Sentiment Analysis of Yelp Review and Rating Prediction

Anqi Li 1[‡] (UID: 405392430) Yufei Wang 2[†] (UID: 805307029)
Stella Wang 3[‡] (UID: 105426525) Joshua Zhang 4[†] (UID: 805573810)
[‡]Department of Mathematics
[†]Department of Statistics

1 Introduction

A. Background

In today's consumer focused world, user reviews are crucial for the shopping process that allow people to better understand businesses' product and service. In fact, reviews are so important that a whopping 93 % of consumers say reviews influence their purchasing decisions. However, in addition to consumers, businesses also heavily consider their own reviews to learn about themselves, seeing which areas and products need redevelopment and which are successful.

On most sites, such as Amazon and Yelp, user reviews have two main parts: a free-form text section in which users can voice their opinion on the product or service of the company, and a star rating from one to five. While many consumers will sometimes look to the opinions of others, the hundreds sometimes thousands of comments on the business are often too much to read for those that are making quick purchasing decisions. This is especially true for users making food purchases, who will often decide on which restaurants to eat at just by the average star rating of the business. Many consumers will not even engage with business with less than a 3.3 star rating (3).

Since the star rating is so integral to a businesses profit, the problem of predicting the star rating of reviews given the textual comment of users has become a popular, but difficult issue. Discovering the relationship between review and star rating can be complicated as users can give the same review and have differing opinions or have the same opinion and have different star reviews. For example, consider the case of two users giving the same star rating of 2/5. User A can give a 2/5 because they enjoyed the food but hated the service of the restaurant. User B can give a 2/5 because they thought well of the service, but disliked the food of the restaurant. As we can see there can be many facets with textual comments that increases the difficulty of setting a star rating to it.

B. Bag of Words

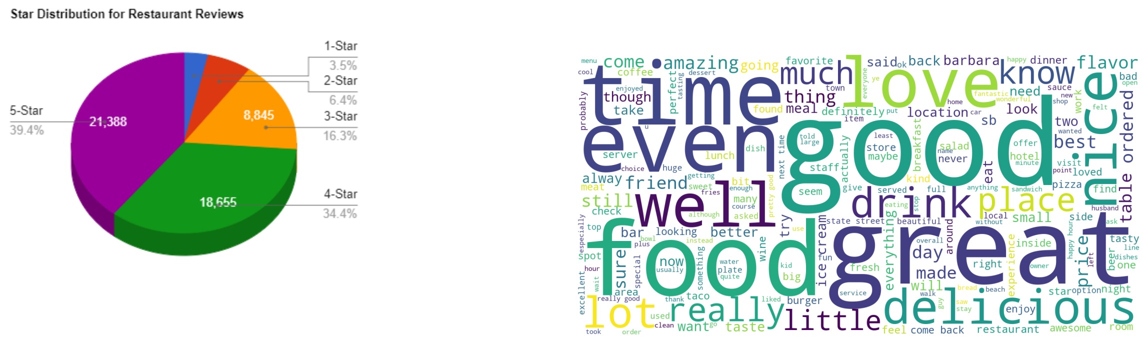
However, while challenging many models have been developed to solve this issue of drawing out the sentiment from a piece of text (also called sentiment analysis). One specific model, first originating all the way back from Zellig Harri's 1954 article Distributional Structure, is called the bag-of-words model. In most applications, the bag-of-words model is used as a tool of feature generation. By transforming textual reviews into quantifiable features, data scientists can then use these numerical values in many machine learning models to formulate relationships with other variables. The most common type of quantitative feature used in bag-of-words model is frequency, which is the number of times a term appears in the text. After defining a dictionary or bag of words that you want to keep track of, you can then create a vector of values that lists the frequency of each of the words in the review. This report will be using the bag-of-words model to transform textual reviews from Yelp into features that can be used to predict/classify the star rating of the review.

C. Goal

Our goal is to analyze the sentiments of user reviews and create a classification model to predict the Star(from 1 to 5) that users give to restaurants. We use 53,845 Yelp reviews on restaurants by users in California to perform an study on text preprocessing techniques. We also compare the accuracy and effectiveness of several machine learning and regression models on predicting the Stars that users give to the restaurants (from 1 to 5).

D. Data Description

As shown in Figure 1(a), the majority of user reviews have ratings of 4 or 5, while 1-3 star ratings only account for around 26 % of the reviews.

