Prediction of Yelp Ratings Based on Reviewer Comments Segmented by Business Type

By Kent Lee[1] and James Ross[2]

[1]Stanford University, Department of Structural Biology, Biophysics Program
[2]Stanford University, Department of Computer Science

Abstract

Improvements in machine learning yield new opportunities to make sense of large datasets. Datasets made publicly available by companies offer prospects of finding new insights into how word choice correlate with sentiment. Here, we apply machine learning to a dataset published by Yelp, a business rating website, to predict Yelp ratings based on reviews. By applying machine learning algorithms and improving features, we achieved 59% prediction accuracy. We further discovered that training data segmentation by business type has a significant effect on prediction accuracy, improving by 5% compared to non-segmentation.

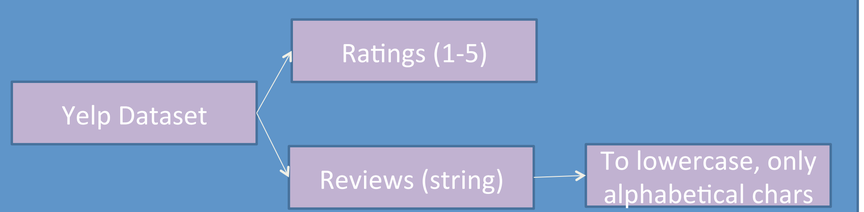
Introduction

To accurately predict Yelp ratings based on reviews, our predictor must quantitatively capture the sentiment of the reviewer with respect to some product or service. To this end, we needed to consider a few questions:

1) Which features to use? How many features to use?
2) Do we want to predict ratings for reviews that span many business segments or focus on one business segment?
3) Which learning algorithm produces the best predictor?

Dataset and Primary Pre-Processing

Yelp's published dataset includes over 1.2 million user-submitted business review texts and corresponding numerical ratings given to over 61 thousand businesses. The dataset is available as an array of JSON objects representing Yelp reviews.



Initial Approach

Do segmented training sets offer higher accuracy when compared against unsegmented training sets? Additionally, do segmented training sets have synergistic effects with other model optimizations? To address this, we start with a baseline comparing accuracies of all businesses and restaurant/food. The baseline results are as follows:

	All Businesses (AB)	Restaurant/Food (RF)
Algorithm	Multinomial Naïve Bayes	Multinomial Naïve Bayes
Vocabulary Size	5000	5000
Training Data Size	6000	6000
Testing Data Size	1000 (naïve)	1000 (naïve)
Training Accuracy	0.72	0.72
Testing Accuracy	0.52	0.54

Iterative Optimization

First we attempt to remove stop words and apply stemming (Figure 1). We notice high variance, suggesting not enough data or too many features, so we optimized feature size from 5000 to 200 highest frequency features (Figure 2). Dropping feature size further to 20, we began to see high bias effects. Then we increased training examples from 6000 to 9000 until convergence.

Figure 1

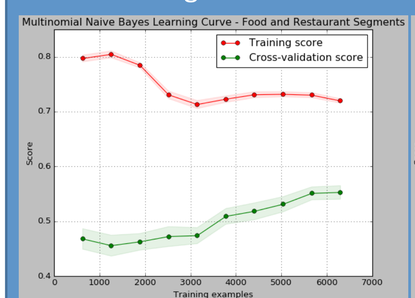
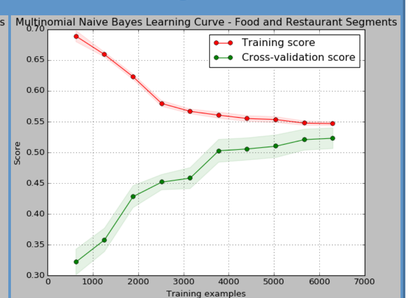


Figure 2



Iterative Optimization (continued)

Improvements	Test Set Accuracy (RF)
Baseline	0.54
Stop Word Removal	0.54
Stemming	0.53
Feature Size Reduction (5000 -> 200)	0.52
Feature Size Reduction (200 -> 20)	0.46 (High Bias)
Increase Training Set Size (6000-9000)	0.56
Manual Stop Word Removal	0.54
SelectKBest (k=500)	0.59

Once feature size and training example size are set, we optimize the features themselves, as choosing highest frequency features may not be optimal. We attempted to manually remove neutral features (ex. burger, food, shake), but this yielded no significant difference. We then attempted SelectKBest from scikitlearn and saw significant improvement in accuracy. The table above summarizes the evolution of our predictor. Running the optimized model on AB and RF, we get accuracies of 0.51 vs 0.59.

Moving Forward

Moving forward, what could improve our predictor? Our hope to do better seems to lie in our feature choices. English is a big language, and this makes for potentially large feature vectors. With only about 10000 examples to work with, we seemed to have suffered from onset of high variance at even a modest number of features. Additionally, our features were simple unigram counts. More sophisticated unigram features like tf-idf weighting with term frequency log normalization may prove more effective. Furthermore, what if we're drawing from a somewhat nonsensical distribution? Yelp's dataset has examples of reviewers lavishing praise on restaurants to which they only give 2 stars. This is another potential weakness of us not having enough data.