa Barbara, and Useful have among the highest positive correlations to the response, leading us to suspect some of these columns might not be useful during our modeling. In the future, we delve into feature selection methods to help us choose the most optimal features.

3.3 Exploration of Review Column

3.3.1. TEXT MINING AND SENTIMENT ANALYSIS

To explore and analyze the Review column of the dataset, we decided to employ Text Mining to find word frequency; we then performed Sentiment Analysis to evaluate customer review attitudes about the establishments they visited.

We prepare the data by lower casing the Review column, followed by the tokenization, removal of stopwords, and lemmatization of the column's contents. To help us prepare the data, we use the nltk package in Python.[4] The following were the most frequently used words, with "good" coming in as number one.

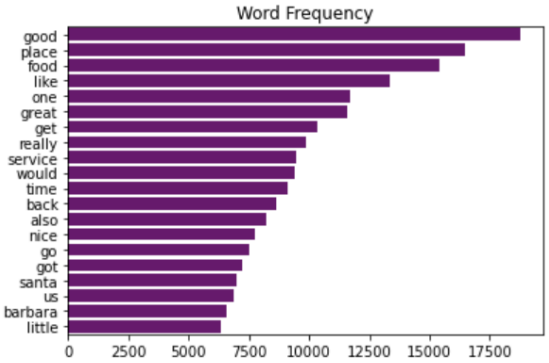

Figure 4. Bar chart depicting most frequent words

To perform sentiment analysis, we used the SentimentIntensityAnalyzer() function on the lemmatized review column to find the polarity of the words written. We continue by finding the sentiment of these words, categorizing them as Positive (word polarity compound > 0), Neutral (word polarity compound = 0), or Negative (word polarity compound < 0). Most words were positive. From

[4] Jan Kirenz, "Text Mining and Sentiment Analysis with NLTK and Pandas in Python," Text Mining and Sentiment Analysis with NLTK and pandas in Python (Hugo, June 16, 2022), https://www.kirenz.com/post/2021-12-11-text-mining-and-sentiment-analysis-with-nltk-and-pandas-in-python/text-mining-and-sentiment-analysis-with-nltk-and-pandas-in-python/.

our brief analysis of the Review column, we learned that even the most negative reviews had positive words in them, leading us to believe that to perform a predictive model based on the Review column, we need to do more research on models that can use context to successfully predict business Star ratings; this is a limitation we can work towards in the future. See counts for each sentiment category below:

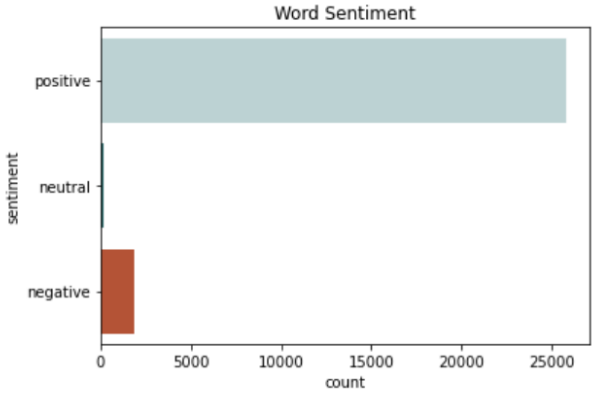

Figure 5. Bar chart depicting the sentiment word count of reviews

4. Feature Selection

For our feature selection, we first employed a brief multicollinearity analysis to give us an idea of the variables that would probably be removed in different feature selection methods. As our actual variable selection methods, we decided to perform a subset selection method, stepwise selection methods, and ridge and lasso regression to find which is the best performing method for our feature selection. We will then choose the variables based on the best performing feature selection method.

4.1.1. MULTICOLLINEARITY

Before beginning with variable subset selection methods, we observed the multicollinearity between variables. This is to see what variable selection methods we need to perform. We applied correlation plots and VIF scores to check the most significant variables. Any variables with the VIF score greater than 5 are highly correlated. We saw that the variables with a VIF of 5 or higher are User_Useful_count, User_Cool_count, Useful, Cool, and Funny. The correlation plots below show that the predictors highly correlated with each other are Useful, Cool, and Funny; City and User_Fans; and User_Useful_count, User_Funny_count, User_Fans, and User_Cool_count. This indicates that we need to perform lasso and ridge regression methods, since these methods take into account multicollinearity. We will also perform subset and stepwise selection methods for comparison.

