Predicting the Useful Values with Machine Learning Models

[†]Department of Statistics

Gyujin Seo[‡] (UID: 605357614)

Department of Statistics
University of California, Los Angeles
United States of America
12/02/2022

Contents

Abstract

This paper studies reviews of California restaurants on Yelp to figure out which machine learning model gives the best prediction on if a review is voted as "useful". During the pre-processing procedure, I removed unrelated variables, fixed errors in the data, calculated necessary predictors and converted the reviews to numerical variables using Natural Language Processing (NLP). In the Experiment part, I split the data as training data and testing data and used 10-fold cross validation to choose the best parameters of selected models. The results of this paper are demonstrated by the Random Forest Model, Logistic Regression Model, LASSO Regression Model, Decision Tree Model, and the K-Nearest Neighbor Model (KNN). With the Receiver Operating Characteristic (ROC) curve and the Area Under the ROC Curve (AUC), it was concluded that the Random Forest Model performs the best among all the models.

1 Introduction

Yelp is a website that provides a one-stop local platform for consumers with business information, photos and reviews. In Yelp, most things are done by the customers' own experiences. With offering various things to people on Yelp, a useful button plays an important role in determining and reserving stores for people. I decided the useful button as the dependent variable because people decide to press the useful button if it is useful to them by reading the reviews; otherwise, if people thought it is not, they do not press the button. I thought that it would give others more confidence if people pressed a useful button after they experienced, read or thought about whether it was helpful or not. Therefore, I decided to know if I can guess whether it is useful or not even before people start pressing the button.

2 Variables

In the initial dataset, I was given 53845 unique observations, with 17 total predictors to choose from. For the variables, I have User ID, Business ID, Star, Useful, Cool, Funny, Review, State, City, Business average count, User review count, User useful count, User funny count, User cool count, Elite[1], User fans, and User average star.

Variables	Description
User_id	A random code with letters, numbers and special characters “_” and “” that identifies each user
Bus_id	A random code with letters, numbers and special characters “_” and “” that identifies the business
Star	In a scale of 1 to 5, the star represents different levels of satisfaction a user may have for the service at the restaurant, where each number represents the following: 1 - Not good, 2 - Could’ve been better, 3 - OK, 4 - Good, 5 - Great
Useful	The number of “Useful” votes a review gets from other users
Cool	The number of “Cool” votes a review gets from other users
Funny	The number of “Funny” votes a review gets
Review	A detailed description of the user’s experience at the restaurant
State	The business state
City	The business city
Bus_Ave_Star	The averaged stars of a business
User_Review_count	The total number of reviews by this user
User_Useful_count	The total number of “Useful” votes of all the reviews by this user
User_Funny_count	The total number of “Funny” votes of all the reviews by this user
User_Cool_count	The total number of “Cool” votes of all the reviews by this user
Elite	The years a user is selected to be an Elite
User_Fan	The number of followers a user has
Users_Ave_Star	The average of all the ratings a user gives

[**Table 1**] The descriptions of the variables

First, the <User_id>, <Bus_id>, <State>, and <City> variables were eliminated from the data. In addition, the <Useful> and <Review> variables were also changed to binary variables. The <Review> variable was changed to <positive.counts> and <negative.counts>, which indicate the number of positive and negative words in each review.

After making the correlation plot with changing variables, I deleted <User_Funny_count> and <User_Cool_count> because the correlation coefficients between <User_Useful_count> and those are over 0.9.

[1] The Yelp Elite Squad measures how active a Yelp user is. Chances of earning an elite status are limited and are given to users who write over 40 reviews each year (Patterson, 2021).

3 Pre-processing Step and Method

3.1 Eliminating variables that are not related

In this dataset, I am mainly looking at the rating stars and information of the users that can reflect whether their reviews are useful. Therefore, before moving on to the next steps, I would need to eliminate some unrelated variables such as the User_id, Bus_id, State and City. Since the User_id, Bus_id, State and City are in the second, third, ninth and tenth columns.