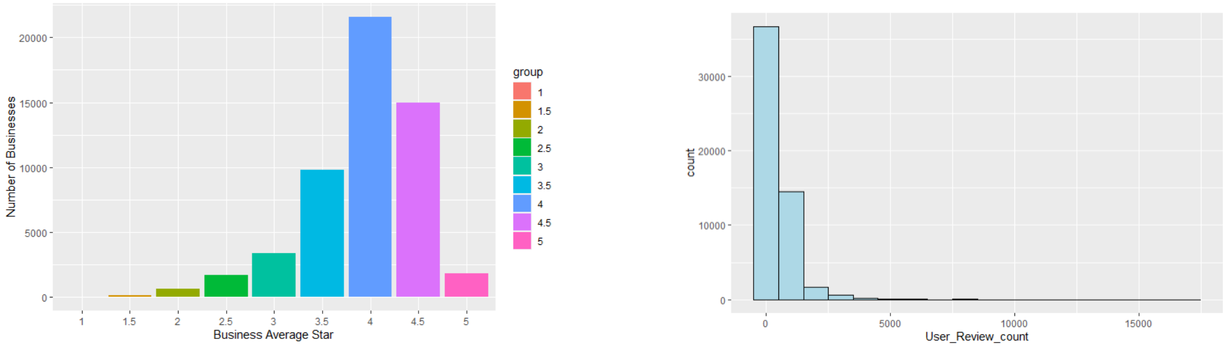


(a) Distribution of Stars

(b) WordCloud of Frequent Word

Figure 1: Dataset Features Part I

As shown in Figure 1(b), we can see that the most frequent sentimental words appear in the review includes good, great, love, well, nice, delicious, etc. And noticeably, most of the frequent sentiment descriptions are positive descriptions.



(a) Distribution of Business Avg Star

(b) Distribution of User Review Count

Figure 2: Dataset Features Part II

As shown in Figure 2(a), the distribution of the average business stars is left-skewed with most businesses having an average star rating of over 3.5.

As shown in Figure 2(b), the distribution of the user review count is strongly right-skewed, with many users in the zero to one thousand review range. There are very large outlying users with over 7500 and 17500 number of reviews. Thus most of the reviews come from experienced Yelp users who comment daily.

2 Pre-processing Step

A. Preparation

The dataset we use is already cleaned and there is no NULL data in the "Review" column. Also, it only contains the reviews of users that published more than 150 reviews. We named the cleaned data 'Data_Final' and change it into a csv file for further investigation. The raw dataset contains 53845 rows and 18 columns. Each row represents the information of one review. The columns includes the index, User_id, Bus_id, Star, Useful, Cool, Funny, Review State, City, Bus_Ave_Star, User_Review_count, User_Useful_count, User_Funny_count, User_Cool_count, Elite, User_Fans, and Users_Ave_Star. In order to apply the variables in various machine learning models, we process the columns of interest and transform the Review State and the Star variables in the proper forms that are ready to use in the following up procedures.

Specifically, we applied the sentiment lexicon method to the Review State column. We obtain two sentiment lexicons online, one containing 2006 positive sentimental words and the other containing 4783 negative sentimental words. We first unified the form of the text content and deleted the content that could cause possible confusion - we removed the special characters, single characters, white space, and prefix with no sentimental meanings from the text; unified the form of the words by converting every word into lowercased form; and applied the lemmatization method to unify the word class of the sentimental words to avoid possible variations caused by the difference of word class and increase the accuracy of the following up models. After that, we compared the existing sentimental word list with every word in the processed Review State column, found the sentimental words in the text, and transformed the column into a list of vector with sentimental words marked with value 1 and the rest words with value 0.

Besides, while there are 5 levels of rating in the current data set, it can be too much information for the models to find the effective pattern of the data and can be not helpful with the prediction power of the models. Also, because of the fact that many machine learning models take in binary input, we classified the Star column and created a new binary column - we marked the rating value from 1 to 3 with binary value 0, and the rating value of 4 and 5 with binary value 1.

B. Variable Creation

One of the last steps of the pre-processing process was transforming some of the original parameters into numerical features and creating new parameters that could be used in classification and regression. In total there were four new variables that were created: a positive score feature that indicated the amount of positive words in the review, a negative score feature that indicated the opposite, a total score feature indicating the difference between the number of positive and negative words in the review, and an elite feature indicating the amount of years the user had been elite status. The first three new parameters were created by splitting up the bag of words into a positive and negative list, while the last elite feature was a variable transformation from the list of years to the number of years.

C. Distribution

Here, include the plots of the outcome variables and the predictor variables. The analysis of the original data(include boxplot of Stars, the distribution of each variables, correlation plot) (not yet finished; plan to apply word cloud plot to show the most frequent sentimental words & a frequence plot with y = count, x = high freq words)

D. Train Test Split

In order to avoid the problem of overfitting and ensure the accuracy of our model. We applied random split method to the dataset with the ratio of 80% training and 20% test data.