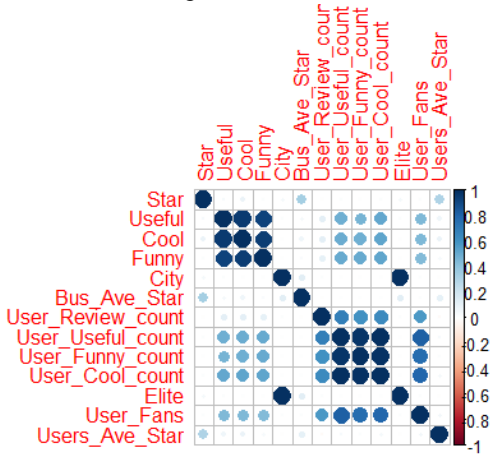

Figure 6. Correlation plot with all variables

4.1.2. SUBSET SELECTION METHOD

Our first variable selection method after observing multicollinearity is exhaustive method selection. Exhaustive selection compares all possible models for each number of predictions. To visualize our results, we plotted the optimal number of variables that minimizes RSS, Cp, and BIC, and that maximizes adjusted r-squared. The results showed that 14 predictors optimized RSS, Cp, and R-squared, while 8 predictors minimized BIC. However, observing the plots for RSS, adjusted R-squared, and Cp showed that the change for these plots was minimal after 8 predictors. From the exhaustive method selection, we found that the optimal 8-variable model contains the variables: Useful, Cool, Funny, Bus_Ave_Star, User_Funny_count, User_Cool_count, Elite, and Users_Ave_Star.

4.1.3 STEPWISE SELECTION METHODS

For our first stepwise selection method, we employed a forward selection method, for which results were the same as exhaustive selection: 14 predictors were optimal for RSS, Cp and Adjusted R-squared, and 8 predictors were optimal for BIC. However, the change for these variables was instead minimal after 6 predictors. In this case, the best 6-variable model contains the variables: Useful, Cool, Funny, Bus_Ave_Star, Elite, and Users_Ave_Star.

We performed a backward selection model, which resulted in 14 predictors being optimal for RSS, adjusted r-squared, and Cp, while 6 predictors were optimal for BIC. The change for these variables was also small after 6 predictors, and the best 6 variable model was consistent with the forward selection method.

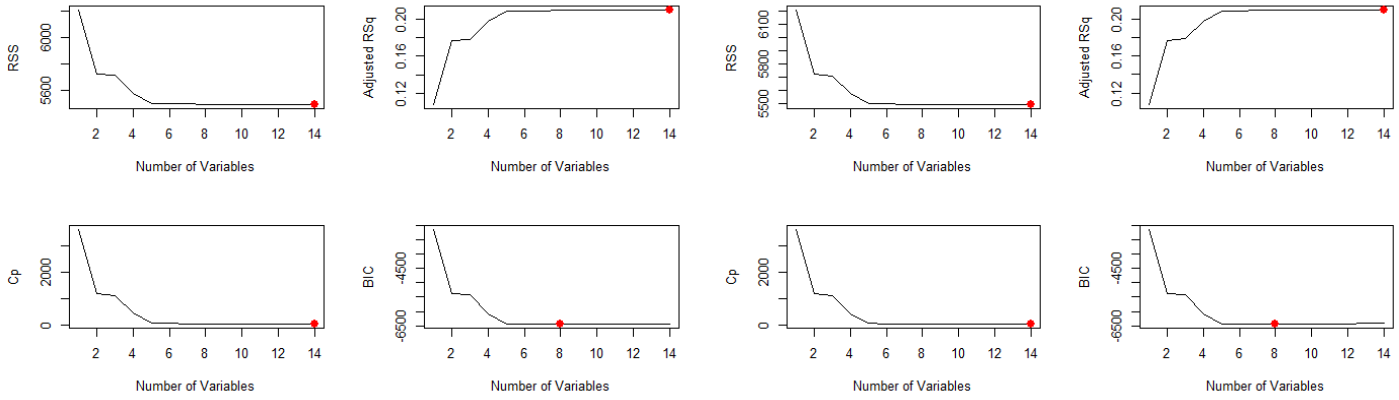

Figure 7. Diagnostic plots of RSS, adjusted r-squared, Cp, and BIC for exhaustive regression (left) and forward regression (right)

