# STAT 154 Final Project Report

Yakub Akhmerov, Sho Kawano, Lydia Maher

May 5, 2017

## 1 Introduction

This report illustrates the analysis of Yelp reviews and using them to predict the star rating of businesses. The idea behind this is to use one data set and learning from it to create a model. This model will be used to predict values, the equation for one of these business is a linear model which conventionally looks like the following, for p variables and n businesses:

$$\hat{y}_i = \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_p x_p, \forall i = 1, ..., n \tag{1}$$

For a bit of a description of what is going on here, the left side of the equation is the output, dependent variable. It is the response of a collection of attributes known as variables. In our case, this value is the predicted star rating of a business. These variables which help predict this star rating are all held together by the addition operator, which is shown on the right side of the equation is the addition of several unknown variables, precisely $n$ of them. In our context, every $x$ represents a variable from our data, examples of this in our context could be average amount of time a business is open a week, whether they accept credit cards, how many times a person says "disgusting" in their review, etc. These variables could be ones that we were given in our uncleaned data-set, or they could be ones that were created through analysis. Many of the variables were created through splitting of existing variables, the specifics of which will be discussed later. The reason for the hat on $y$ and on the coefficients in front of the $x$'s, the betas, is because this is a fitted model. The coefficients, betas, are estimated through calculus to create multiples of the variable values and help predict the response variable. The premise of which is obviously having the data split into training and testing and use the training data to predict what the output variable will be for the testing data. The training data was given to us in a few different tables. The most efficient option for us was to combine them all into one matrix to get an X matrix. Pre-processing was motivated since to be good and ready for analysis, the values in the data frame should be quantitative values.

## 2 Pre-processing

The initial step of the analysis was the pre-processing. We started by examining the testing data sets and the values of their variables. In data analysis, it is crucial to have values in a quantitative format to run analysis between the variables representation. Many of the variables were formatted in text and they had to be split up to get work done on them. Regarding the training set of the reviews, where each row was a review by a different user, with a variable called business_id specifying which business had the review written for it. There were a few variables which needed some work done to them. The text variable had all the reviews in plain text which motivated some work to be done to them. For starters, many had stop words such as it, the, but, etc. which don't provide much meaningful information and we looked to take them out. The next stop was to parse through the vocabulary of all the words and see which ones occurred the most. Originally, we sought to use 100 common phrases for the dividing up of words. Using python code, we could find the most common words in the texts. Those 100 words were then changed to variables and the values each row took was how many times the review had that word, essentially creating a NLP takes a set of text and creates them as variables where each row takes values of how often that value appears in the review. This procedure, known as one-hot-encoding, helped not only the text review data but also the business train data, which had a few variables of this nature, hours, attributes, and categories. Each entry for those values was a list. The objective was to enlist all the lists of all these entries and create them into variables to analyze them. The business reviews dataset had for starters, the hours variables were difficult because every company had their unique set of hours that were unlike any of the other ones so one-hot-encoding would cause extremely sparse data since the likelihood of different business having the exact same hours is rather low. This motivated us to split the data into time frames, with the times being those which were most popular, i.e. occurred the most through text vectorization. The attributes Upon completion of dividing up the values in each row, it was time to combine the datasets. Our goal was to design an X matrix which had all our features in it. Our original matrix had 549 features in it. Fitting a model proved to be difficult to do so brain storming lead us to remove more variables. The criteria to doing this was with threshold of how often strings (variables) would appear in whichever context we were using it in. For the review datasets, the threshold was whether the word appeared in at least 11646 cases (at least 10%). For the attributes variable in the business training dataset, the cutoff was whether it appeared in at least 250 cases (again 10%), and for the dates, whether the hours open was in 100 business ( 5%). This shrunk the features rather well and brought us down to around 300 variables. Which was still a rather large number of features. Though the actual fitting of the model will be discussed later, the specific amount of variables pertains to it so it will be mentioned briefly. When fitting the model, problems were encountered with the predictors, there were 21,919 cases to predict. However, when fitting the lasso and ridge model, there was a loss of 233 variables to bring up down 21,686

