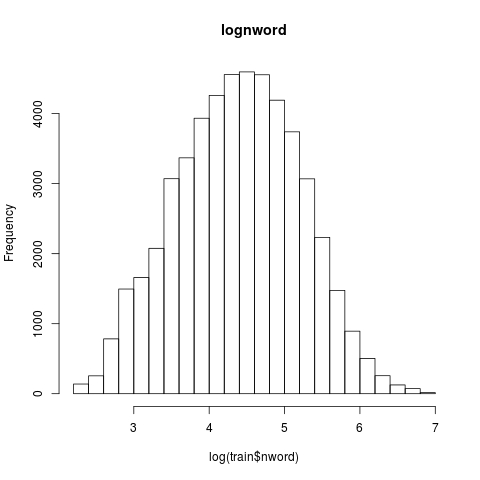
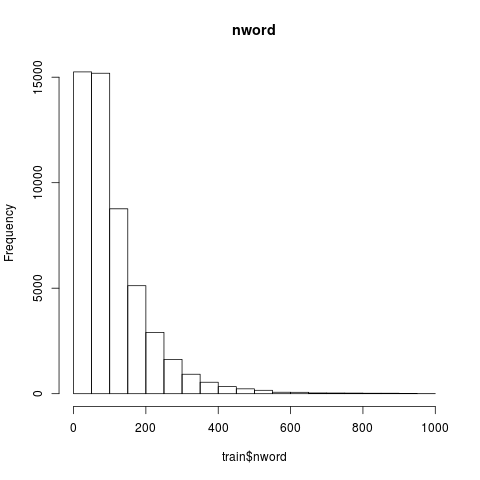
**Introduction** (*Crystal Liu)*

Nowadays, reviews play important roles in helping people to make better decisions. People are used to search online and read the reviews before they decide where to eat out. Yelp is a local-sourced forum that provides search services powered by crowd focusing on “help people find great local businesses”. The star rating system in Yelp, which is 1- star represents “Eek! Methinks not”, and 5-star means “Woohoo! As good as it gets!”, offers a succinct guideline for users to rate restaurants. The text contains detailed information why he or she gave that rating. While it is easy for humans to understand the “sentiment” behind the reviews (whether the review is positive or negative), computers are not sophisticated enough to do that without any background. For example, a machine would find it difficult to understand grammatical nuances, slangs, cultural variations, different spellings, and others. Therefore, our goal is to develop a ‘model’ which could allow machine to grasp the underlying emotions and predict the star rating based only on the review texts (since star rating could be a good measurement of the sentiment). Our motivations are: 1. To figure out what are the factors make a review positive or negative. 2. Build a multiple linear regression model with those factors and predict the star-rating with the review text. The data we used includes 85,543 comments about local restaurants in Wisconsin from Yelp, and the included comments are businesses with at least 3 comments and also older than 14 days.

**Data Preparation**

In this project, we have 2 datasets: one is a train dataset for constructing a model and the other is a test dataset for testing. The train data has 51326 observations with 508 variables such as ID, star, name, city, postal code, text and some specific words extracted from reviews.

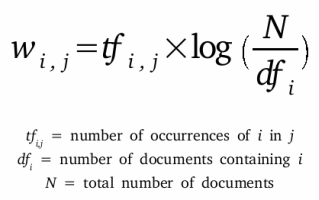
The test dataset has 34217 observations without star rating. Investigating the dataset, we noticed that there are errors in city names. We corrected the cities name: “De Forest”, “Deforest” and “DeForest” into “De Forest”, “Mc Farland”, “ Mcfarland” and “McFarland” to “McFarland”, and “Sun Praiie” and “Sun Prairie” to “Sun Prairie”. We also noticed that the number of words is right-skewed, hence we took log of the number of words to make it normally distributed.

