Yelp Summary

Group 4 - Houming Chen, Xiaozhe Cheng, Xinzhu Wang

Background

Yelp is one of the most-popular business directories worldwide that millions of users use to discover and review local businesses and events based on a one to five star scale. Yelp feedback provided by previous customers help customers make better selections and help restaurants make improvements and increase benefits.

Introduction

In this project, the data contains 92,236 English Yelp reviews of 1361Madison area businesses. Our goal is to use R to find out what makes a review positive or negative and predicting a review's rating. Predicted Yelp ratings for the testing and validation data are submitted in Kaggle for competence. We try to remove outliers, do log transformations, eliminate proportional linearity effects, add categories and names to predictors, and use Lasso as the regression model to lower the root mean squared error (RMSE) to the largest extent.

Motivation for the model used and statement of the model

1. Remove outliers by removing high cook distance observations in simple linear regression model between starts and sentiment.

Deleting outliers will help the model to generalize problems in common cases and might lead to better prediction. Since the sentiment score reflect people's attitude to the restaurants, we decided to make a simple linear regression with scores and sentiment and consider observations with high cook distance (cook distance > 0.001) as outliers.

The Id of the outliers are 8509, 10006, 16957, 18442, 19299, 19865, 24577, 29436, 30347, 36943, 38024, 42289, 44097, 46120, and 47467.

2. Use transformations of provided sentiment scores and sentiment scores calculated by R package syuzhet.

The original data provided sentiment scores calculated by AFINN lexicon. We have also found another R package called 'syuzhet' (https://cran.r-project.org/web/packages/syuzhet/index.html) that can calculate sentiment scores for texts in another algorithm.

Sentiment scores are important in predicting people's rating. Therefore, in order to find a better relationship between the sentiment scores and the star rating to improve the prediction of the model, we added a variety of transformations of these two sentiment scores. The transformations are listed as follows:

$sentiment^{0.1},\ sentiment^{0.2},\ sentiment^{0.3},\ ...,\ sentiment^{5.0}$

$e^{sentiment}, \frac{|sentiment|}{sentiment}e^{0.5|sentiment|}, \frac{|sentiment|}{sentiment}e^{|sentiment|}, \frac{|sentiment|}{sentiment}e^{1.5|sentiment|},$
$\frac{|sentiment|}{sentiment}e^{2|sentiment|}, \frac{|sentiment|}{sentiment}ln(|sentiment|+1), |sentiment|$

$syuzhet^{0.1},\ syuzhet^{0.2},\ syuzhet^{0.3},\ ...,\ syuzhet^{5.0}$

$|syuzhet|, \frac{|syuzhet|}{syuzhet}ln(|syuzhet+1|)$

3. Add log(nword), log(nchar), log(useful+1), log(cool+1), and log(funny+1) to the predictors

We found that the data of nchar, nword, useful, cool, and funny are right-skewed. Therefore, we added log(nword) and log(nchar) into our predictors. Since useful, cool, and funny can sometimes be 0, we added log(useful+1) and log(cool+1), and log(funny+1) into our predictors.

4. Find the semantic occurence of the 5000 most frequently appeared words in the training set, and then divide it by nword to construct predictors.

In order to find the relation between the text and the star rating, we decided to construct predictors with the most frequent appeared words, because they are more likely being used in the prediction, and thus more likely contribute to minimizing the rmse. Therefore, we selected the most frequently appeared 5000 words in the texts of training texts to construct predictors.

For each word, simply counting the occurrences of the word cannot accurately reflect the contribution of the word to the meaning of the sentence, because sometimes negators can reverse the meaning completely, and some intensifiers can amplify the meaning. Therefore, we define semantic occurrence of word to be a value representing the occurrence of the meaning of the word. In order to estimate semantic occurrences, We used regular expressions to match five patterns and assign different weights to them to get the semantic occurrence of the word(shown in Table 1). Our intensifiers include “really”, “so”, and “very”, and our negators include “no”, “not”, and every word ends with “n’t”.

Table 1

Pattern	**Weight**	**Example**
Word	1	Simply a word “good” counts as 1 semantic occurence of “good”
Negator + Word	-1	“not good” counts as -1 senmentic occurrence of “good”
Intensifier + Word	1.5	“very good” counts as 1.5 senmentic occurrence of “good”
Negator + Intensifier + Word	-0.5	“not very good” counts as -0.5 senmentic occurrence of “good”
Negator + Other word + Word	-1	“didn’t taste good” counts as -1 senmentic occurrence of “good”

There might be proportional relationship between many frequently appeared words and the number of words of the text. Therefore, we decided to divide the semantic occurrences of each of the 5000 frequently appeared words by nword to be our predictors.

5. Count the occurrence of phrases with “number + stars” form

People might mention their rating in their reviewing text. Therefore, the phrases with “number + stars” form might be useful for the prediction, so we added the 10 variables - occurrence of phrases from “1 star” to “5 stars” and from “one star” to “five stars” - to our predictors.

6. Add categorical data Categories and Names to the predictors

We find that all the categories and most of the names that appear in Train data also appear in Test data, and people might rate similar for the same restaurant or similar restaurants. Therefore, we used 255 categories and 1361 names as categorical predictors.

7. Choose Lasso for the regression model

6762 predictors were used in our model, which is relatively many considering that we have 55327 observations. Since this project focused on predicting with low rmse, we decided to use a penalized regression with cross-validation to prevent overfitting. Also, because the predictors are automatically generated, so it is not sure that each of them contributes to the prediction, we chose lasso regression over ridge regression, because it can delete useless predictors to minimize rmse.

