Prediction of Rating on Yelp Dataset

Team 23: Jialuo Li and Bob Dai

Introduction

Yelp is an Internet company and a crowd-sourced restaurant online evaluation platform to "help people find great local businesses". Yelp has been devoted to providing users with exact and latest information about nearby restaurants. It offers various recommendation of restaurant with just a tap on the screen. Yelp provides people with a platform to rate and write reviews of restaurants.

Our project will focus on (i) finding out what makes a review positive or negative and (ii) predicting a review's rating based on its text and a small set of relevant attributes. The raw data provided by Yelp contains 6.7 million reviews of 193K businesses from more than one thousand cities. To make the data more manageable, we have narrowed the original Yelp challenge dataset to restaurants of the Madison area. Specifically, our subset consists of almost-all English Yelp reviews of Madison area businesses with the tag "Restaurants" as part of its category values. In total, the dataset contains 92,236 reviews of 1,361 Madison businesses.

Methods

Data Preparation

First step, data cleaning: use "strsplit" functions to split texts into words, then remove the space, numbers, and symbols in each "text string" and gain a long vector of strings.

Second step, gaining keywords: we need to find out the frequency of each words appears in the text. If the words do not appears in text much, it may not be helpful when producing model to fit ratings because the predictor can almost be zero in each text.

Third step, gaining predictors: extract the words that has appeared repeatedly in the dictionary and their frequency in each text. Finally add the new developed dictionary and its frequency to the end of "Data.train" and "Data.out" (combination of "Data.test" and "Data.validation").

Fourth step, Lasso regression: using cross validation to choose lambda and then use Lasso regression to select useful predictors.

In total, our dataset contains 2250 predictors.

Modeling

We used log regression for the prediction. Because the predictors we used most are 0. It may has a huge standard variation which will in turn, increase our error term of the final prediction. Thus we converted all of the predictors into "log(predictor appear in each text+1)".

There are some reasons why we convert predictors in such way. We didn't use log because there are many predictors that have 0 terms, and log0 makes no sense. Besides, this step will lower the leverage of some influential words. "Bad", "cool", "delicious" may appear to have coefficients much larger than other words. Adding a log will significantly reduce effects of such words.

Results

R^2	0.6702
RSS	31024.95
TSS	94067.93
ESS	63042.98
RMSE	0.78323
R^2 adjusted	0.6562
Standard error	0.7644

As we can see, R^2 is 0.6702, RMSE 0.78553 and the standard error term is also low.

Inference

Our goal is to test whether the overall predictors are useful in predicting the "stars" given by users. We do not care whether it has a positive or negative connection to the evaluation of customers. The method we used is a two sided F test based on the coefficients of our model. Besides, since we converted data set, all the tests are based on log(predictors appear in each text+1) .

H_0: Coefficients of all the predictors equal to 0

H_a: at least, one coefficient is not equal to 0 using 0.95 confidential level

Residual standard error: 0.7644 on 53091 degrees of freedom
Multiple R-squared: 0.6702, Adjusted R-squared: 0.6562
F-statistic: 47.95 on 2250 and 53091 DF, p-value: < 2.2e-16

p-value: < 2.2e-16 < 0.05 Thus we reject H_0 hypothesis, and accept H_a that at least one predictor should be useful in predicting the stars.

(Below is part of anova table)

```
Response: stars
                   Df  Sum Sq Mean Sq    F value    Pr(>F)
useful              1   748.1   748.1  1280.1519 < 2.2e-16 ***
funny               1   843.3   843.3  1443.0717 < 2.2e-16 ***
cool                1  8288.3  8288.3 14183.3051 < 2.2e-16 ***
nword               1  1943.3  1943.3  3325.3926 < 2.2e-16 ***
sentiment           1 22807.3 22807.3 39028.5945 < 2.2e-16 ***
gem                 1     5.2     5.2     8.8764 0.0028901 **
incredible          1   269.1   269.1   460.4197 < 2.2e-16 ***
perfection          1   184.4   184.4   315.5606 < 2.2e-16 ***
heaven              1   165.5   165.5   283.2349 < 2.2e-16 ***
phenomenal          1   131.9   131.9   225.7334 < 2.2e-16 ***
```