3.2 Fixing the algorithmic error

In order to quantify the activeness of a user on Yelp, I would need the elite column for processing. However, the variable Elite in the dataset was created based on an algorithm that separates the years with every "20" in them. For example, a user who has been Yelp Elite for 2020 and 2021 was shown in the dataset to have elite years "20,20,2021". Apparently, the given values for the elite years have errors. Since Yelp Elite represents how active and committed a user is as a reviewer on Yelp, I want to use the Yelp elite years as a measurement in the standard of good reviews.

By creating a new variable elite with the Elite years column in R, I am able to make changes on the elite variable only. In order to fix this problem, I applied a for loop to the elite variable to go over each row in the dataset. If a user has never been an elite, I will skip them and proceed to the next. If the user was an elite before, I aimed to search for "20,20" in the row. Since each row of the elite variable comes with all the elite years as a single element, I need to separate the years to multiple elements first in order to search for a specific year. I utilized the unlist() and strsplit() function in R to split the year elements. For example, for the first row, I split "2015, 2016, 2017, 2018, 2019, 20, 20, 2021" to "2015", "2016", "2017", "2018", "2019", "20", "20", "2021". In this way, I have the years separated and can search for the error element.

For each row, I applied an if statement first to convert all the "20" to "2020" in each row. At this point, for users who have become elite in 2020, there should appear two consecutive "2020" in the row. Then, I used a for loop to go over each year in a row to eliminate one of the "2020". After that, what I have is the corrected but still separated rows. For the next step, I put the years together as a single element again using the paste() function and overwrite the original elite column with the corrected elite column. The new dataset is then saved and named as Data_Final_fixed.

3.3 Calculating the average and the mean of the <useful> column

The <useful> indicates whether each rating and review are helpful to the user. I decided to convert the useful column to a column with binary variables that differentiates the "useful" and "not useful" reviews with 0 being "barely useful" and 1 being "highly useful". In order to convert the <useful> column to binary variables, I calculated the average of useful votes of all users; the average number of ”useful” votes for all the users is 2.923 and the median is 1. If I set the threshold to the average 2.923, only 20% of the reviews will be considered as “very useful” and it’s skewed. If I set the binary threshold to the median, which is 1, there will be 22490 reviews that have a useful count greater than or equal to 1, which is 42% of the total reviews. Therefore, I apply 1 as the threshold, and the distribution becomes 42% as “useful” and 58% as “not useful”, which is a relatively more balanced result. Hence, if the useful votes of a review is above 1, I use 1 to represent it (very useful); otherwise, I use 0 to represent it (not useful). I converted the <useful> column to the <is_useful> column with 0’s and 1’s.

3.4 Calculating how many years the user has consistently written reviews

In this step, I want to calculate the number of years each user has been a Yelp elite to measure their activeness in Yelp. I believe that the more active members of Yelp give more useful reviews. Since I am focusing on the elite column solely, I created a new vector “elite” that contains only the elite column from the fixed data.

Now, I want to start counting the numbers by using a new blank vector “Number_of_elite” that has the same row numbers as the data. Then, again I used a for loop that went over all the rows of the elite vector. For each row, I first break down the years into separated elements, and then count the number of elements using the length() function. After counting the numbers, I replaced the “Number_of_elite” in that row with the counted number of years. After going through the for loop, I have the vector “Number_of_elite” representing the number of years each user has been an elite.