4.1.4. RIDGE AND LASSO REGRESSION

Due to multicollinearity issues found earlier, we needed to perform ridge and lasso regression. We began with ridge regression. After running our ridge regression, we plotted our regression and saw that when log-Lambda equals to 5, all coefficients approach 0. To find the best value for lambda, we used a 10-fold cross-validation, where we found that the best lambda that minimized the test MSE was equal to .01, where log-Lambda was equaled to -4.60517. When lambda equals to .01, the variables that had a slope that was not approximately 0 were: City, Useful, Cool, Funny, Bus_Ave_Star, Elite, and Users_Ave_Star. The test MSE for lambda equal to .01 is 1.193133.

Like ridge regression, the lambda value that minimizes MSE for lasso regression was also .01. The number of optimal predictors was 6 predictors: Useful, Cool, Funny, Bus_Ave_Star, Elite, and Users_Ave_Star. These were the same variables we found using stepwise variable selection; these are the same predictors found using ridge regression with the exception for the variable “City”. However, the “City” variable does not have enough observations to include all city factors for both training and testing datasets. We could not include this variable in our regression models. Therefore, ridge and lasso regression identified the same variables that can be used in modeling. Unlike the ridge regression, the lasso model reduces the number of predictors by setting the slope of the variables equal to 0. The test MSE of this model was 1.198229, which was similar to the ridge regression MSE.

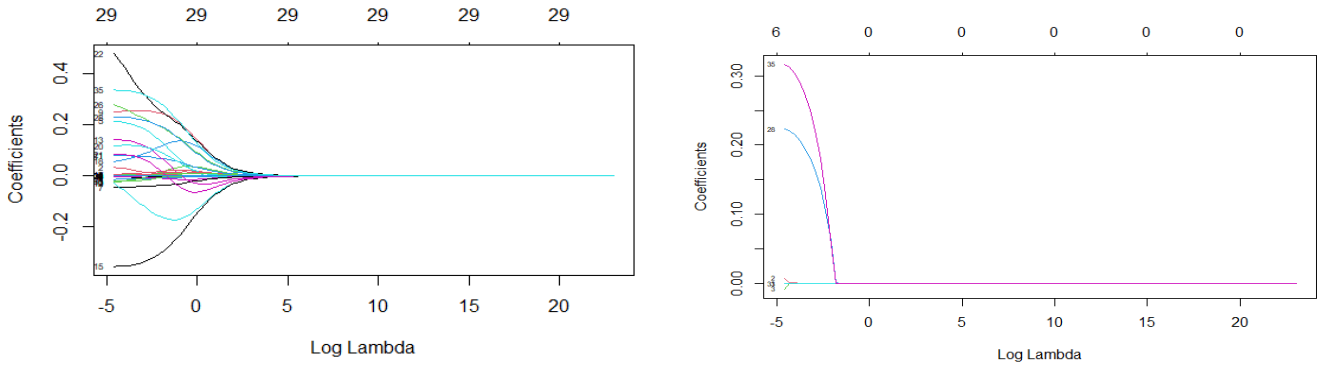

Figure 8. Plot for Ridge Regression (right) and Lasso Regression (left)

We observed that ridge and lasso regression found the same usable predictors. As for our final variable selection, since stepwise and lasso regression found the smallest number of variables, we will use the variables found in stepwise and lasso regression in our predictive models. These variables are: Useful, Cool, Funny, Bus_Ave_Star, Elite, and Users_Ave_Star. This will allow us to compare the accuracy and true positive rates of each model to determine the most important variables in our models.

5. Methods

For our methods, we employed a variety of classification models and support vector models to determine the best model to predict business star ratings.

5.1 Classification models

The classification models we used were Logistic models, LDA and QDA, K-Nearest Neighbors, and a Random Forest Classifier.

5.1.1. LOGISTIC MODELING

We fit our logistic model with the training data created in pre-processing and then compare it to the testing data to calculate the confusion matrix. Our logistic model contained the 6 variables identified with forward selection, lasso regression, and backward selection: Useful, Cool, Funny, Bus_Ave_Star, Elite, and Users_Ave_Star. Using 10-fold cross validation, we found that the accuracy rate was 0.69574 and the true positive rate was 0.716914. Using ROC as a metric, the most important variables in the logistic regression were Bus_Ave_Star, Users_Ave_Star, Cool, Useful, and Funny. The variable with the highest importance was Bus_Ave_Star.