Therefore, we finally used a Lasso regression, and cross validation with 5 folds is used to optimize λ minimizing mse, which is equivalent to minimizing rmse.

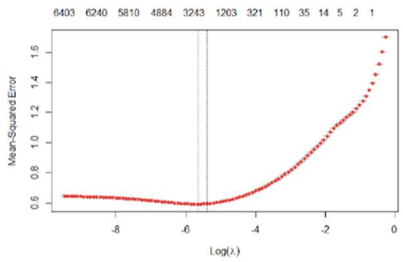
8. Processing the prediction result after the prediction.

In order to have a smaller RMSE, if the predicted star rating was greater than 5, we changed to 5, and if the predicted star rating was smaller than 1, we changed it to 1.

Summary about relevant parameters

The MSE-log(λ) graph is shown as follows (Graph 1):

Graph 1



The minimum MSE is 0.5918591, which is reached when λ = 0.003462392. In the left part of the curve, the MSE decreases when log(λ) increases, while in the right part of the curve, the MSE increases while log(λ) increases. This means the cross validation have found a suitable λ for the lasso regression, and it might be a decent choice to use lasso rather than other models.

To minimize λ, the cross validation of lasso also deleted some useless predictors to minimize the MSE. Only 2352 predictors left after the deletion. The cross validation of lasso assigned β for other predictors. Some of the β for words' semantic occurrence, categories, and names are shown in Graph 2, 3, 4 respectively as examples.

Graph 2, 3, 4

us	-3.330129e-01	Peruvian_category	2.712649e-02		
well	.	EventPlanning.Services_category	-6.083730e-02	A8 China	-2.971435e-02
no	-8.035992e+00	LatinAmerican_category	.	Abarrotes El Primo	1.860116e-01
got	-2.306060e-01	SushiBars_category	.	Abuelo's	-5.670355e-02
best	5.935688e+00	Japanese_category	.	Ace's Main Tap	1.563949e-01
your	-4.041770e-01	Chinese_category	-2.924622e-02	Adamah Neighborhood Table	.
by	.	Diners_category	.	Africana Restaurant and Lounge	.
other	-7.878914e-01	Grocery_category	6.088967e-02	AJ's Pizzeria & Diner	-5.351250e-02
dont	-2.843156e+00	InternationalGrocery_category	.		

The β for the intercept is 3.676445.

The predictor nchar is deleted, while the β for nword and is -1.567624e-03, and the β for log(nword) and log(nchar) are respectively -4.042155e-02 and -1.685394e-14.

The predictor funny is deleted, while the β for cool and useful are respectively 1.610880e-02 and -3.909730e-04, and the β for log(useful+1), log(cool+1), and log(funny+1) are respectively 2.896323e-01, -1.696293e-02, and -1.474377e-01.

For the transformations of sentiment score and syuzhet's sentiment score, only the following terms are left after deletion useless predictors. Their β are detailedly shown in the appendix.

$sentiment^{0.3},\ sentiment^{3.4},\ sentiment^{3.5},\ sentiment^{3.6},\ sentiment^{3.7}$
$sentiment^{3.8},\ sentiment^{3.9},\ sentiment^{4.0},\ \frac{|sentiment|}{sentiment} ln(|sentiment|+1)$
$|syuzhet|,\ syuzhet^{0.7},\ syuzhet^{0.8},\ syuzhet^{0.9}$
$syuzhet^{3.9},\ syuzhet^{4.0},\ syuzhet^{4.1},\ syuzhet^{4.2},\ syuzhet^{4.3},\ syuzhet^{4.3}$

Exact values of β of the predictors are all shown in the appendix.

Interpretation of the estimates

Interpretation for β for the intercept: β for the intercept is 3.676445, which means that, on average, a review with all other variables equal to 0 will have a 3.676445 stars rating.

Interpretation for β for the words' semantic occurrence: Different undeleted words have different β values, which reflects the star rating changes caused by one unit increase of the semantic occurrence of this word. For example, the β of the word "clean" is about 0.65, which means on average, when other variables are unchanged, one more semantic occurance of the word "clean" will lead to 0.65 star increasement.

Interpretation for β for the names and categories: Different undeleted names and categories have different β values, which reflects the star rating changes when the restaurant falls in this category or has the name. For example, the β for Chinese category is about -0.03, which means on average, when other variables are unchanged, restaurants in Chinese category will have 0.03 star less than normal restaurants.

Interpretation for β for nword, nchar, useful, cool, funny, log(nword), log(nchar), log(useful + 1), log(cool + 1), log(funny + 1), and transformations of sentiment and syuzhet: The β of these variables reflects the the star rating changes when these variables increase by 1.

Strengths:

1. The predictions is relatively good and the RMSE of the model is relatively low
2. We considered many factors that might affect the star rating so the model is likely to be more accurate.

Weaknesses:

1. The process of building model is time-consuming and memory-consuming.
2. There are too many predictors which makes the model too complex.

Conclusion

With the 55342 observations in the training set, our model can make a relatively good predictions based on these factors and the RMSE is relatively low which is 0.74618 on kaggle.

Contribution

- Houming Chen: Wrote R code, contributed to the model design, and part of writing up.
- Xiaozhe Cheng: Making powerpoint and part of writing up. Provide ideas and contributed to the model design.
- Xinzhu Wang: Making powerpoint and the background and introduction part of the writing. Also asked questions on piazza.

Peer Review by Houming Chen: Xiaozhe Cheng−5 Xinzhu Wang-5