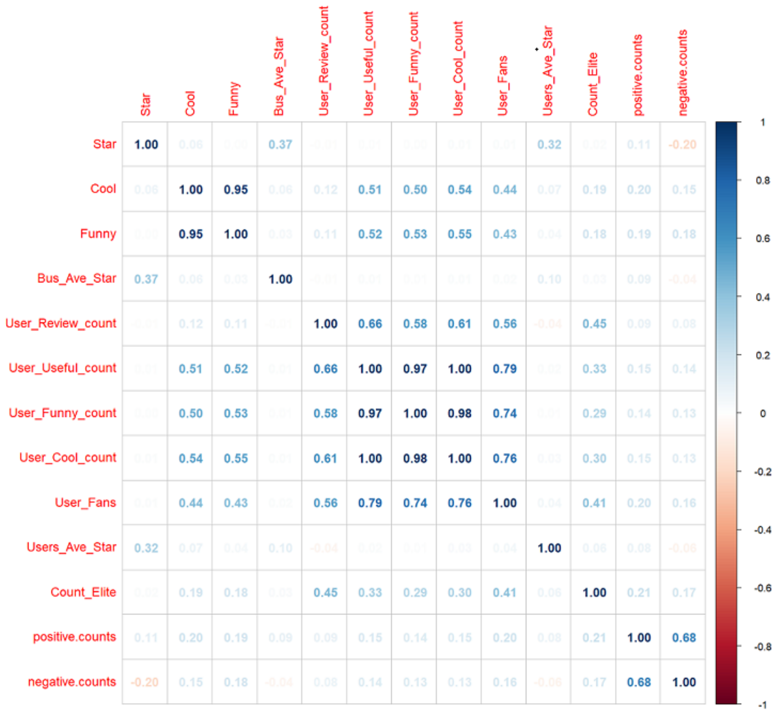
As I am interested in finding the relationship between the number of useful votes and the number of years a user is selected as elite, I want to binarize the “Number_of_elite”. I calculated the mean of the number of elite years of all users and I got 4.67. Therefore, I use 5 as the threshold and I got a distribution of 48% as “elite” and 52% as “not elite”. By using a for loop again, I am able to go over all the rows and generate a new is_elite column.

3.5 Taking review data with NLP

Before I start processing the data, I want to train the review data with NLP (Natural Language Processing) to count the number of positive and negative words for each review. Besides using them as a parameter to evaluate the review usefulness, I want to examine if the amount of positive and negative words in a review is related to the stars given by the reviewers. Therefore, I want to know how many positive words and negative words are used for each review.

To count the number of positive and negative words, it is first necessary to identify which words are used in each review. I found out if each given positive and negative word was used in each review by using “*grepl*()”. Then, I figured out how many positive words were written in the reviews and how many negative words were placed in the reviews by summing the output from *grepl*() I made.

3.6 Eliminating highly correlated variables



[**Figure 1**] The Plot of the Correlation Coefficient between Each Variable

By seeing the correlation plot [Figure 1], I can see some correlated variables. Since correlation coefficient between Funny and Cool is very high (above 90%), Funny is dropped. Also, the correlation coefficient between User_useful_count, User_Funny_count and User_Cool_count are very high (above 90%), so User_Funny_count and User_Cool_count are dropped.

4 Experiment

4.1 Data partitioning

4.1.1 Splitting the data in two (Train data and Test data)

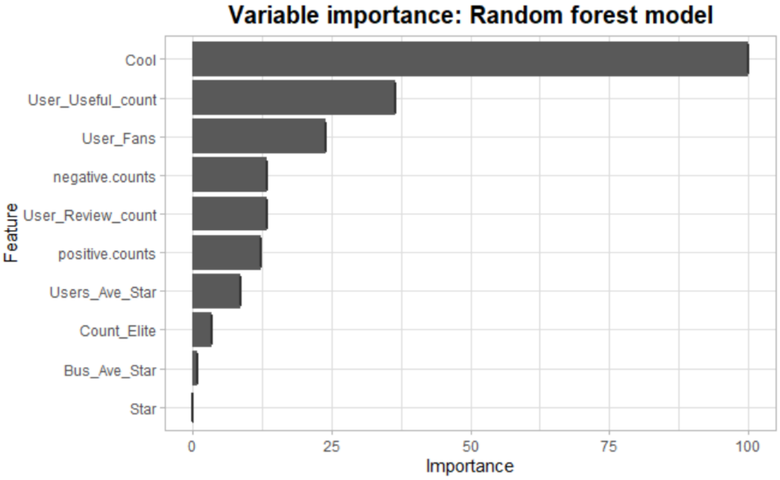
I specified a split of the data into 70%(Train data) and 30%(Test data). The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. Splitting the dataset is essential for an unbiased evaluation of prediction performance.

4.1.2 Using 10-fold cross validation on the machine learning models

With this method I have one data set which I divide randomly into 10 parts. This is because Cross-validation helps to determine a more accurate estimate of model prediction performance.

