Yelp review star rating prediction using review text

10 December 2018

Sahithya Sridhar
sahsrid@iu.edu
Indiana University

Prajakta Patil
patilpr@iu.edu
Indiana University

Abstract

Yelp is a website used for searching local businesses. It provides platform to host reviews for various businesses in form of text and star rating given by users. Work done in this project shows star rating of 1 and 5 are easy to predict from text review alone. It is difficult to predict star rating of 2 and 3 from text review.

Keywords Yelp, Sentiment analysis, NLP.

1 Introduction

Yelp is a website that hosts information for various types of businesses such as restaurants, shops, salons etc. Website allows users to provide reviews for businesses that they have visited or just read reviews provided by other users. Users providing reviews can do so in form of either text review or star rating going from 1 to 5 with 1 meaning their experience was not so good and 5 being they really liked the business. Users reading reviews provided by others can click thumbs up or down for whether they found the review *useful/cool/funny*. Reviews help other users in deciding if they want to visit the business for their next outing or not. However, it is not guaranteed as to how many of these reviews are objective. It becomes necessary find if the text review and star rating are well correlated because it is possible that a lot of users want to quickly refer to just the star rating before deciding on visiting a business. This can be done by generating features from review texts and applying machine learning methods to see how well we can predict star rating for a review based on text review. Looking at precision values of these machine learning methods will allow us to evaluate efficiency in making these predictions. Some problems in trying to do this are;

- Ability to process large volumes of data i.e. computing power
- It is difficult to handle usage of ironic or sarcastic language in reviews to train machine learning algorithms to understand true emotions behind such sentences.
- We do not know if reviews provided by users are fair or biased based on their food preferences.

2 Proposed Question

How well can you guess a review's rating from its text alone?
When a user is looking at reviews for a business, they may not always have time to read through full review and would just want to look at rating to make their choice. If the review rating is not capturing the gist of a detailed review, it might mislead the people who use these reviews to make a wrong choice about selecting a business. Knowing how well review ratings match review text can help to suggest Yelp users to make their business choice wisely.

3 Literature Review

(Peter D. Turney,)[1] performed unsupervised learning on reviews provided on Epinions (http://www.epinions.com)) by users. He first assigned part of speech tags to words and then extracted 2-word phrases that followed a certain POS pattern. He then calculated semantic orientation of these phrases for each review. This was done using algorithm called PMI-IR and it stands for Pointwise Mutual Information applied to Information Retrieval. PMI basically calculates semantic orientation (SO) of 2-word phrases found in earlier stage with words *excellent* and *poor*. Higher SO with word *excellent* means phrase is conveying positive sentiment. Higher SO with word *poor* means phrase is conveying negative sentiment. If average SO of all phrases in a review is above 0, review is classified as *recommended*. Meaning, the review recommends product covered in it. Average SO below 0 is classified as *non-recommended* i.e. the review does not recommend the product. Applying PMI-IR on 410 reviews from *Epinions* website, author could achieve average accuracy of 74 % in tagging reviews as recommended or not recommended with highest accuracy of 84 % for automobile category and lowest accuracy of 66 % for movie category. Accuracy was calculated based on comparing PMI-IR classification with actual user review. Turney suggested this method can be used to classify all the reviews on search engines or e-Commerce websites to tag reviews for products as either recommended or not-recommended and present it to interested users in form of what percent of reviews recommend the product and how many dont.

(Fan and Khademi,)[2] worked on predicting Yelp review star rating based on review text only. They randomly chose 1000 restaurants and 35645 reviews for them with 90 % data being testing data and 10 % as training set. Authors started by creating 3 different feature sets from reviews. First were most frequent words from raw text reviews. Second was most frequent words after applying part of speech tags to them and lastly, top frequent adjectives based on part of speech tags. They created buckets of 30, 50, 100, 200, 300, 500 and 1000 words for each 3 feature sets. Then they applied 4 different machine learning methods on these. These methods were Linear Regression, Support Vector Regression with and without normalized features and Decision Tree Regression. Parameter selected to compare results amount 3 different feature sets and 4 different learning methods was Root Mean Square Error. Authors observed that Linear regression had lowest RMSE among all 4 methods. RMSE increased for bucket of more than 500 words for all 3 feature sets. RMSE was lowest for all 4 learning methods for word bucket size of 50-100. Authors suggested that if there is time limitation for applying this method, they would recommend using Decision Tree Regression as it had comparable RMSE with Linear Regression because model training is fast in case of Decision Tree. Another observation made by authors was that normalization reduced RMSE for Support Vector Regression.

