# Predicting star scores on Yelp Reviews

**Xiner Ning**
Georgetown University
Department of Data Analytics
xn11@georgetown.edu

**Jin Young Yang**
Georgetown University
Department of Data Analytics
jy583@georgetown.edu

## Objective

Predicting the star scores of Yelp reviews is an important task to distinguish which place is the best choice to visit. This document provides a method to predict the star scores of Yelp reviews based on the review comments and the semantic features that we build to analyze. To test the prediction, we use the logistic regression, linear regression, and SVM model to predict the star scores and evaluate each model to suggest which model provides the greatest prediction score. In addition, we use the two emotional features of LIWC to the three models to see if they improve the prediction scores.

## 1 Dataset

The dataset is given as a csv file from the website:https://medium.com/tensorist/classifying-yelp-reviews-using-nltk-and-scikit-learn c58e71e962d9. The dataset contains 10,000 Yelp reviews with the following information for each one: **business_id** (ID of the business being reviewed), **date** (Day the review was posted), **review_id** (ID for the posted review), **stars** (1–5 rating for the business), **text** (Review text), **type** (Type of text), **user_id**, {**cool**, **useful**, **funny**} (Comments on the review given by other users). We select only the relevant features for the analysis: *stars, text, cool, useful, funny*. The star scores are mostly four and five and so we create a binary feature for star scores with two categories, above three and below four. We create the structural and semantic features to build some models: linear regression, logistic regression, and SVM on this dataset.

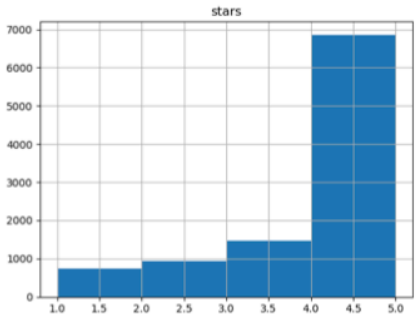


Table 4. Histogram of *stars*

## 2 Background

Yelp is widely used to find a place to eat, play, and exercise. The most important thing to make a decision where to go is the rating and the reviews of the place. In this document we explore the review text to see if we can predict the star rating. In addition, the review comments, given in three categories of cool, useful and funny, have been used to see if they can also contribute to predict the star scores.

With the numerical values of built-in features, linear regression has been used to predict the star scores. Linear regression is useful to model the relationship between a scalar response (star scores ) and explanatory variables (features). Linear models are relatively simple to describe and implement, and have advantages over other approaches in terms of interpretation and inference. However, a standard linear regression can have significant limitations in terms of predictive power since the linearity assumption is almost always an approximation, and sometimes a poor one. In addition, logistic regression has been used in predicting the level of star scores such as above three or less than four. We use the binary categories on the star scores such as over three and less than four to compare with the linear regression. Furthermore, we use SVM regression

model with RBF kernel as suggested in the similar article (Yang Yan Qiu and Bao, 2015). In the end, we evaluate the models with different combination of features to find which one predicts the dataset the best.

## 3 Methodology

Driven by the hypothesis that the star scores depend on the text itself, we consider text-based features only. Features used in the related work, namely Structure (STR) (Xiong and Litman, 2011), and the review comments in the dataset are considered as baselines.

We then introduce two semantic LIWC features, *posemo* and *negemo*; Positive emotion feature counts the number of positive words in each review and negative emotion feature counts the number of negative words in each review. Our rationale to use these features is that the star scores reflect opinions, emotions and personal experience of the review writers.

### 3.1 Data cleaning

Even though the dataset is provided as clean, the review comment features, (*cool, useful, funny*) have some outliers as high as 77. We set a threshold 10 to restrict a potential bias in the model. After removing the observations with the review comment values greater than 10 for cool, useful and funny, the observations decreased to 9,830 from 10,000. In order to better use the comment features, binary features are being generated. The values for *cool* and *useful* above 2 is *True* and the values below 3 is *False*. The values for funny above 1 is *True* and the values below 2 is *False*.

