Predicting Yelp Review Star Ratings with Language Feature Analysis

Sebastian Cheah
University of California: San Diego
9500 Gilman Dr
La Jolla, CA 92093
swcheah@ucsd.com

Tianqi Chen
University of California: San Diego
9500 Gilman Dr
La Jolla, CA 92093
tic011@ucsd.edu

Linjie Li
University of California: San Diego
9500 Gilman Dr
La Jolla, CA 92093
lil121@ucsd.com

ABSTRACT

For an assignment, we investigate multiple features regarding Yelp reviews in order to construct a predictor for review star ratings. Our supervised learning model uses linear/ridge regression to observe the correlation between a set of features and review star ratings. Basic readily available features include the business' star rating, the user's average star rating, and the total number of votes associated with the review. For advanced features, we discovered that some language processing techniques on review text lead to good features correlating with review star ratings. We combine Latent Dirichlet Allocation (LDA) with other optimizations such as stemming and rounding of edge cases to improve upon the basic feature model. We compare the model's results with a baseline model using the mean squared error (MSE) as the metric. The baseline resulted in an MSE of 1.67836502285. Our model using the features we described resulted in an MSE of 0.732726208483, which is an improvement over the baseline results by %56.342857572

Keywords

Feature Analysis; Linear Regression; Natural Language Processing; Latent Dirichlet Allocation (LDA); Stemming; Regularization

1. INTRODUCTION

Supervised learning is a machine learning task that can help create a predictor given an input training set to infer features from. As such, the extraction of useful features will be key during the learning task.

For our assignment, we use the dataset from Yelp to conduct our experiments. Inside this dataset is information regarding reviews and other associated data.

Our goal is to extract features from this dataset to improve upon a baseline predictor (to be described later). In particular, we conduct an exploratory experiment with natural language processing techniques to determine its viability in extracting good features.

DOI: 10.1145/1235

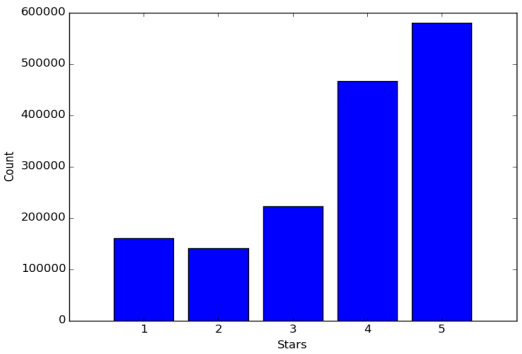

Figure 1: Count of reviews with 1...5 star rating

2. THE DATASET

The current dataset from Yelp we are using is advertised to have:

- **1.6M reviews** and **500K tips** by **366K users** for **61K businesses**
- **481K business attributes**, e.g., hours, parking availability, ambience.
- Social network of 366K users for a total of **2.9M social edges**.
- Aggregated check-ins over time for each of the 61K businesses

Using this dataset from Yelp, we view some properties in order to determine good uses for feature modeling. Yelp's dataset includes three data types useful for review star ratings: user, business, and review data schemas. Our hope is to see if there are features that correlate with a review's given star rating. First, we determine and plot the count of reviews distributed by its respective star rating. This way, we can get a first-look into review rating frequencies. Looking at figure 1, there is a left-skew distribution, with the majority of review ratings being at 4-5.

We then view other readily available features, including the user writing the review and the business being reviewed. With figure 2, we see that the business rating distribution

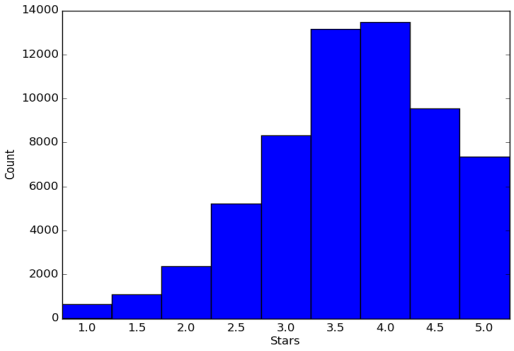

Figure 2: Count of business star ratings with 1...5 star rating (rounded to half-stars)