3 Experiment

3.1 Binary Models

After we transform and standardize the review data into features consisting of 0's and 1's, we classify sentiments in the review data using several ML algorithms. For ML models, we split the data set into training and testing sets where 80% of them are training data and 20% of them are the validation data which would be used to test the accuracy of the trained model.

A. Machine Learning Models

We first select three popular machine learning classifiers and apply them to the processed Yelp data to achieve text classification. Specifically, we apply Random Forest, K-nearest Neighbor, and Logistic Regression to the dataset. The k-nearest neighbors algorithm, also known as KNN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point (8). The random forest is a classification algorithm consisting of many decisions trees and it gives results competitive with boosting and adaptive bagging, yet do not progressively change the training set (6). The logistic model (or logit model) is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables, which is favorable for describing and testing hypotheses about relationships between a categorical outcome variable and one or more categorical or continuous predictor variables (1). We inputed the standardized reviews as the predictor. To ensure the prediction accuracy of these three classification models, we transformed the rating Stars into binary form and input as the response variable. Besides, to optimize the performance of the models, we applied cross validation to each model. Specifically, we used Grid search in K-fold cross validation to achieve hyperparameter tuning for the models (4). After applying the grid search to each model, we found that the optimal "criterion" parameter for Random forest model is "entropy"; the optimal number of neighbor groups for K-nearest Neighbor model is 4; and the optimal parameter "C" for Logistic Regression model is 0.01. Employed Random Forest, KNN Classifier, and Logistic Regression models and conducted cross validation to our data, we achieved an accuracy of 80%, 73%, and 82%, respectively.

Table 1: Comparison of results in different models

ML	Precision	Recall	F-Score	accuracy
Random Forest	0.84	0.91	0.87	0.80
K-nearest Neighbor	0.82	0.82	0.82	0.73
Logistic Regression	0.82	0.96	0.89	0.82

These three models takes in similar type of inputs and are likely to obtain similar result.

But they use different methods to classify the data - Random Forest uses decision trees to classify new input object from the input vector; K-nearest neighbor classifies input object into assigned number of groups to group the relatively similar objects; and the Logistic regression measures the statistical significance of the input elements and makes prediction based on that. Thus, we apply all three models to our data. And as shown in the table below, we can see that based on the performance parameters, the Logistic Regression performs the best among the three model.

B. Classification Tree

A decision tree is a graphical representation of all possible solutions to a decision based on certain conditions. On each step or node of a decision tree, used for classification, we try to form a condition on the features to separate all the labels or classes contained in the dataset to the fullest purity (7). By using the rpart library, we were able to create a classification tree showcasing three different splits. Two of the splits resulted from the same feature (totalscore), and the other split was from the business average star. The results shown by the tree are very logical, as if a review has two more positive words than negative words, it is likely that the review is a positive star. Also for businesses that already have an average star of over or under 3.8, it makes sense why reviews for the business would also be beneficial or degrading. However, though the tree does showcase logical insights, it does specify the specific numbers that create splits in classification.

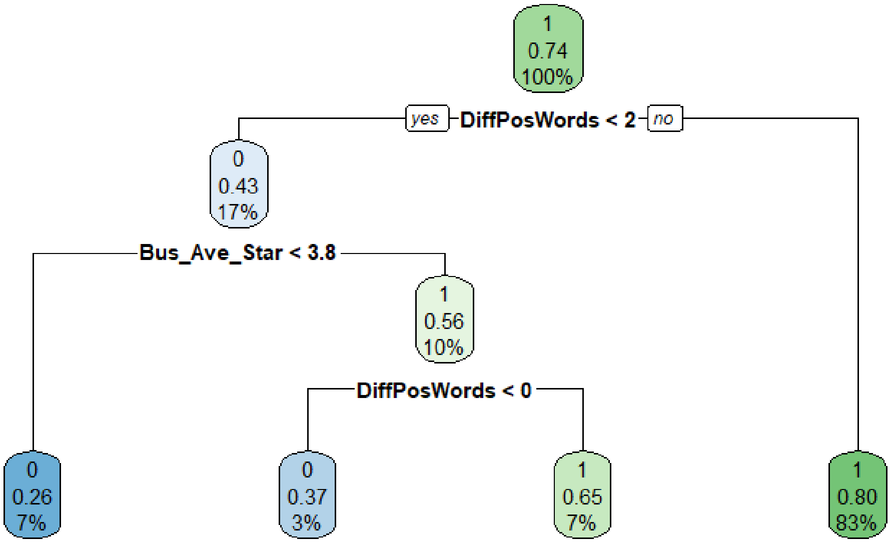


Figure 3: Classification Tree

While being able to use the textual reviews to classify whether or not the star review would be good or bad, another important issue in the dataset is finding out which variables besides just the textual reviews, have the most influence on the star rating. While there are many different ways of conducting variable importance, one visual method of finding important parameters is the classification tree. Classification trees are constructed by repeated splits of subsets of x into two descendant subset. Each split is determined by a specific criterion, and the higher the split, the more important it is.