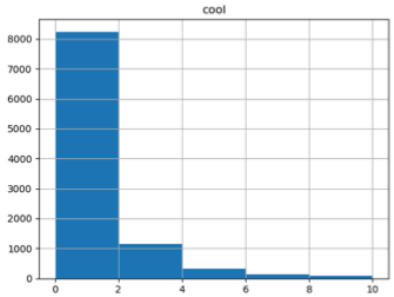


Table 1. Histogram of *cool*

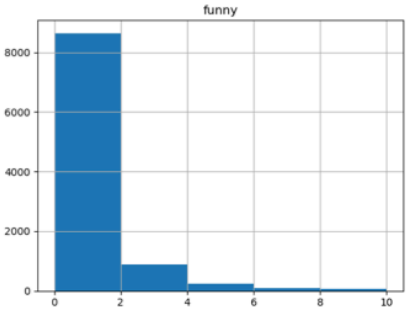


Table 2. Histogram of *funny*

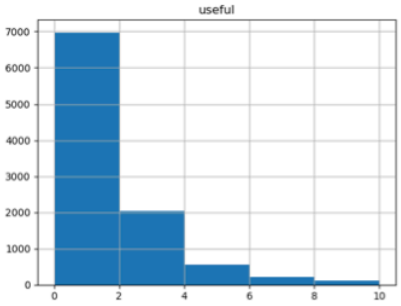


Table 3. Histogram of *useful*

In addition, the star scores are divided into two categories, above three and below four, for logistic regression.

### 3.2 Features

In this article, we use structural features (word length, number of sentences, number of exclamation marks), the three review comments, and semantic features to make a feature matrix. Structural features represent the number of words, the number of sentences, and the number of exclamation marks for each review text. The length of the review can represent the level of happiness or satisfaction towards the business along with the exclamation marks emphasizing this level. The three review comment features (*cool, useful, funny*) are provided in the dataset but changed into binary features as described above.

For semantic features, LIWC has been used to draw the number of emotional words from each review. As shown below in Table 1, LIWC provides many features on each review text but we only take the positive and negative emotion features since in the hypothesis, the emotional level of the review text contributes the star score. LIWC output variables are percentages of total words within a text. For example, if the Positive Emotions (or *posemo*) number is 4.20, it means that 4.20% of

all the words in the review are positive emotion words.

| A | WC | Analytic | number | posemo | negemo |
|---|---|---|---|---|---|
| My wife | 156 | 30.21 | 0.64 | 9.62 | 0 |
| I have no | 262 | 40.18 | 4.2 | 4.96 | 1.91 |
| love the gyro | 16 | 5.18 | 0 | 18.75 | 0 |
| Rosie, | 77 | 93.26 | 0 | 3.9 | 0 |
| General | 84 | 23.74 | 0 | 10.71 | 1.19 |
| Quiessence | 366 | 73.17 | 1.64 | 3.55 | 1.64 |
| Drop what | 293 | 62.79 | 0.34 | 6.14 | 1.37 |
| Luckily, I | 50 | 52.71 | 0 | 12 | 0 |
| Definitely co | 64 | 79.58 | 3.12 | 7.81 | 0 |

Table 5: LIWC feature matrix

### 3.3 Model development

First, we split the dataset into training and test with proportion of 80% and 20% respectively and use the training dataset for the following three models: linear regression, logistic regression, SVM regression.

To test if adding LIWC features improve the three models, we first build the three models with the baseline feature (structural and review comments) and acquire the model scores and error rates. Then, we add LIWC feature (*posemo* and *negemo*) to the three models and acquire the model scores and error rates to compare with the baseline models.

For linear regression, the predictor should be numerical. We use the star scores provided in the dataset for linear regression. We run the linear regression model with baseline features on the training dataset and predict the star scores on the test set. Then, we add the LIWC features on the same procedure. On the other hand, to build a logistic regression, the predictor should be categorical. Thus, we use the binary star score feature that we create in the data cleaning part. We run the logistic regression model with baseline features on the training dataset and predict the star scores on the test set. Then we add the LIWC features in the model on the same procedure. Lastly, we use the numerical star scores to train the SVM regression model on the training dataset and predict the scores on the test set (Yang Yan Qing and Bao, 2015).