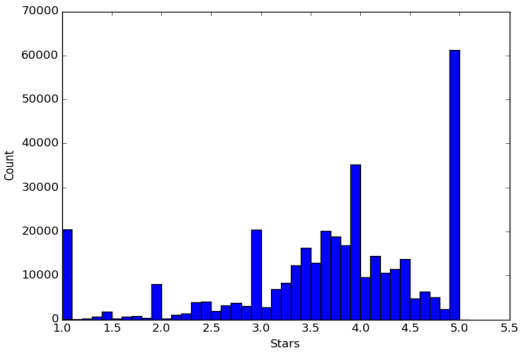

Figure 3: Histogram of user average star ratings

is also left-skew. However, we note that there are multiple reviews for a single business in the dataset, so a correlation between the two distributions cannot be observed immediately. In contrast, user average ratings show a varied distribution that is also left-skewed in figure 3. Another available feature is the number of votes a review receives. Votes are distributed between useful, funny, and cool. Any user can vote at least once, for each vote category. For simplicity, we sum the categories to create a "total votes" feature. Using a scatter plot in figure 4, we can see that the majority of the distribution of ratings is between 3-5 as the number of votes increase. There are a small number of outliers associated with high votes and low star ratings.

We preprocess the data due to the immense size of the dataset. We first take a look at businesses separated by category to retain a well-represented subset of the dataset. Table 1 shows the top 10 categories. There are many other categories that have a small number of business associated with them, and could prove useful for community tasks such as clustering. However, this is outside the main focus of our predictive tasks.

We will process reviews for restaurants only (990627 reviews from 1569264 total reviews). Businesses that are restaurants (21892 businesses from 61184 total businesses). We then take 150000 random reviews from the pre-processed restaurant review dataset. The restaurant review dataset will be further split up into 3 sets with an 8:1:1 ratio split:

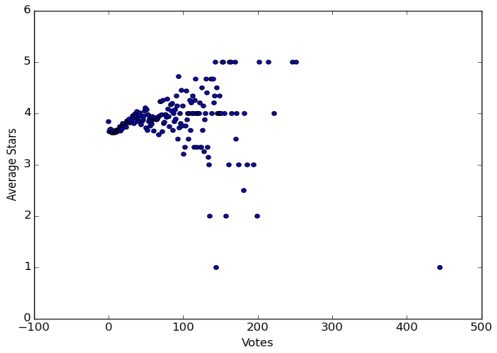

Figure 4: Average number of stars per review with n votes

Table 1: Top 10 Business Category Counts

Business Category	Count
Restaurants	21892
Shopping	8919
Food	7862
Beauty & Spas	4738
Nightlife	4340
Bars	3628
Health & Medical	3213
Automotive	2965
Home Services	2853
Fashion	2566

Training (size 120000), Validation (size 15000), and Test (size 15000).

3. PREDICTIVE TASKS

3.1 Baseline

For our baseline, we calculate the global average review star rating. Using this as our baseline feature, we determine its accuracy through the calculation of the Mean Squared Error (MSE) through the following formula.

$$MSE = 1/N \sum_{i=1}^{N} (y_i - X_i \cdot \theta)^2$$

With a global average review rating of 3.61844166667 based upon the training set, the baseline case (with θ equal to 1) resulted in an MSE of 1.67836502285 with the test set. The baseline is a good measure to compare against when determining whether or not a set of features correlate effectively.

3.2 Linear Regression

Linear regression is an algorithm that models a target value Y using feature values X_f and their corresponding weights θ_f. Note that there is a base feature value of 1 to set an initial bias. The following formula describes the predictor using linear regression.

$$Y = \sum_{f \in features} \theta_f \times X_f$$

Table 2: Individual Basic Feature Correlations

Feature	Initial Bias	Correlation Coef.	Train MSE	Validation MSE	Test MSE
business stars	0.23118219	0.93947228	1.38275398	1.33961795435	1.472758096
user average stars	0.0074526	0.97277473	1.26107949	1.21922160516	1.28202036485
total votes	3.63148846	-0.00687633	1.6517479	1.60593037535	1.67404382551
Above 3 combined	-2.26595628	[0.75764692 0.85405507 -0.00926946]	1.09015062	1.04932669113	1.14555894359

Table 3: LDA Topic Feature MSE's with 10 topics

Text	Train MSE	Validation MSE	Test MSE
Unchanged	1.23284007	1.23991515279	1.24364137738
Stemmed	1.09244281	1.11862814344	1.06806245847

Linear regression applies a linear model that attempts to minimize the sum of squared residuals between predicted and truth values. It is a simplistic model that can be used to quickly check the effectiveness of features in predicting review ratings.