5.1.2. LDA AND QDA

LDA assumes that the predictors within each Star class are drawn from a multivariate normal density with individual mean and shared covariance matrix. The training data used for the LDA model was scaled and centered since LDA is sensitive to non-normalized data.[5] The LDA model with 6-predictor had an accuracy of 0.6916 and had a true positive rate of 0.7346 using 10-fold cross-validation. Using ROC as a metric, the most important variables in the LDA regression were Bus_Ave_Star, Users_Ave_Star, Cool, Elite, and Funny.[6] The variable with the highest importance was Bus_Ave_Star.

QDA assumes that the log odds of Y (which is variable *Star*) given *x* are quadratic, and it assumes that the covariance matrix is different for each predictor within each *Star* class.[7] The QDA model uses normalized training data as well. Using the same 6-predictors from LDA, we found that the QDA model has an accuracy of 0.68279 and a true positive rate of 0.62951 using 10-fold

[5] Gareth James et al., *An Introduction to Statistical Learning: With Applications in R*, 2nd ed., 2021.
[6] De Silva, Kushan. "Machine Learning with Linear and Quadratic Discriminant Analyses, Binary Logistic Regression, Cross Validations, Model Selection, ROC Curves and AUC." RPubs, July 30, 2017. https://rpubs.com/Kushan/295412.
[7] Gareth James et al., *An Introduction to Statistical Learning: With Applications in R*, 2nd ed., 2021.

cross validation. Using ROC as a metric, the most important numerical variables in the LDA regression were Bus_Ave_Star, Users_Ave_Star, Cool, Elite, and Funny. The variable with the highest importance was Bus_Ave_Star.

5.1.3. KNN

Third, we used a non-parametric approach to model the data. Unlike logistic, LDA, or QDA regression, KNN makes no assumptions about the posterior distribution given x.[8] We performed a 10-fold cross-validation to obtain the most optimal number of neighbors, which in this case were 9, so we used k = 9. We used normalized data for this model because KNN calculates euclidean distance using the predictor values. If one predictor is larger in scale, then it will dominate the distance of the KNN and therefore create biased results.[9] Using a 10-fold cross validation on the test dataset KNN fit, the accuracy of KNN was 0.6832, and the true positive rate was 0.703467. Using ROC as a metric, the most important variables in the KNN regression were Bus_Ave_Star, Users_Ave_Star, Cool, Elite, and Funny.[10] The variable with the highest importance was Bus_Ave_Star.

5.1.4. RANDOM FOREST CLASSIFIER

The last classification model we performed was a Random Forest Classifier. When compared to other models, a Random Forest uses uncorrelated Decision Trees to perform classification, giving us a more precise model compared to one Decision Tree. For this model, we did not use the Review column. From our variable selection, we used the features: Useful, Cool, Funny, Bus_Ave_Star, Elite, Users_Ave_Star, supporting the results from Lasso and stepwise variable selection. To help us employ this model, we used the sklearn package in Python.

We trained our model using 50 to 400 trees, skipping every 50 as a rule of thumb, using a 10-fold cross-validation. We found that the optimal number of trees is 250, and we observed approximately 85.68% train model accuracy. To test the model's performance with 250 trees, we used the test portion of the data as the validation. Using a 10-fold cross validation, we found that the average model performance was around 66% with a true positive rate also close to 66%. This means that our accuracy was most likely due to overfitting; this causes Random Forests to be weaker when trying to predict data.[11]

We also found the Variable Importance estimated by the Random Forest. We found that the most important factors when predicting a review's star were Users_Ave_Star, Bus_Ave_Star, Useful, Cool, Funny.