## 4 Results

As a result of the linear regression model, the score with baseline features is approximately 0.045 and the score with baseline and LIWC features is approximately 0.149. With the two features, *posemo* and *negemo*, the score increases by around 0.104.

From the logistic regression model, the model score with baseline features is approximately 0.7 and the model score with baseline and LIWC features is approximately 0.75. The score increases by 0.05 by adding LIWC features. The following table shows the confusion matrix of the model.

| | Predict False | Predict True |
|---|---|---|
| Actual False | 57 | 541 |
| Actual True | 47 | 1321 |

Table 6. Confusion matrix with LIWC features in logistic regression

| | Predict False | Predict True |
|---|---|---|
| Actual False | 214 | 384 |
| Actual True | 100 | 1268 |

Table 7. Confusion matrix with baseline and LIWC features in logistic regression

Using SVM regression model, the score with baseline features is approximately -0.038 and the score with the baseline and LIWC features is approximately 0.07. The score increases about 0.1 by adding more features of LIWC.

| | Baseline | Baseline + LIWC |
|---|---|---|
| Linear | 0.0451567 | 0.1491705 |
| Logistic | 0.7009155 | 0.7538148 |
| SVM | -0.0387342 | 0.0746153 |

Table 8. Model scores of the six models

In conclusion, all three models improve with the addition of *posemo* and *negemo* features. This proves the hypothesis that the emotional level of the review writer reflects on the level of star scores.

Moreover, the results of the linear regression models and the SVM regression models are not as great as those of the logistic regression models. Including the LIWC features, the linear regression model could predict 15% correctly and the SVM regression model could predict 7% correctly. This is insignificant and trivial to predict the scores. Since the predictors are skewed to the left, predicting the binary scores are easier than predicting the actual score values. Therefore, the logistic regression including LIWC features predict the star scores the best among the five other models in this experiment.

## 5 Discussion

As the result of the logistic regression including LIWC features is the greatest, we may experiment with more LIWC features to improve the prediction further more.

For example, the emotional features like *anger*, *sad*, and *tone* will help improving the logistic model to strongly prove that the emotional level of the users affects the star scores of the business.

In addition, one may work on balancing the training dataset before fitting it into predictive models in further steps as *stars* feature is strongly skewed. In the cleaned dataset, there are 6740 out of 9830 reviews that have star scores higher than 3. The unbalanced data will lead to predicting most reviews as *True*, namely having star scores higher than 3.

In order to solve this problem, Undersampling technique can be applied. Undersampling uses a subset of the majority, in this case reviews with star scores higher than 3, to train the models. Because it ignores many majority class examples, the training dataset becomes more balanced and the training process becomes more efficient (Liu Wu and Zhou, 2009). More specifically, one can achieve this by first finding the number of reviews with high star scores in the training set. Then randomly sample a same amount of reviews with low star scores from the training set. Finally, combining these two datasets, the training set should have balanced classes.

## References


Yinfei Yang, Yaowei Yan, Minghui Qing and Forrest Shang Bao. 2015. *Semantic Analysis and Helpfulness Prediction of Text for Online Product Reviews*. Association for Computational Linguistics, Beijing, China

Wenting Xiong and Diane Litman. 2011. *Automatically Predicting Peer-Review Helpfulness*. Association for Computational Linguistics, Portland, Oregon.

Soo-min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. 2006. *Automatically Assessing Review Helpfulness*. Association for Computational Linguistics, Stroudsburg, PA

Wenting Xiong and Diane Litman. 2014. *Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews*. International Conference on Computational Linguistics, Dublin, Ireland

Liu, X., Wu, J., & Zhou, Z. (2009). Exploratory Undersampling for Class-Imbalance Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(2), 539-550. doi:10.1109/tsmcb.2008.2007853