predictions, which was strange considering the OSL predictors had preserved the amount of cases. Naturally, what made the most sense was that there were some values with NA in them and the fitting was tossing it out. A quick eye test of the R function complete cases showed there were some cases where the rows values were not complete and had missing values. Naturally the most effective way to preserve the true values of the cases was to remove the variables which had such values. Consideration was put into filling in the NAs with values such as 0 or its predeceasing value, but this would be detrimental to the predictions since it relies on these values. Not so coincidently, the summary of OLS, had these variables as having a low effect on the output variable. Which was done fitting a linear model and examining the summary of it and examining the table of variables and examining the p-values of them. The p-value is the probability of getting the observed data given the null hypothesis. So, a low p value means the variable has a high level of affect on the output and vice versa. So having variables very high p values implies that variables are having little to no affect on the output. The process by which we filtered out the missing values started with using supply on the columns of the testing dataset (testing has the NAs) to create a dataset with columns that had missing values. The colnames of this dataset was set in a character string. The columns were then extracted from the original testing dataset and removed to bring the variables count to 228 for testing and 229 for the training since the training obviously includes the response variable to be trained on. Finally, the tedious yet highly necessary process of preprocessing the data was finished and we had our dataset ready for analysis.

# 3 EDA

The next step was the crucial step of exploratory data analysis. A good introduction to EDA is to perform the scree plot, as its a simple analysis of the data. The scree plot summarizes the proportion of variance among the principal components, if it arises that the proportion of variance is high among the principal components, it can be useful to removed a large proportion of the data to reduce it and fit a better a model. Intuitively if one can get a few components to explain the a large portion of the data, it can be useful. Nonetheless it turns out 41% of the data can be explained through the first 3 components, while 73% can be said of the first 10 components of the data. Given that we have over close to 230 variables, it is pretty efficient that 4% of the features can explain around 2/3rds of the data. The star ratings have a median of 3.5, an upper quartile of 4 and a lower quartile of 3. Between 3 and 4,the distribution is pretty even. Generally, they are skewed slightly towards the upper values of y, with thewhiskers not extending below 1.5 stars. Reviews of 1 star are seen as outliers.

# 4 Model Fitting

The following step was the model fitting. Through the help of EDA, we decided to use ordinary least squares regression, lasso regression, ridge regression, neural nets, random forests, and PCA regression. Given the magnitude of this data, it was thought necessary to run the fitting on other servers which proved to be efficient. For OLS regression, we fit a model with the lm() function in R and saved the output as a variable and predicted it against the testing data and saved those as variables and to get the predictions. It was rather simple saving it due to the simplistic nature of the functions calls in R. Ridge regression was then done and we took a different approach for Ridge regression since it introduces a penalty, the L1 penalty. With the L1 norm being i— and the L2 norm being i2. Both penalties are different than ordinary least squares as they provide a penalty on on the esitmation of the coefficients. OSL estimators are BLUE (Best Linear Unbiased Estimators), meaning if you took estimate of the coefficients in the regression model for some large value N times, the average of those estimates will be very close to the true value of the coefficients. On the other hand, ridge and lasso drop the unbiased critera and focus on lowering the variance as much as possible, this comes with an added side effect, which is the bias-variance trade-off. It is important to find the optimal point between the bias and variance, because having a large amount of variance means as you take N number of estimates and average them out, they will deviat far from the the true value of the coefficients, while having a large amount of bias means the expected values for the estimators are very different. Essentially, we wanted to train the models by cross validation to find the optimal penalty by examining the best_tune coefficient The value was then saved and using the output coefficients to find the best alpha and lambda parameters. The model was then saved and used to predict the output for the testing data. The next step was a similar technique, lasso. The technique was obviously pretty much identical and we used the optimal model to predict on the testing data. Following our 3 different regression techniques was a different type of model fitting procedure, random forests. The random forest is an ensemble approach that can also be thought of as a form of nearest neighbor predictor. Ensembles are a divide-and-conquer approach used to improve performance. The main principle behind ensemble methods is that a group of weak learners can come together to form a strong learner.

# 5 Conclusion

In summary, the report showed some interesting findings. The analysis was certainly a good looking glass into industry. It taught about the importance of using data and working with it. Essentially