5.2 Support Vector Classification (SVC) and Support Vector Machines (SVM)

In order to perform SVM, we needed to reduce the number of training data to input into the model. In this case, we randomly sampled 800 data points for our training data. We also based the SVM model off the 6 predictors we used for the classification models: Users_Ave_Star, User_Review_count, Elite, Bus_Ave_Star, Cool, and Useful. This was to prevent our training time from growing too large.[12]

We begin our analysis with a support vector classifier model (SVC). For this model, we assume that there is a linear hyperplane that can separate the two classes of the *Star* variable.[13] We found C using a tuning grid, which contains the C values .25,.5, 1, 5, 8, 12, and 100. We found the best C using repeated k-fold cross validation with 5 folds repeated 4 times. The best C was 100; this allows a wide margin to include many support vectors. This accounts for higher bias but lower variance. We were using a very small amount of training data points to fit this model in comparison to the given dataset. This means that this model will be robust to data points that are very far away from the hyperplane.[14] The accuracy of the support vector classifier model with C equal to 100 is 0.6915043; the true positive rate was 0.6825035.

---

[8] Ibid.
[9] Ibid.
[10] De Silva, Kushan. "Machine Learning with Linear and Quadratic Discriminant Analyses, Binary Logistic Regression, Cross Validations, Model Selection, ROC Curves and AUC." RPubs, July 30, 2017. https://rpubs.com/Kushan/295412.
[11] Šiklar, Martin. "Why Is My Model Performing Poorly?" Medium. Towards Data Science, June 12, 2021. https://towardsdatascience.com/why-is-my-model-performing-poorly-b4be05ad3ec6.
[12] Wei Hao Khoong, "When Do Support Vector Machines Fail?," Medium (Towards Data Science), August, 2021. https://towardsdatascience.com/when-do-support-vector-machines-fail-3f23295ebef2.
[13] Gareth James et al., *An Introduction to Statistical Learning: With Applications in R*, 2nd ed., 2021
[14] Ibid.

We also calculated a support vector machine model (SVM). We assume that there is a non-linear hyperplane that can separate the two classes of the *Star* variable. In this model, we chose a radial kernel to model the data. We used cross-fold validation with 10 folds to find the best gamma and C value to minimize the training error. The potential gammas were .5, 1, 2, 3, and 4; the potential C's are .1, 1, 10, 100, 1000. The best C and gamma were C = 1 and gamma = .5. The accuracy of this model was 0.6787645; the true positive rate was 0.6428571.

6. Results

6.1 Variable selection

We can see that our forward, backward, lasso, and ridge methods identified the same best, usable 6 variables. Despite the stepwise and subset models identifying the optimal number of predictors to be large, the diagnostic plots for forward, backward, and exhaustive selection show that models with more than 6-8 predictors have little change in BIC, Cp, adjusted r-squared, and RSS. This indicates that these 6 variables were the best predictors to use in our models, such as logistic, LDA and QDA models, KNN, and Random Forest.

6.2 Classification models

The logistic model has an accuracy rate of .6957 and a true positive rate of .7169; the LDA model has an accuracy rate of 0.6916 and a true positive rate of 0.7346; the QDA model has an accuracy rate of 0.68279 and a true positive rate of 0.62951. The logistic and LDA models have nearly equal predictive power. We can see that QDA has a lower accuracy and true positive rate compared to the LDA and logistic models. This is due to overfitting of the data, causing a decrease in accuracy and true positive rate. The KNN model has an accuracy of 0.6832 and the true positive rate of 0.703467. KNN has a higher accuracy and true positive rate compared to QDA, but it has both lower accuracy and true positive rate compared to the LDA and logistic models. We believe that this is also due to overfitting, although not to the extent of the QDA model. Since both non-linear models performed worse compared to LDA and logistic models, we can conclude that this data is better predicted using a linear separator model. Additionally, since KNN performed better than QDA, we can conclude that the data cannot be separated with a nonlinear model. We can conclude this because the KNN model can roughly approximate a linear model because there are a large number of data points in this dataset.

The logistic and LDA model have very similar predictive power. The accuracy of the logistic model is slightly higher than the LDA model, but the LDA has a higher true positive rate compared to logistic. In order to compare the two, we calculate the harmonic mean of precision and recall. Recall is the true positive rate. Precision is the proportion of true positives that were identified correctly out of the positive identifications. There is a tradeoff between recall and precision, where an increase in recall will decrease precision. To take both of these calculations into consideration for comparison, we use the harmonic mean, which is a weighted average of recall and precision. It is found with the formula: $2 * \frac{precision * recall}{precision + recall}$[15]. The harmonic mean for logistic is 0.705431. The harmonic mean for LDA is 0.6993. We can therefore conclude that the logistic is the better model for this dataset. Since the logistic model is a discriminative model, maximizing the conditional likelihood of Star given the prior probabilities is more effective than maximizing the joint likelihood of our response variable and prior probabilities.[16]

From the Random Forest Classification, we obtained a model accuracy of 66%. This is not an improvement over logistic, LDA, or QDA due to overfitting.[17] However, using Random Forest we found that the most important variables that predicted the star rating of reviews were in order: average stars users gave, average business stars and whether the review given was useful, cool or funny.