3.3 Predictions with Basic Features

Previously, we viewed the distribution of a review's star rating versus individual features including the business's star rating, the user's average star rating, and the review's total number of votes. Applying linear regression with each feature individually reveals some degree of correlation, as observed in table 2.

When we combine all these features together, we achieve a test MSE of 1.14555894359, which is an improvement over the baseline case's test MSE of 1.67836502285. This is a %31.745542356 improvement over the baseline. There are other possible features to extract from the dataset. We make note of the review's text, and determine new features to add to improve the basic prediction model.

4. FEATURE ANALYSIS AND MODEL

4.1 Latent Dirichlet Allocation (LDA)

In natural language processing, Latent Dirichlet Allocation (LDA) is a topic model generated from input text documents. In our case, the text documents are the texts associated with each review. With this model, each review text can be viewed as a mixture of topics. For each generated topic, there are words with their own corresponding frequencies, which will vary between topics. LDA will generate a specified number of topics from the training set. Each word in the review text can be seen as being assigned a topic. The feature values generated from LDA is a probability distribution based on the frequency and weight of words associated with the generated topics. These statistics can be inferred for the validation and test set when extracting topic model features from the review text.

4.2 Stemming (Pre-processing)

Initial results reveal a correlation between topic distributions and review rating. Using linear regression with just the topic distributions alone, results are shown improving upon the baseline case. However, we make note of another language technique: stemming. Stemming merges different inflections of words, which can help model a more accurate distribution among topics generated from LDA. On another hand, the stemming technique can possibly merge words with different meanings. To demonstrate the effectiveness of LDA with and without stemming, we perform linear regression with two models: one with the review text words stemmed and the other not stemmed. Table 3 show a substantial improvement with stemming before topic generation. With this in mind, all our generated LDA feature distributions will utilize stemming, as it consistently provides improvements in multiple tests. To demonstrate topic modeling and topic words, we provide the top 10 words for 3 of the 10 generated topics from LDA (note that they are stemmed) based off the training set in the restaurant reviews dataset. This is documented in table 4.

Table 4: Top 10 stemmed words for 5 out of 10 generated topics

Topic				
1	2	3	4	5
good	food	order	s	s
sushi	great	t	restaur	t
roll	place	food	locat	place
dish	servic	time	room	like
chicken	good	us	look	can
rice	alway	servic	park	get
order	love	back	old	go
food	time	wait	area	just
like	friendli	ask	strip	good
place	go	tabl	wall	don

4.3 Combining Basic Features with LDA

With the current data, we must see if currently selected individual features have a good relationship with the star ratings. Poorly selected features may add noise during the training phase of our predictor. To verify, we perform a simple linear regression task with 10 generated topic features and basic features which include business stars, user average stars, and total votes. Results finally show the predictor with MSE values lower than 1, as seen in table 5.

4.4 Rounding Edge Cases

We observe that some of our predictions are less than 1.0 or greater than 5.0, which lies outside the range boundaries of possible star ratings. We set predictions less than the minimum, to the minimum, and predictions greater than the maximum, to the maximum. This leads to some small improvements which can be observed in table 5.

4.5 Number of Passes

An LDA model also takes as an input a number of training passes through the training corpus. Previous results were using a single pass. We increase the number of passes to further optimize our features from LDA. Ultimately, an in-

Table 5: Basic Features + LDA, 10 topics, with stemming

Rounding?	Train MSE	Validation MSE	Test MSE
No	0.856218784176	0.854454074693	0.862671573968
Yes	0.845763856351	0.844435046853	0.852894522549

Table 6: Increasing the number of passes on Basic Features + LDA, 10 topics, with stemming and rounding

Passes	Train MSE	Validation MSE	Test MSE
1	0.845763856351	0.844435046853	0.852894522549
20	0.810688013962	0.814545737192	0.82159026054

crease in passes would increase the number of updates of the features, increasing the likelihood of reaching convergence during training. We tabulate our results in table 6, finding an optimal number of passes (20) to be used during topic modeling.

4.6 Number of Generated Topics

LDA takes as input a number of topics to generate based on the training dataset. We model the relationship between the number of topics and MSE of the resulting model combined with the basic features and previous optimizations. In figure 5 we find that as the number of topics increase, the MSE fluctuates between increasing and decreasing. Previously, we included a validation set, but did not specifically use it to tune our model. This is where that set plays a role. We view validation MSE and choose the number of topics yielding the lowest MSE. In our case, 50 topics provided the best results.