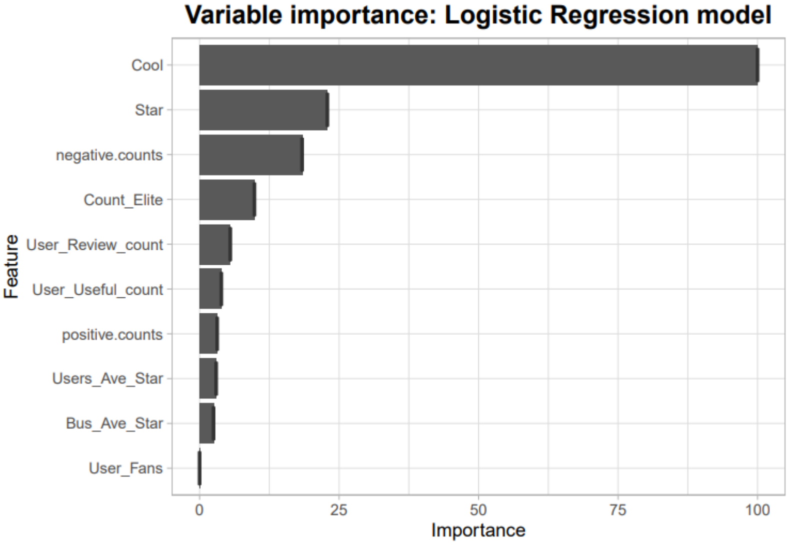
4.2 Use Machine Learning methods

4.2.1 Random Forest



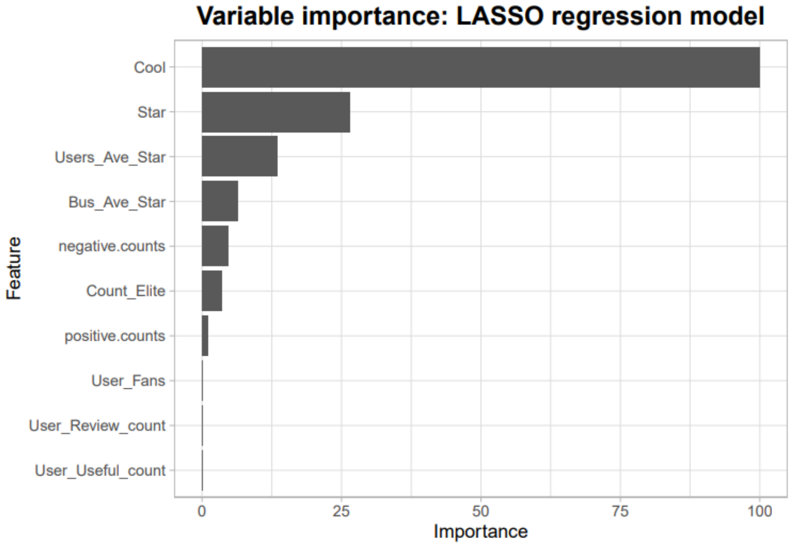
[**Figure 2**] The Random Forest Model Plot

4.2.2 Logistic Regression



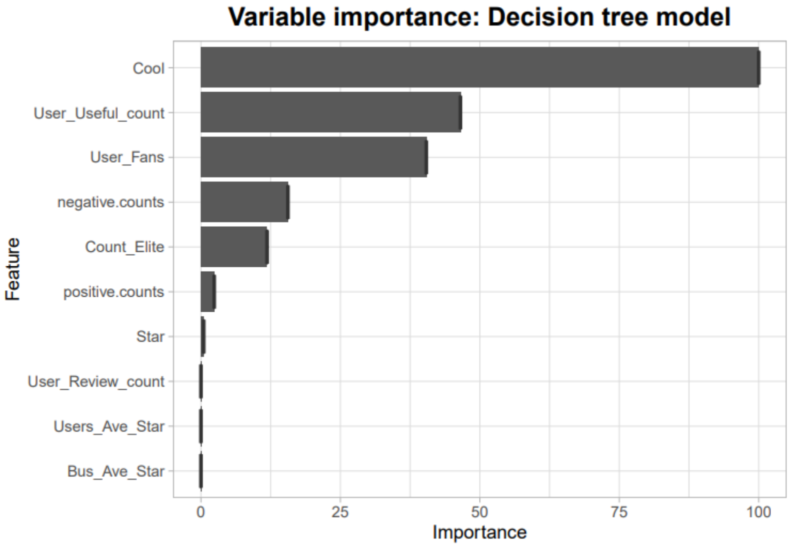
[**Figure 3**] The Logistic Regression Model Plot

