Predicting Positive/Negative Yelp Reviews Using Textual Features

Authors: Jon Kaykin & Jin Jeon

Summary:

By analyzing Yelp's dataset, specifically star ratings and text reviews, we created a classifier that predicts whether reviews are positive (star ratings of four or five) or negative (star ratings of one or two). We excluded star ratings of three because we weren't sure whether they were positive or negative.

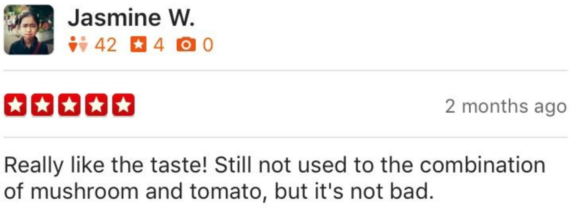
While Yelp's star ratings are helpful for concise overview of local businesses, they are also crucial metrics for businesses as the ratings reflect their reputations. However, we realized that star ratings are often misleading as they are subject to user bias and preference. Thus, we wanted to predict ratings solely based on textual features of the reviews and exclude any potential human errors and biases.

Performing logistic regression with the combined five features, we were able to correctly predict the reviews with an overall accuracy of 79%.

Introduction:

More than ever before, people make decisions of where to visit or what to eat based on other people's opinions. Smartphones enabled access to the Internet everywhere, and we now have better access to other people's reviews and comments. Yelp, the most popular review rating service for local businesses, became the hub for people to share their experience and information. From finding the nearest local businesses to exploring authentic local eateries and ordering food, Yelp now plays major roles in influencing its users' decision making process.

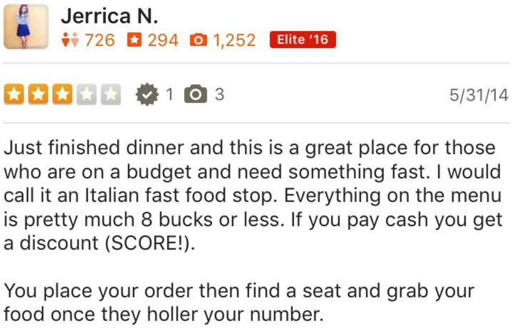
Yelp averages the star ratings to summarize the overall quality and experience of local businesses. It is now a gigantic database for places and local businesses by having its users leave ratings, relevant photos, and reviews. While such crowdsourcing of information allowed us to access different opinions and perspectives from various people, these user-generated reviews are often misleading. Regardless of Yelp's effort to classify quality reviews by incorporating "elite user" and "most helpful reviews" systems, generally the reviews are inconsistent in terms of length, content, writing style, and usefulness because they are written by unprofessional writers. Important information can easily be obscured because users do not tend to leave extensive, thorough reviews. While the star ratings purposefully provide quick overview of local businesses, the ratings can suffer from users' subjectivity and can easily be biased.



Jasmine W.
42 4 0

★★★★★ 2 months ago

Really like the taste! Still not used to the combination of mushroom and tomato, but it's not bad.



Jerrica N.
726 294 1,252 Elite '16

★★★☆☆ 1 3 5/31/14

Just finished dinner and this is a great place for those who are on a budget and need something fast. I would call it an Italian fast food stop. Everything on the menu is pretty much 8 bucks or less. If you pay cash you get a discount (SCORE!).

You place your order then find a seat and grab your food once they holler your number.

The two reviews of Berkeley's famous Gypsy Italian restaurant above illustrate how reviews can be misleading. In the examples above, a user summarizes the overall experience in two lines with notable comments of "really like the taste, but it's not bad" and ends up giving a full 5-star rating. On the other hand, the other user comments how “great” the place is and exclaims “SCORE!” with excitements but only ends up giving it a 3-star rating. Nothing about that review says anything bad about Gypsy’s, so why is there no 4 or 5-star rating?

Looking at how easily the star ratings can be misrepresented under user preference, we became particularly interested in the Yelp Dataset and decided to build a model that classifies star ratings based solely on reviews’ textual features. Because we have seen that human rational and decision making are subjective measures, we wanted to focus on interpreting just the textual features of reviews. By only taking textual features into account, we built a model that predicts the actual reviews without taking user preferences into account.

In this project, our goal was to create a classifier that has at least 75% accuracy in predicting whether the review is positive (star ratings above 3) or negative (star ratings below 3). We hypothesized that positive reviews tend to have more word length, fewer stop words, and more positive words than negative reviews. We spent a lot of time organizing the JSON data and analyzing the review dataset. We came up with five major features to help classify reviews: review length, average word length, stopword counts, positive word counts and negative word counts.

There are previous studies that analyze review text to predict the star ratings. However, most work includes using sentiment analysis or opinion mining, applying machine learning algorithms and n-gram techniques. Although previous studies share the common goal of predicting Yelp star reviews, we incorporate unique textual features to do so.