**Data Transformation and Predictor Choosing** (*Yupei Lin)*

In our project, we utilized R package called “Quanteda” to manage textual data. Quanteda allowed us to tokenize the raw text into a document-feature matrix(DFM), which is a structure provided by Quanteda for further analysis. We first removed all the numbers, all the punctuations, and all the symbols including hyphens. Then we converted the texts into lowercase and removed stop words. Finally, we extracted the words into 2-word phrase and constructed a DFM. We extracted phrase because the same word could have different meaning. For example, a word “restaurant” could be neither negative or positive by itself, but it could be negative or positive with adjective (eg. good restaurant).

Second, we selected 4000 words and phrases with the highest frequencies. The reason we took 4000 is because we wanted to get a prediction as precise as possible.

We introduced a method called “Term Frequency-Inverse Data Frequency” (TF-IDF) to weigh the words differently, TF-IDF is generally used as a weighting factor for text mining. Although some words could occur frequently across the reviews, they could be not significant. For example, “food” and “madison” appear a lot but less significant than “fantastic” when predicting the star rating. The main function of it is to reflect how important each phase/word to the reviews in the total collection. First, we calculate the term frequency of word in each review by dividing total number of words in review from times of the word appears in one specific review. Then, we compute the inverse data frequency by diving number of reviews containing the word from total number of reviews and taking the logarithm of the quotient. So, the words that occur rarely in the corpus have a high IDF score. Finally, we combine TF and IDF together to get our data frame. We wrote three functions to calculate TF, IDF and TF\_IDF in our code for this data transformation.



**The MLR Model (***Nan Yang)*

After the word extraction and TF-IDF transformation, we finally obtained overall 4002 predictors: 4000 words we chose based on word frequency, “lognword” and “city” from the train data. We decided to build a multiple linear regression model with all 4002 predictors to predict the star rating for the reviews.

|  |  |  |  |
| --- | --- | --- | --- |
| Word | Coefficient | T-test value | P-value(>|t|) |
| embarrass | -10.400869 | -5.331 | 9.81e-08 |
| never\_come | -8.096793 | -5.417 | 6.08e-08 |
| sick | -12.057062 | -7.652 | 2.01e-14 |

A table above shows a few examples of our predictors, which we think significant. Even though it is not a perfect way to interpret it, they have huge coefficient which could mean they have huge influences on the prediction. The coefficient could be interpreted in a way that as TF-IDF score of a word “embarrass” increases by 1, the star rating decreases by about 10.4. It is complex because the rating cannot be lower than 1 and how TF-IDF increases by 1 is not clear. Low p-values of t-test of those predictors demonstrates that we are 95% confident that these predictors are meaningful in predicting the star rating, considering other predictors as well. Our model has an adjusted R² value of 0.632, which means 63.2 percent of variance could be explained with our model. Our model seems to provide pretty precise predictions. For example, looking at two reviews (ID-7795 and ID-37940), our model predicted them to have 4.56 and 4.84 star ratings. With human cognition, those review seems to be 5-star reviews or at least 4., which is very close to our predictions.

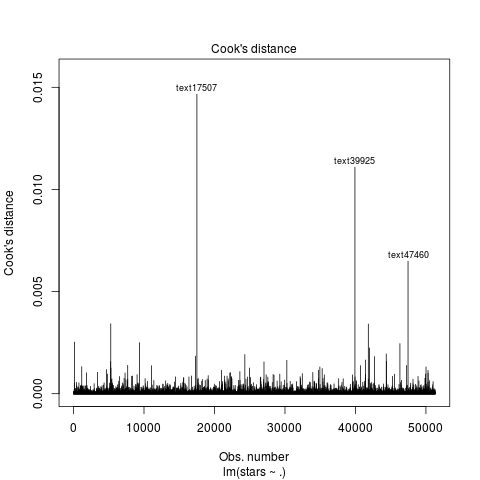
It is complicated to suggest a rule of thumb, since we use TF-IDF transformation. For example, using a word, “sick”, once more in a review does not lead to increase in TF-IDF score by 1. Moreover, it would be misleading if the review is “sick”, then the prediction would result in less than 1, which is not robust.