4.2.3 LASSO Regression

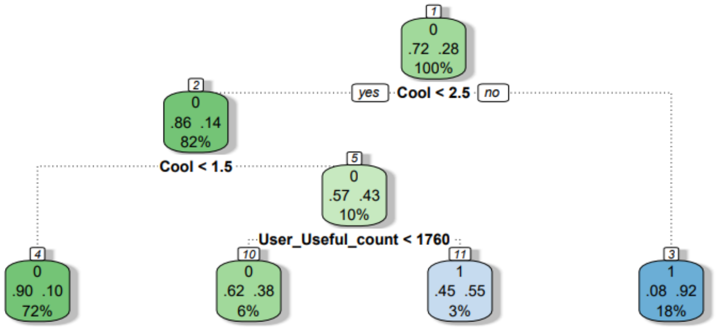


[**Figure 8**] The LASSO Regression Model Plot

4.2.4 Decision Tree

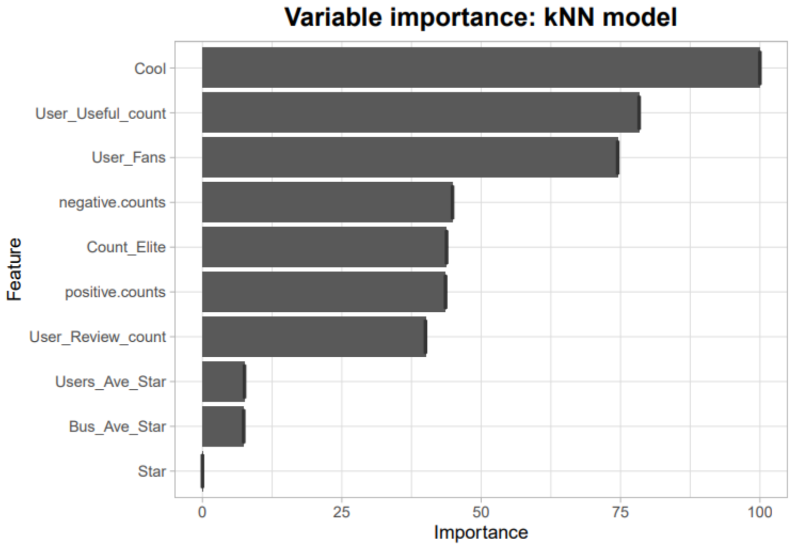


[**Figure 5**] The Decision Tree Model Plot



[**Figure 6**] The Decision Tree Model - Gini Impurity

4.2.5 K-NN



[**Figure 7**] The K-NN Model Plot

There are some similarities among all the models I made; the most important value is <Cool>. Comparing the Random Forest model [Figure 2], the decision tree [Figure 5] and KNN model [Figure7], the values of the importance are the same up to the first fourth. In addition, comparing the logistic regression [Figure 3] and the LASSO regression [Figure 4], the most and the second most important values are the same, but the others are different. Through the plots above, I can conclude that <Cool> is the most important variable.

5 Limitation

5.1 Review Order

When reviewing the restaurants, people would only read a small part of the reviews since there are over 50 thousands reviews for the restaurant. The order of reviews matters a lot in which reviews are shown to users first. As the Yelp official website stated, the order of the reviews are determined largely by posted time (the most recent) and the useful counts of the reviews, which means the more useful ones come first (Yelp Support Center, 2022). Therefore, in this case, some reviews will be viewed by more people and they have a higher chance to be voted as useful. Although users can adjust the orders based on their preference, with such a huge data set, it is impossible to equally treat all reviews.