3.2 Non-binary Models

A. Neural Network

Besides the models we learn in class, we also construct a Keras Functional model that uses neural networks to train our data set. A neural network enables multi-layers of analysis, more flexibility and customizes methods based on the data specifically, not necessarily needs to take in binary inputs. (2)

All of the models in the previous sections convert “Stars” into a binary variable. Although the highest prediction accuracy reaches 82 %, we lose a lot of information when we reduce the level of reviews data from five to two. Thus, we decide to employ the Keras functional model using Python’s TensorFlow library to train a non-binary deep learning model. For the data pre-processing step, in addition to the numericalization and standardization steps mention in Section Two, we implement the categorical embedding for the “Stars” column. Then, we perform the training and testing data split. For each of the two splits of the reviews data, we batch them into small chunks of data, which helps with reducing the training run time. Then, to construct the model, input layers are applied to all reviews data, which are truncated to a specific length to deal with different lengths for short and long reviews. Then, we work on embedding layers. Also, Conv1D and GlobalMaxPooling1D layers were applied to each layer.A dropout layer is implemented before the final dense player is applied to control over-fitting. During the model builing process, we experiment with the value of hyperparameters such as filter in the Conv1D layer and rate in the dropout player, and we set filter to be 64 and rate to be 0.2. Finally, the model achieves an optimal testing accuracy of 55%.

B. Regression Model

In this method, we run a regression of the “Stars” on “Review” as well as other independent variables. To change “Review” into a numerical variable, we transform the contents of the reviews into matrices. We use the two dictionaries, one consisting of positive sentimental vocabularies and another consisting of negative vocabularies, to extract the sentimental words from the reviews and represent the time that a word appears in the review as a cell in the matrix. We have 53,845 observations in this dataset, 4783 vocabularies in the negative sentimental words dictionary, and 2006 vocabularies in the negative sentimental words dictionary. We create two matrices for each dictionary. The matrix of the negative dictionary has the dimension of 53845 rows and 4783 columns, where each cell represents the time that a certain negative sentimental word appears in a review. Similarly, the matrix of the positive dictionary has the dimension of 53845 rows and 2006 columns. In order to reduce the computational length, we eliminate the columns with all zeros and generate two new matrices. Then, we sum the number of all positive sentimental words and the number of all negative sentimental words that appear in each review. Next, we compute the proportion of the positive sentimental words in all sentimental words for each review and save them to the original dataset.

Now we can start to run the regression model. We split our dataset into 80 percent of training data and 20 percent of testing data. To see the correlation between the numerical variables, we use a correlation plot to visualize the correlation, shown in Figure 4. For the categorical variables "User_id", "Bus_id", "State", "City", and "Elite", we don't include them in our model because they are not helpful to our predictions. The reviews are all from California, so "State" does not matter. From the regression of outcome variables "Star" on "City" and "Elite", the variable "City" and "Elite" both do not have a big influence on Star, so we will not include them in our predictors as well.