**Hypothesis test (F-test):**

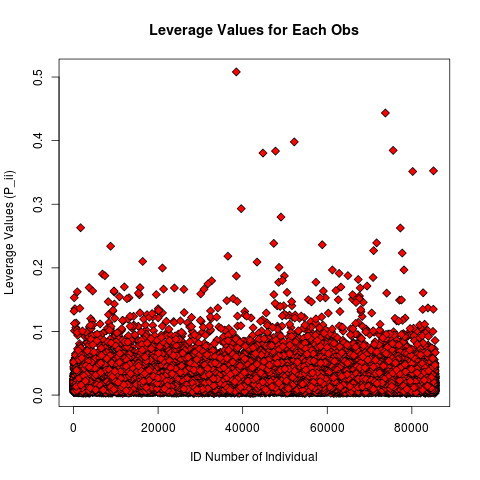
We conduct a F-test to test our model.

H0:  all the β\_i = 0 Ha: at least one of the β\_i is not 0

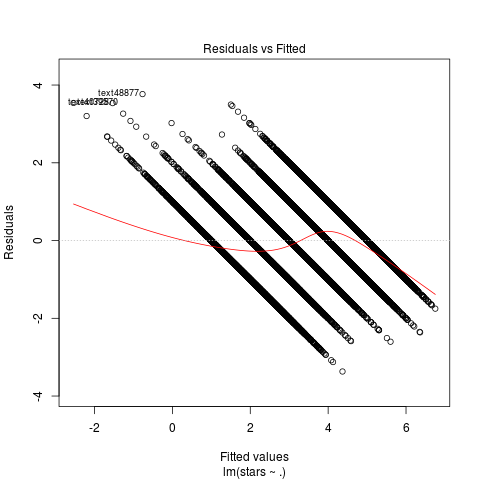
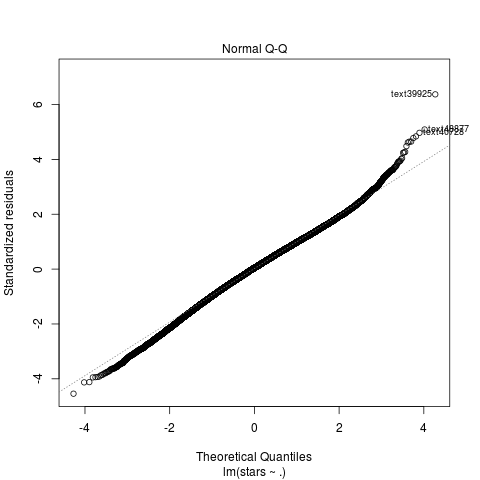
According to p-value of F-test, 2.2e-16 with degrees of freedom of 4022 and 47303, we are 95% confident that some of predictors are useful in predicting the star rating.

**Strengths and weaknesses of the model** (*Sukyoung Cho)*

Strengths:

* Additivity: It is reasonable to infer that an increase in the count of the word would result in the star rating. For example, a review with two “good” would result in higher star rating than a review with one “good”, and vice versa for negative words.
* Constant effects: it seems to be reasonable in general but may have some exceptions such as redundant use of the same word. For example, an increase in star rating from a review with one “good” to a review with two “good” would not be same with the increase of star rating from a review with 49 “good” to a review with 50 “good”
* Context: Sometimes, the same word could have a different influence when it is combined with other adjective words. Our model may not consider the interacting effects between the words. For example, a word, “restaurant”, would not have neither positive nor negative impacts, but it could be positive if it is matched with positive adjective such as “fantastic restaurant”. Our model took phrases into consideration as well.
* Outliers and Leverage: Based on Cook’s distance and Leverage point plots, our model does not seem to have any outlier in ‘x’.