5.2 Elite Years

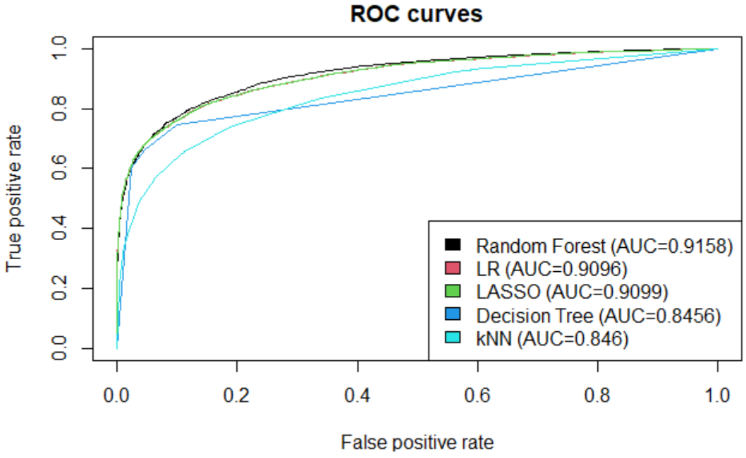
Elite years represents the activeness of a user, rather than the usefulness of the review itself. Elite members are determined by the quantity of the reviews a user posts in a year. Hence, being an elite member only means that the user is active throughout the year. Having chosen to be elite does not necessarily mean that the users produce useful reviews.

5.3 Language Processing

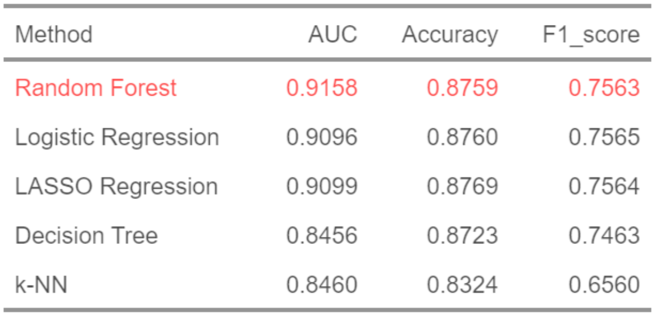
When counting the number of positive and negative words, the numbers of words are counted only once despite whether the reviewer wrote the word more than twice. Moreover, I only counted the number of positive and negative words, but the given words do not represent all the positives and negatives. Although a user wrote a negative review with given more positive words, the data is recognized as a positive review.

6 Conclusion and Summary

6.1 Summary



[**Figure 8**] The ROC Curves of All of the Models



Method	AUC	Accuracy	F1_score
Random Forest	0.9158	0.8759	0.7563
Logistic Regression	0.9096	0.8760	0.7565
LASSO Regression	0.9099	0.8769	0.7564
Decision Tree	0.8456	0.8723	0.7463
k-NN	0.8460	0.8324	0.6560

[**Table 2**] Summary of the Five Prediction Models

By the ROC (receiver operating characteristic curve) curves of all of the models [Figure 8], the random forest has the highest number of AUC (area under the ROC Curve) values among the regressions I made; it is the best model for predicting. Random Forest has an AUC score of 91.58% that shows the model is learning the data well enough.

6.2 Conclusion

By using the Random Forest model that I made, the restaurants can predict whether a review is useful or not even before people start pressing a useful button on Yelp. If a review is useful, which is written as a reasonable good, bad or both things, the restaurants need to change the customers' complaints or maintain the good contents the reviewers have told for their restaurants' development. However, if a review is not useful, which is written as an unreasonable and unfair comment, the restaurants can ignore it or they can also appoint the unreasonable users as black consumers. Accordingly, the restaurants can develop their restaurant by values that support whether a review is useful without wasting the time until people start pressing the useful button.

References

Patterson, B. (2021, May 13). *What businesses need to know about the yelp elite program*. MarTech. Retrieved November 27, 2022, from https://martech.org/businesses-need-know-yelp-elite-program/

Yelp Support Center. (2022). *How is the order of reviews determined*? Yelp. Retrieved November 30, 2022, from https://www.yelp-support.com/article/How-is-the-order-of-reviews-determined?l=en_US#:~:text=Yelp's%20default%20sort%20order%20shows,appear %20before %20a%20newer%20one.