Interpretation

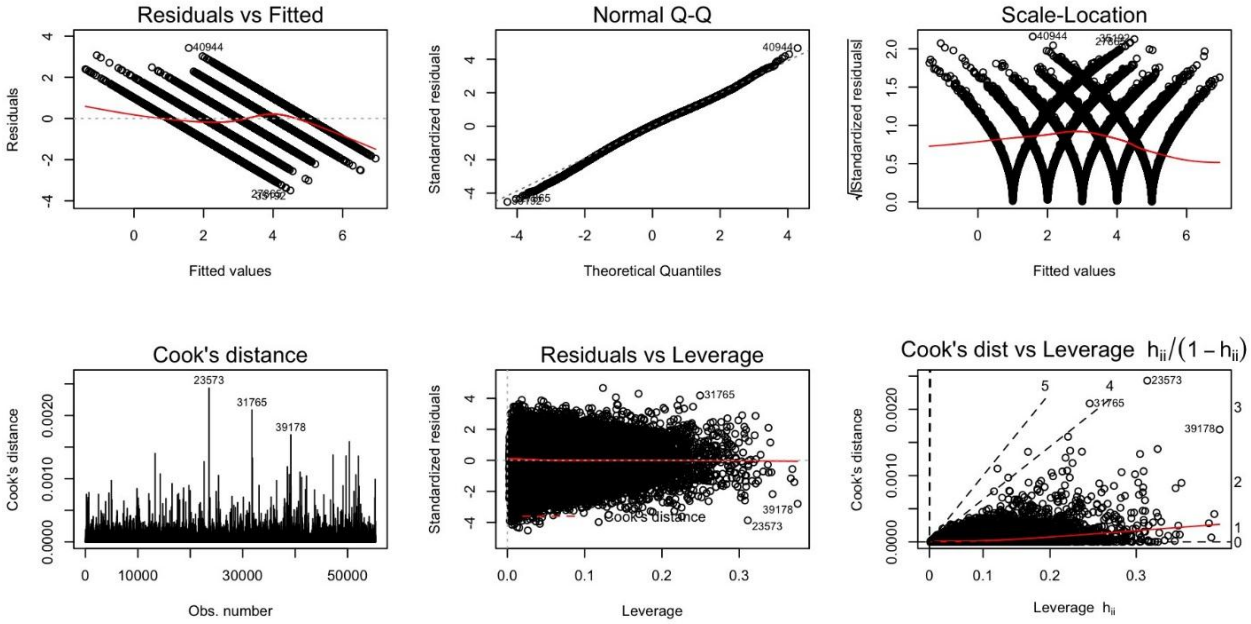
(Part of summary table)

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     4.125e+00  6.600e-02  62.501  < 2e-16 ***
useful         -9.101e-02  6.845e-03 -13.295  < 2e-16 ***
funny          -1.762e-01  1.036e-02 -17.015  < 2e-16 ***
cool            3.044e-01  9.234e-03  32.962  < 2e-16 ***
nword          -4.056e-01  3.215e-02 -12.616  < 2e-16 ***
sentiment       3.221e-01  4.529e-03  71.117  < 2e-16 ***
gem             1.654e-01  4.330e-02   3.821 0.000133 ***
incredible      3.269e-01  4.124e-02   7.927 2.29e-15 ***
perfection      6.626e-02  5.236e-02   1.265 0.205753
heaven          2.216e-01  5.096e-02   4.348 1.38e-05 ***
phenomenal      1.575e-01  9.199e-02   1.712 0.086815 .
```

Because we have transformed the predictors into log(Predictors appear in each text+1), interpretation of the coefficients is different from MLR regression. Take "cool" as an example, its p-value is less than 2e-16, which means that it almost have significant effect on the rating of customer. If a customer uses "cool" one time, the 3.044e-01 means that if he use "cool" one more time while holding all other variables unchanged, the prediction on rating tends to increase 0.3044*(log(2+1)-log(1+1)) = 0.1234236 (we converted the predictor, thus the predictor we used should be log(1+1) and the predictor after increase should be log(2+1))

Model diagnostic

We need to check whether the model violate linearity, normality, homoskedasticity and whether there is influential points in model.



According to the plot above, the model doesn't violate Homoskedasticity and Normality, but linearity may not hold. If we choose 0.03 as critical value then, there is no influential points up to now this is the final model we made

Strengths and weakness

Strengths: We used log predictors, which greatly lowered the error term of the model. Compared to linear models, the RMSE (Root mean square error) is only 0.78323. During model preparation, we used the most frequently appeared words in the text, thus there is less zero terms in our model. Most predictors are useful, thus the model is more efficient. Besides, it holds homoskedasticity and Normality and do not have any influential points.

Drawbacks: Linearity doesn't hold perfectly, and the increase on result is not linear per unit increase on the predictors. The interpretation is not meaningful without linearity. In all of our predictors there might be some predictors shows less influence on the rating.

Conclusion

Based on model building process, sentiment words in each review and sentiment scores are the two most influential factors to make a review positive or negative. Most predictors have a little influence to the ratings, but the prediction based on sentiment words and sentiment score is effective overall. Because of the log transformation, per unit increase on predictors does not lead to per unit increase on Stars. It's had to interpret those predictors.

Peer rating

Jialuo Li: 5 points, mainly worked on coding and extracting dictionary words and summary of model

Bob Dai: 5 points, mainly worked on improving model, splitting text in to strings and presentation