Weaknesses:

* Linearity: We expected that it would violate linearity as we have a discrete variable on ‘y’ - star rating are only integers between 1 and 5. Based on the standardized residual plot on the right, points are not randomly dispersed - there is a clear pattern of 5 parallel lines. It clearly shows the violation.
* Normality: Looking at QQ plot, points are not well-aligned on the diagonal line, showing a curve pattern on the right end. Therefore, error terms are not normally distributed.
* Homoscedasticity(constant variance):based on the residual plot, the variance of the error terms does not seem to be constant as there are 5 parallel straight lines.
* Accuracy: Since our model only accounts for 63.2% (adjusted R²) of variation in star ratings, our predictions will not be exactly accurate.
* Predictor selection: Since TF-IDF select variables based on the frequency of the word occurs, the model might ignore the words that are significant but appearing less in texts when it selects the variables. For example, even if a word, “gorgeous”, has significant influence in predicting star rating, it could be not considered if it does not frequently appear in the comments.
* Inflexibility: As our model is based on only the reviews which are from Wisconsin and written in English, it would only be able to predict the star ratings of reviews from businesses in Wisconsin. It would be inaccurate for reviews from the different state or countries, because each region has different cultural background, nuances, expressions, and language. If we want to predict star ratings of another data set, we would reconstruct the model with new sets of words.
* Simplicity: The model has 4002 predictors: 4002 words and city names. There is no “rule-of-thumb”, which is succinct for people to use. It would be difficult for human to calculate and predict the star rating with our model.
* Interpretability: The interpretation of the model coefficients is complicated, as we are using TF-IDF on the variables. For example, a word, “kind”, has a coefficient about 1, which means that if TF-IDF of “kind” increase by 1, the star rating will increase by 1. The coefficients and TF-IDF values are not intuitive.
* Over-fitting: There are numerous predictors, which would lead to overfitting.
* Robustness: our model could result in a prediction less than 1 or greater than 5.
* Time-consumption: it takes much time to build our model

**Use of the Model**

It could be a prototype of the model for sentiment analysis, as the star rating could be a good scheme to measure the sentiment. For example, when a company wants to monitor customers’ opinions about its new product, it cannot read every review people posts online. Then, the company could build a model based on ours, to observe customers’ feedbacks real time and respond quickly.

**Conclusion**

Our model has many limitations. First, the model is not simple - a multiple linear regression model with thousands of variables. Therefore, it would not be appropriate for humans to predict the star rating. Adjusted R² value demonstrates that the variability in star ratings is not sufficiently explained even with four thousand variables. It could also over-fit due to numerous predictors, and there are other weaknesses as described above. However, it results in pretty small mean squared prediction error of 0.80549 according to the test dataset, which means the model is quite precise. Moreover, it follows the assumptions behind the linear model except what we expected to be violated such as linearity or normality. There is a chance that our model result in a star rating less than 1, which is not robust. However, we can correct predictions by converging all predictions lower than 1 into 1 and ones greater than 5 into 5.

**References**

Data.library.virginia.edu. (2019). *A Beginner’s Guide to Text Analysis with quanteda | University*

*of Virginia Library Research Data Services + Sciences*. [online] Available

at:https://data.library.virginia.edu/a-beginners-guide-to-text-analysis-with-quanteda/

[Accessed 21 Apr. 2019].

Quanteda.io.(2019).*QuickStartGuide*.[online]Availableat:<https://quanteda.io/articles/quickstart.ht>

ml [Accessed 21 Apr. 2019].

Rdrr.io. (2019). *quanteda-package: An R package for the quantitative analysis of textual data in*

*quanteda: Quantitative Analysis of Textual Data*. [online] Available at:

https://rdrr.io/cran/quanteda/man/quanteda-package.html [Accessed 21 Apr. 2019].

Rpubs.com. (2019). *RPubs - Sentiment Analysis for Restaurant Reviews*. [online] Available at:

https://rpubs.com/yceron/155272 [Accessed 21 Apr. 2019].