Methods:

First, we randomly sampled 50,000 reviews from the Yelp review dataset and saved them as a giant dictionary in a numpy file. While we could have just taken the first 50,000 reviews in the dataset, we wanted to remove all possible biases. The numpy file became our new dataset. Each review in the Yelp review dataset contained business_id, date, review_id, text, type, user_id, and votes. We only needed text and stars, so when creating the numpy file, we only wrote those two parameters to the file (the text is the key and the star rating is the value). Another consideration made when creating our dataset was: as Yelp reviews range from 1 star to 5 stars, we thought it wouldn’t be right to conclude that 3 star reviews are positive or negative, so we decided to remove 3 star reviews completely from our dataset, that way 4 and 5 stars are positive and 1 and 2 stars are negative.

Unfortunately, the text data wasn’t clean, so we removed unneeded characters and made the text lowercase to keep things consistent. Also, in order to more easily work with the data, we decided to split each sentence into a list of words. Since the words themselves weren’t the features, we created a featurizer function that takes in a review text list and returns a feature array. We then featurized every review and appended it to the list featurized_revs. The first feature is the average word length, the second feature is the review length, the third feature is the count of stopwords (taken from the NLTK corpus), the fourth feature is the count of positive words, and the fifth feature is the count of negative words. The positive and negative words lists for counting come from

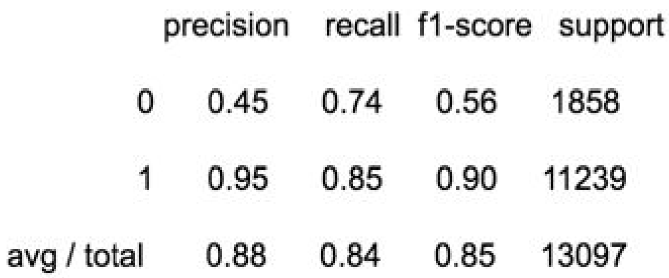
"Mining and Summarizing Customer Reviews"[1]. Once all of the aforementioned was completed, we were left with 43,654 reviews (feature arrays), which we split into a training set (70%: 30,557) and a test set (30%: 13,097). We then, using sklearn, created a multiple logistic regression model based on the feature values in the training set. The model was then used to predict the test data. Finally, we compared each prediction to the actual and received an accuracy rate.

Part 2:

As you'll see in the discussion of our results, we were unhappy with the outcome of the accuracy because we realized that there was extreme bias towards positive reviews, so to aid this we grabbed all the negative reviews from the dataset and randomly sampled that many positive reviews, that way 50% would be positive reviews and 50% would be negative. This new dataset had 20,292 reviews (feature arrays). We then split this into a training set (70%: 14,204) and test set (30%: 6,088). We then, once again, created a multiple logistic regression model using this new training set and compared results. To finish, we needed to see if the accuracy achieved was consistent, so we used 10-fold cross-validation using sklearn's cross validation library.

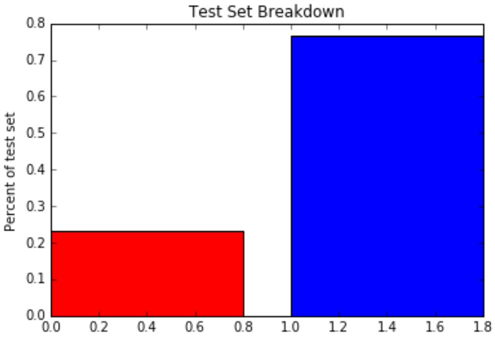
Results:

We received some interesting results. On the test set, we saw 88% precision and 84% accuracy in predicting good or bad reviews (see classification report below).

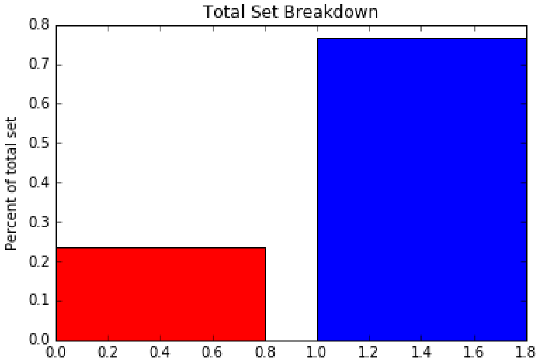


```
             precision    recall  f1-score   support

          0       0.45      0.74      0.56      1858

          1       0.95      0.85      0.90     11239

avg / total       0.88      0.84      0.85     13097
```

You can’t help but notice that precision score for identifying bad reviews (0). Out of 13,097 reviews in the test set, for the ones that the model predicted to be 0s, it was only correct on 45% of them. It turns out that we had a very large bias towards positive reviews, as you can see the test set has 1858 negative reviews and 11239 positive reviews.