We can conclude that the best model from the classification models is logistic. Additionally, from observing the similar predictive power of the logistic and LDA models, we can conclude that a linear model better separates the data. From logistic model assumptions, we can assume that there is a linear relationship between the predictors and the log-odds, there are no extreme outliers,

[15] Urwin, Matthew, and Will Koehrsen. "Use Precision and Recall to Evaluate Your Classification Model When Accuracy Isn't Enough." Edited by Sadrach Pierre. Use Precision and Recall to Evaluate Your Classification Model When Accuracy Isn't Enough. builtin, August 8, 2022. https://builtin.com/data-science/precision-and-recall.
[16] Ibid.
[17] Šiklar, Martin. "Why Is My Model Performing Poorly?" Medium. Towards Data Science, June 12, 2021. https://towardsdatascience.com/why-is-my-model-performing-poorly-b4be05ad3ec6.

and that observations are independent.[18] This means that a rating given to a business by one user does not affect the rating given to a business by another user.

6.3 Support Vector Classification (SVC) and Support Vector Machines (SVM)

Similar to our analysis of Logistic, LDA and QDA, overall, SVM and support vector classification (SVC) have similar predictive power. However, SVM has a slightly higher accuracy rate but lower true positive rate compared to SVC. Since we have balanced this dataset, this result is due to the shape of the model that separates the two classes. Between these two models, we choose a SVC model to fit this data since it has better overall performance. This gives additional support to the assumption that linear models can better separate the classes of the Star instead of non-linear models (like models with a radial shape). [19] One limitation is that we did not try multiple different non-linear SVM models, like polynomial SVM models. This would give more insight into whether a linear classification model has better predictive power than a non-linear model.

However, SVC does not have a higher accuracy nor true positive rate compared to LDA, and SVM does not have a higher accuracy nor true positive rate compared to QDA. Neither model is better than logistic. This indicates that a better model for this data should focus on all data points instead of only points that are difficult to classify (support vectors). This can be due to a relatively low level of outliers in the data.[20]

7. Conclusions

In conclusion, we find that the best overall performance model to predict review score ratings is logistic. Since the logistic model is our best performing model, we decided to find the variable importance based on our logistic model. We found that the most important variable is the average stars per business (Bus_Ave_Star), followed by the average number of stars given by users (Users_Ave_Star), and whether users found reviews cool, funny, or useful (Cool, Funny, Useful). The most important variable found in the logistic model, Bus_Ave_Star, is also found to be the most important variable for the LDA, QDA, KNN, and random forests models. This means that the most important variable for predicting the Star rating of a business, no matter the model, is the average number of stars a business receives. If we were to advise a business on what they should focus on, we would suggest trying to maintain a good average star rating online to attract better online ratings from users. During our word frequency analysis, we found the most used word was “good”, so maintaining a good star rating and perpetuating a good first impression with clients is the first step to improving star rating.