4 Data Collection

We chose review.json and business.json files from Round 12 Yelp data set. Reason for choosing this data file was that it has features that were going to be useful to answer the question we were trying to answer. For example, it contains reviewid, userid and text which can be used to predict rating for given review. Using actual rating from this data set, logistic regression and Nave Bayes model can be trained to predict rating based on review text. This data set also has features that tell if review is useful, cool, funny. Finding correlation between review text and these features can provide additional insights about how well does review text get classified into these categories.

5 Data Processing

We followed two approach in Data Cleaning. We first removed data that we were not interested in, next we cleaned noisy data. To remove noisy data, we did the following:

- We removed all nonletter symbols such as , / etc.
- We transformed all words to lower cases. We removed the numbers from the location column.
- We used a feature vector to perform the classification task.
- We used bag-of-words approach to convert the text corpus to vector form where each unique word in a text will be represented by one number.
- We ignore all symbols that are not letters or numbers. Since we are using bag of words model, the sequence of words and sentence structures can be lost, we removed all the punctuations and split every review to a collection of words.
- Stopwords: we deleted all words that do not contain much meaning using stopwords provided by nltk package.

6 Analysis

Our task here was to detect if a review is either bad or good, so we took the reviews that have either 1 or 5 stars from the yelp data frame and stored them in a new data frame called *yelp-class*.

After cleaning the data we have our reviews as lists of tokens. To enable Scikit-learn algorithms to work on our text, we converted each review into a vector. We then used Scikit-learns Count Vectorizer to convert the text collection into a matrix of token counts. The row in the resulting matrix is a unique word, and each column is a review.

we used the *transform()* method on our bag-of-words transformed object and transformed our data frame text into a sparse matrices.

We extracted numerous features relevant to our problem from the structured data, some of which fall into the following broad categories:

6.1 Structural features

- Total number of tokens in a tokenized list of the review: A longer review is expected to have more useful information to the user.

- Number of sentences per review: Similarly, a review with more sentences is expected to be more useful to the user.

6.2 Lexical features

Lexical features are the most relevant features in a text-based model. Lexical features were extracted after removing stop words.

- TFIDF: For tfidf features, we picked the most frequent words gathered from reviews and calculated their tfidf values.

- Unigrams of the most frequent words.

6.3 Metadata features

- Rating (number of stars) associated with each review. We believe that rating is useful in determining the performance. Because a business receiving a higher raring from the customer is more likely to satisfy the customer and perform well in the business. And the customer giving higher rating may tend to write the review more carefully.

- If a customer writes a review casually, it is very probable that he/she will give a rating near average rating. But if the reviewer is subjective enough to overlook the average rating, then the review should include some extra information that most people dont give, and will likely be more helpful.

6.4 Training Models

Next step is processing the review text, by splitting our data review and data star into a training and a test set using *train_test_split* from Scikit-learn. We used 30 % of the dataset for testing.

1. Multinomial Naive Bayes
   - This is a specialised version of Naive Bayes designed more for text documents.
   - After training the model, we predict the ratings of previously unseen reviews (reviews from the test set). We also evaluated our predictions against the actual ratings (stored in y_test) using confusion_matrix and classification_report from Scikit-learn.
2. Random forest
   - Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees [4].
3. Logistic regression
   - Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes) [5]
   - Depending on the words present in the review our model predicts if the particular review is positive or the negative one and predicts the star rating accordingly. 1 for negative review and 5 for positive review etc.

7 Results

For comparing various machine learning algorithms, we decided to use their precision and recall values as evaluation criteria. Reason for choosing precision and recall as evaluation criteria was that these measures show how accurately we can find

Multinomial Nave Bayes			
Review Class	Precision	Recall	F1-Score
Good	0.81	0.92	0.86
Neutral	0.47	0.30	0.36
Bad	0.68	0.62	0.65
Average	0.73	0.75	0.73

Table 1: Multinomial Nave Bayes for all 5 ratings

Multinomial Nave Bayes			
Star Rating	Precision	Recall	F1-Score
1	0.83	0.81	0.82
5	0.94	0.95	0.94
Average	0.91	0.92	0.92

Table 2: Multinomial Nave Bayes for 1- and 5-star rating only

relevant documents (i.e. reviews in this case) and classify them as correct star rating[6].

Machine learning algorithms applied on processed data in 2 different ways showed interesting results. When reviews were classified as Good, Bad and Neutral; application of machine learning algorithms on them had overall lower average precision values when compared to when same were applied to only star ratings 1 and 5.

This makes sense as words used to describe good experience are obviously different than for bad experience.

In case of reviews classified as Good, Bad and Neutral maximum precision and recall was achieved with Logistic Regression. Precision of our model was 77% while recall was 78%.

Highest precision and recall for prediction were achieved with Logistic Regression when they were applied for star rating 1 and 5 only. Our model achieved 94% precision and recall. It is particularly easy to predict star rating 5 from review text only.