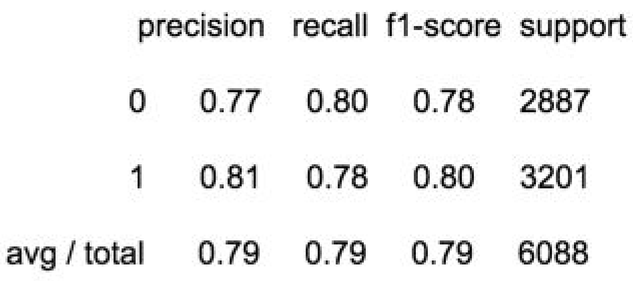
As I sampled randomly, this should be like the full Yelp dataset. Below is the breakdown of reviews for the total Yelp dataset. It looks just like our test set.



What we concluded, by looking at the mean (0.768) of our 50,000 review set, was that we could have 76.8% accuracy just by always predicting yes. So while 84% accuracy is good, only 7.2% of it comes from guessing no.

Part 2

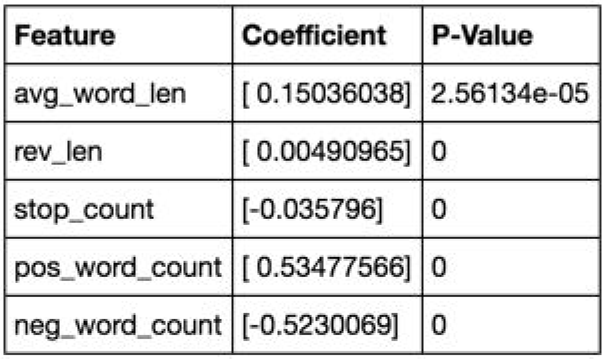
We got much better results when we sampled an equal amount of good and bad reviews from our 50,000 reviews dataset.



```
             precision    recall  f1-score   support

          0       0.77      0.80      0.78      2887

          1       0.81      0.78      0.80      3201

avg / total       0.79      0.79      0.79      6088
```

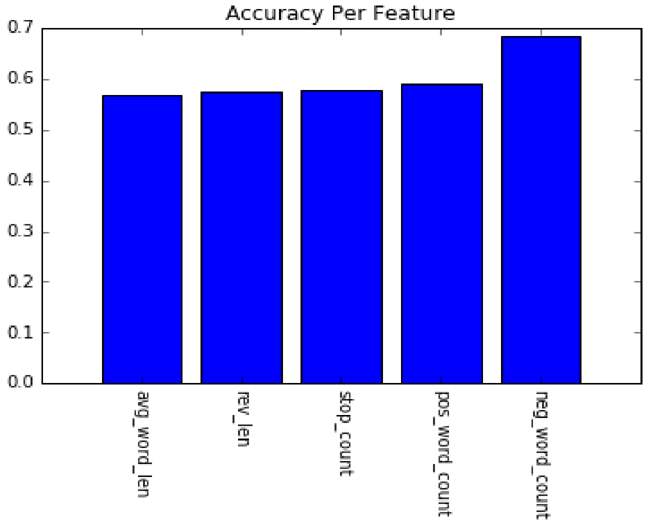
Even though our total precision and recall went down, we were still above our predicted hypothesis of 75%.

Let's take a look at the coefficients to see what is happening:



Feature	Coefficient	P-Value
avg_word_len	[0.15036038]	2.56134e-05
rev_len	[0.00490965]	0
stop_count	[-0.035796]	0
pos_word_count	[0.53477566]	0
neg_word_count	[-0.5230069]	0

As average word length and positive word count increases, so does the likelihood of the review being positive (1). It looks like review length and stopword count don't have much of an influence on the reviews. As the count of negative words increases, so does the likelihood of the review being negative (0). Looking at the individual features for prediction, all but one are below 0.6, negative word count is right under 0.7. If you notice the p-values are 0 for all except the average word length, which is also extremely significant. From this we can deduce that every feature is significant in our logistic regression model. None of these features alone give us at least 75% accuracy, so it's the combination of the features that gives us the accuracy for which we were looking. After running 10-fold cross-validation, we received a 79% accuracy score.



Conclusion:

We were able to successfully predict whether reviews are positive or negative using the five textual features. Our model showed a 79% accuracy rate (4% higher than our original goal of 75%) after cross validation. Our results generally aligned with our hypothesis that positive reviews would have longer average word lengths and more positive word count than negative reviews. However, we were incorrect in our assumption that positive reviews would have fewer stopwords than negative reviews. From our data analysis, we identified key metrics for determining the ratings. We found that average word length and positive word count in text reviews are the two most influential features that in predicting a positive review (star rating of four or five). On the other hand, negative word count was the single most important measure in predicting a negative review (star rating of one or two). The average review length and stopword count had minimal impact on our prediction.

For future work, we may look into more attributes of the data to create a more accurate classifier. Also, we may look into using sentiment analysis and n-gram techniques to potentially observe more interesting results.

References:

[1] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, Washington, USA

Acknowledgements:

Thank you Professor Yang Xu for the inspiration and a fantastic semester.