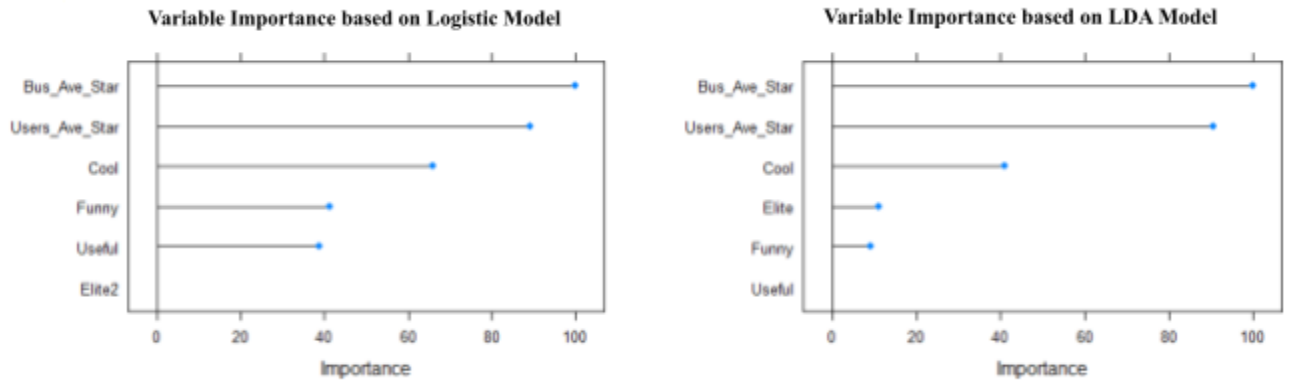

Figure 9. Importance plot describing variable importance obtained from logistic (left) and LDA (right) using ROC

[18] Kenneth Leung, “Assumptions of Logistic Regression, Clearly Explained,” Assumptions of Logistic Regression, Clearly Explained (Medium, September 13, 2022), https://towardsdatascience.com/assumptions-of-logistic-regression-clearly-explained-44d85a22b290.
[19] Gareth James et al., *An Introduction to Statistical Learning: With Applications in R*, 2nd ed., 2021.
[20] Ibid.

References

De Silva, Kushan. “Machine Learning with Linear and Quadratic Discriminant Analyses, Binary Logistic Regression, Cross Validations, Model Selection, ROC Curves and AUC.” RPubs, July 30, 2017. https://rpubs.com/Kushan/295412.

“Cv.glm: Cross-Validation for Generalized Linear Models.” RDocumentation. DataCamp. Accessed December 9, 2022. https://www.rdocumentation.org/packages/boot/versions/1.3-28.1/topics/cv.glm.

“Fast Facts.” Yelp. Yelp. Accessed December 1, 2022. https://www.yelp-press.com/company/fast-facts/default.aspx.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. 2nd ed., 2021.

Khoong, Wei Hao. “When Do Support Vector Machines Fail?” Medium. Towards Data Science, August 29, 2021. https://towardsdatascience.com/when-do-support-vector-machines-fail-3f23295ebef2.

Kirenz, Jan. “Text Mining and Sentiment Analysis with NLTK and Pandas in Python.” Text Mining and Sentiment Analysis with NLTK and pandas in Python. Hugo, June 16, 2022. https://www.kirenz.com/post/2021-12-11-text-mining-and-sentiment-analysis-with-nltk-and-pandas-in-python/text-mining-and-sentiment-analysis-with-nltk-and-pandas-in-python/.

Leung, Kenneth. “Assumptions of Logistic Regression, Clearly Explained.” Assumptions of Logistic Regression, Clearly Explained. Medium, September 13, 2022. https://towardsdatascience.com/assumptions-of-logistic-regression-clearly-explained-44d85a22b290.

Libretexts. “Analysis the Similarity and Difference between SVM, Lda and Qda (Heng Xu).” Statistics LibreTexts. Libretexts, August 17, 2020. https://stats.libretexts.org/Bookshelves/Computing_and_Modeling/RTG%3A_Classification_Methods/4%3A_Numerical_Experiments_and_Real_Data_Analysis/Analysis_the_Similarity_and_Difference_between_SVM%2C_LDA_and_QDA_(Heng_Xu).

Lutz, Ashley. “Restaurant Manager Says Yelp Is Killing His Business.” Business Insider. Business Insider. Accessed December 1, 2022. https://www.businessinsider.com/owner-yelp-is-bad-for-small-business-2013-4.

Lyles, Taylor. “How to Become a Yelp Elite Member and Get Exclusive Perks from the Service.” Business Insider. Business Insider. Accessed December 1, 2022. https://www.businessinsider.com/guides/tech/how-to-become-yelp-elite.

Šiklar, Martin. “Why Is My Model Performing Poorly?” Medium. Towards Data Science, June 12, 2021. https://towardsdatascience.com/why-is-my-model-performing-poorly-b4be05ad3ec6.

Urwin, Matthew, and Will Koehrsen. “Use Precision and Recall to Evaluate Your Classification Model When Accuracy Isn't Enough.” Edited by Sadrach Pierre. Use Precision and Recall to Evaluate Your Classification Model When Accuracy Isn’t Enough. builtin, August 8, 2022. https://builtin.com/data-science/precision-and-recall.