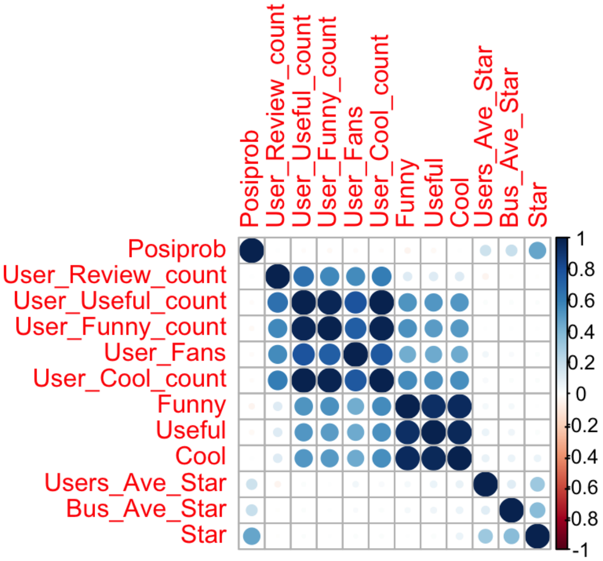


Figure 4: Correlation plot of the numerical variables

Next, we construct a full model that consists of all numerical variables and see if the variance inflation factors (VIF) are above 5. If there exists predictors with VIF larger than 5, we remove the predictor with the largest VIF from the model, run the new regression, and acquire the new VIFs. Repeating the steps above, we finally do not have severe multicollinearity among the independent variables in a multiple regression model. Then we apply the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) to the current model. The result is our final model, which uses "Bus_Ave_Star", "Users_Ave_Star", and "Posiprob" as the predictors to predict the "Star." Table 2 below shows the R output of the model.

Table 2: R output of the final regression model

Coefficients:	**Estimate**	**Std. Error**	**t value**	**P value**
(Intercept)	-2.112210	0.049914	-42.32	close to 0
Bus_Ave_Star	0.516557	0.007315	70.62	close to 0
Users_Ave_Star	0.639330	0.011239	56.88	close to 0
Posiprob	1.989216	0.021998	90.43	close to 0

Lastly, we fit the model to our testing dataset and predict the "Stars" for each observation in the testing dataset. The results include decimals, so we round it to the nearest integer. Note that some results are larger than 5 or less than 1. We categorize all results larger than 5 to be 5 and all results less than 1 to be 1. Using the 20% of data as testing data, finally, among 10769 data, the number of correct predictions is 4839, so the accuracy of the predictions is 0.4493453.

We also use k-fold cross validation to evaluate our regression model. Let k = 10, the mean of the ten mean squared error (MSE) we get is 0.7338315. The mean of the ten root-mean-square deviation (RMSD) is 0.8566397. For the calculation of accuracy, we round the predicted results to the nearest integer and still categorize all results larger than 5 to be 5 and all results less than 1 to be 1. The mean of the ten accuracy we get is 0.4526323.

4 Discussion and Conclusion

We currently build machine learning models to predict the future rating based on the text variable and build a regression model to take other significant variables into consideration. To further increase the predictive power of our models in future studies, we would like to build a model that takes both text information and the rest of significant factors into consideration.

Besides, there is a language limitation to our current prediction models. Currently, we are only able to analyze reviews written in English and conduct text mining on them. However, there are reviews written in other languages like Japanese. We did not take those reviews in other languages into consideration. We can make possible improvements in the future study.

In our study, we apply different approaches to utilize the Review text to make prediction for future Star rating - multiple binary classification models: Random Forest, K-nearest Neighbor, Logistic Regression, and Classification Tree, and a non-binary neural network model using the Tensorflow package. Besides, to take other significant factors into consideration, we build a regression model to find significant factors. For ML models taking in binary input, we achieve a highest accuracy score of 82%. For the neural network model, we achieve an accuracy of 55%. For the regression model, we reached a R^2 value of 45%.

References

[1] An introduction to logistic regression analysis and reporting. (n.d.). Retrieved December 10, 2022, from https://datajobs.com/data-science-repo/Logistic-Regression-%5bPeng-et-al%5d.pdf.

[2] Jason Brownlee (2017, October 27). How to Use the Keras Functional API for Deep Learning from https://machinelearningmastery.com/keras-functional-api-deep-learning/.

[3] Jill Anderson (2018, November 27). Why are customer reviews so important? Medium. Retrieved December 2, 2022, from https://medium.com/revain/why-are-customer-reviews-so-important-185b915d4e5d.

[4] Malato, Gianluca. "Hyperparameter Tuning. Grid Search and Random Search." Your Data Teacher, 27 May 2022, https://www.yourdatateacher.com/2021/05/19/hyperparameter-tuning-grid-search-and-random-search/.

[5] Podium 2017 State of Online Reviews. https://learn.podium.com/rs/841-BRM-380/images/Podium-2017-State-of-Online-Reviews.pdf.

[6] Random Forests – Random Features. https://www.stat.berkeley.edu/ breiman/random-forests.pdf.

[7] Roy, Abhijit. "A Dive into Decision Trees." Medium, Towards Data Science, 6 Nov. 2020, https://towardsdatascience.com/a-dive-into-decision-trees-a128923c9298.

[8] "What Is the K-Nearest Neighbors Algorithm?" IBM, https://www.ibm.com/topics/knn.

[9] Zellig S. Harris (1954) Distributional Structure, WORD, 10:2-3, 146-162, DOI: 10.1080/00437956.1954.11659520 from https://www.tandfonline.com/doi/abs/10.1080/00437956.1954.11659520.