Logistic Regression			
Review Class	Precision	Recall	F1-Score
Good	0.85	0.92	0.88
Neutral	0.57	0.46	0.51
Bad	0.68	0.62	0.65
Average	0.77	0.78	0.78

Table 3: . Logistic Regression for all 5 ratings

Logistic Regression			
Star Rating	Precision	Recall	F1-Score
1	0.92	0.81	0.86
5	0.94	0.98	0.96
Average	0.94	0.94	0.94

Table 4: Logistic Regression for 1- and 5-star rating only

Random Forest			
Review Class	Precision	Recall	F1-Score
Good	0.74	0.94	0.83
Neutral	0.47	0.17	0.25
Bad	0.60	0.36	0.45
Average	0.66	0.71	0.66

Table 5: Random Forest for all 5 ratings

This means that our model can predict whether a user liked a local business or not, based on what they typed (based on the review). Comparing 94% precision and recall of predicting only star rating 1 and 5 with 77% precision and 78% recall of predicting all ratings shows how rating 2, 3 and 4 are little more difficult to predict. Specially predicting rating of 2 and 3 is difficult as seen from precision of all 3 algorithms in Table 1, 3, and 5.

Reviews classified as Good stand for star rating 4 and 5; Neutral for star rating 2 and 3 and Bad for star rating 1.

Classifying star rating 1 to 5 as Good, Bad and Neutral had distribution for number of reviews as shown in Figure. 1. Review class Good i.e. star rating 4 and 5 had most reviews. Review class Bad i.e. ones with star rating 1 had least number of reviews.

Looking further into details of distribution of review length distribution as shown in Figure. 2, we can see that most people provide small reviews no matter what if their experience was Good, Bad or

Random Forest			
Star Rating	Precision	Recall	F1-Score
1	0.78	0.56	0.65
5	0.87	0.95	0.91
Average	0.85	0.86	0.85

Table 6: . Random Forest for 1- and 5-star rating only

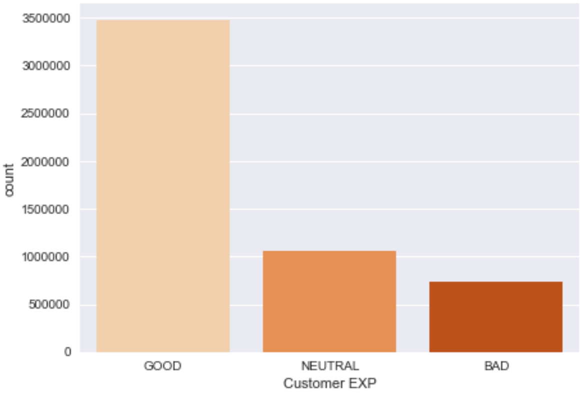

Figure 1: Distribution of number of reviews for different review classes

Neutral. Number of reviews follows same pattern as in Figure. 1 i.e. class Good had most reviews and Bad had least reviews.

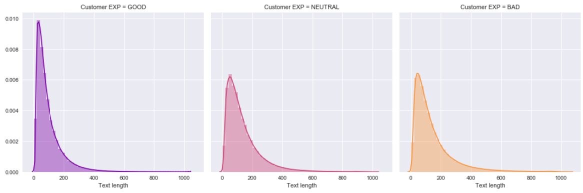

Figure 2: Distribution of review length for different review classes

8 Acknowledgments

We would like to thank our Professor Zheng Gao for his support and guidance. We also extend our appreciation to Yelp website for hosting the data sets. We also acknowledge various online resources which helped us understand Python.

9 Conclusions

Star rating of 1 and 5 can be predicted with high precision from training machine learning algorithms on text reviews. Predicting star rating 2 and 3 from text reviews is difficult. This is probably because words used in these reviews are not as specific as someone would use when they provide star rating 1 or 5.

10 Task distribution

Both of us tried to contribute to almost each section as we both wanted to learn how to perform this type of analysis on real life example. Below are contributions from every team member.

- *Prajakta Patil*: Wrote Abstract/Introduction/Literature review/Data Description/Results in project paper, Data Analysis using Python

- *Sahithya Sridhar*: Wrote Data processing/Analysis methods/Conclusions in project paper, Data Analysis using Python

References

[1] Fan, Mingming Khademi, Maryam. (2014) *"Predicting a Business Star in Yelp from Its Reviews Text Alone"* , https://arxiv.org/ftp/arxiv/papers/1401/1401.0864.pdf.

[2] Peter D. Turney. (2002) *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews.* In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02). DOI: https://doi.org/10.3115/1073083.1073153

[3] *Yelp* (n.d) Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/Yelp

[4] *Random forest* (n.d) Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/Random_forest

[5] *Logistic Regression*, (n.d) Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/Logistic_regression

[6] *Precision and Recall*, (n.d) Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/Precision_and_recall