4.7 Regularization

We introduce a regularization parameter to see whether or not model complexity is affecting our results. Regularization will penalize model complexity during training. Linear regression with this model will try to minimize the sum of squared residuals plus the regularization parameter λ (this

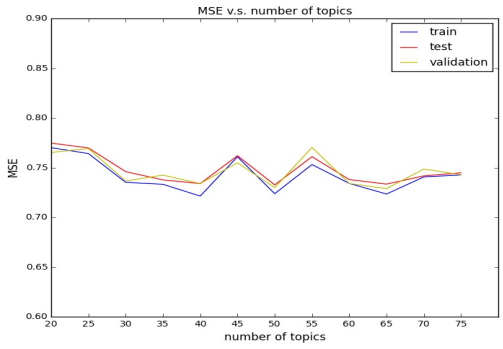

Figure 5: MSE of combined model with increasing number of topics

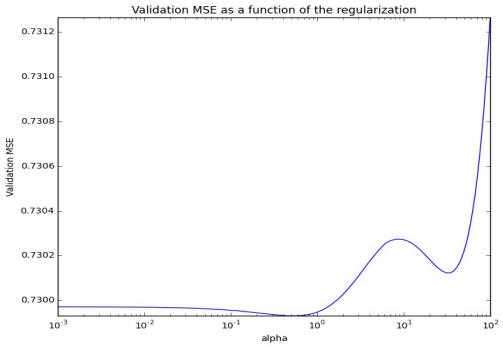

Figure 6: MSE with regularization

is also known as Ridge Regression):

$$\min_{\theta} \sum_{i=1}^{N} (y_i - X_i \cdot \theta)^2 + \lambda ||\theta||_2^2$$

Again, the validation set will be used in determining which model based on the regularization parameter will perform best. We plot the validation MSE values against multiple tuning values and present it in figure 6. Based on the results, we note that regularization did not improve upon our results, as a regularization parameter of 0 yielded the best predictor.

5. RELATED LITERATURE

Personalizing Yelp's star ratings relies on the topic modeling processes that allow us to learn the latent subtopics in review text. Traditional topic modeling lacks methods of incorporating star ratings or semantic analysis in the generative process. In Jack Linshi's paper, an approximation of a modified latent Dirichlet allocation (LDA) in which term distributions of topics are conditional on star ratings was proposed [1]. He assumed that ratings are an approximate function of positively and negatively connoted adjectives and implemented the function by adding two different "codewords", indicating either the presence of a positive or negative adjective, after each positive and negative adjective in the corpus. The approximation of this modified LDA, the codeword LDA, was introduced to show that when examining documents' topic mixture, this approach produces clearer and more semantically-oriented topics than those of traditional LDA.

6. RESULTS AND CONCLUSION

After establishing a baseline case, we attempted to extract useful features from the dataset in order to predict Yelp review star ratings. Readily available features such as business ratings, user average ratings, and total votes provided a moderate increase in predictor accuracy over the baseline case. When we delved into language processing with an LDA model with review text, our results improved even further, and also combined well with the previous readily available features. We do note that while our training MSE decreases, so does our validation and test MSE's. There is a possibility of overfitting on the training data, but the results show that this is not the case with our model.

Table 7: Final Results with our Model: Basic Features + LDA (50 topics, 20 passes) + Stemming + Rounding Edge Cases

Model	Test MSE
Baseline	1.67836502285
Final	0.732726208483

We further try to optimize and improve upon our LDA model. Stemming, rounding edge case, and an generating an increasing number of topics improved upon our feature model. With this increase in feature complexity, we attempted to apply regularization using a ridge regression model. However, this did not see any benefits.

Ultimately, we observed the beneficial techniques with the LDA model and provide final results. We utilize basic features and LDA with 50 topics, 20 passes, with stemming and rounding of edge cases and apply linear regression. With these techniques and parameters, our final results are listed in table 7. Our final model resulted in a %56.342857572 improvement over the baseline model.

Feature inspection using a natural language processing technique such as LDA saw a generous decrease in predictor error. Beyond LDA, there are other language processing techniques that can extract useful data from the review text. One such example is sentimental analysis. To put it simply, sentiment analysis is another natural language processing technique that determines the polarity of a given text. It can determine whether or not the review text is positive, negative, or neutral. It is our hope that such a feature would correlate with a review's star rating. Unfortunately, sentiment analysis proved to have long processing times with our current implementation, and we were unable to generate results given two plus days with our training data.

7. REFERENCES

[1] J. Linshi. Personalizing Yelp Star Ratings: a Semantic Topic Modeling Approach.