Predicting Yelp Review Popularity and Classifying Type of Votes

Anonymous ACL submission

Abstract

Reaction upvotes on Yelp indicates the helpfulness, informativeness, and/or entertainment the review brings to readers who come across the review. This paper uses a Logistic Regression model to predict Yelp review popularity with review data by aggregating reaction votes of "useful," "funny," and "cool" to indicate popularity. In addition, a Naive Bayes multi-classifier model is also used to classify which reaction upvote is the best indicator of review popularity. The dataset is a Yelp review comma separated file from Kaggle with 10,000 reviews and 10 columns of review features. Our Logistic Regression model fairly improves from the baseline model, and the Naive Bayes model shows that the "useful" reaction vote is the best indicator of popularity.

1 Introduction

Yelp is a well-known website as well as mobile application for businesses such as restaurants, shopping, and entertainment services. As of February 2021, Yelp has acquired more than 178 million unique visitors monthly across mobile and desktop. The nature of users posting reviews for businesses makes Yelp an ideal platform for researchers in computational linguistics to analyze textual contents with its data using natural language processing, in addition to the great accessibility of Yelp's textual data thanks to their open dataset.

A complete Yelp review consists of multiple features, including the name of the reviewer, reviewer's location, the date of the post, the star rating of the restaurant by the reviewer, a body of review text, and three reaction upvote buttons under the textbox that allows other users who have read this review to indicate their reactions: whether they think this review is "useful," "funny," or "cool."

This research project aims to predict Yelp reviews' popularity with the review text data and classify the type of votes out of the three that would be the best indicator of popularity. The popularity of a review would be determined by the aggregation of all three reaction votes under a review ("useful," "funny," and "cool").

2 Background

Most literature investigating Yelp datasets has been the prediction of star ratings based on the textual reviews. Nabiha Asghar (Asghar, 2016) attempted to solve the review rating prediction problem as a multi-class classification problem with Yelp review data. Asghar built 16 different models by combining four feature extraction methods: unigram, bigram, trigram, and Latent Semantic Indexing, as well as four machine learning algorithms: logistics regression, Naive Bayes, perceptrons, and linear Support Vector Classification.

A great deal of past research also relied on sentiment analysis to extract features from the review text. Qu, Ifrim, and Weikum (Qu et al., 2010) proposed a bag-of-opinions feature extraction method, which extracts opinions that consist of a root word, a modifier or a negation word from the review dataset. Then they computed their sentiment score, predicted a review's rating by aggregating the scores of opinions, and combined it with a domain-dependent unigram model. Yu, Zhou, Zhang, and Cao (Yu et al., 2017) also deployed sentiment analysis on identifying restaurant features on Yelp reviews. They used a support vector machine (SVM) model to differentiate positive and negative words of each review. Word scores generated from the SVM models, either positive or negative, were then analyzed. The results were reported by dividing the positive and negative reviews, and the restaurants were divided into types

of cuisines for comparison.

Past research with similar objectives as the present research have been conducted to predict helpfulness, one of the three reaction aspects of the reviews. Luo and Xu (Luo and Xu, 2019) extracted the main aspects of Yelp reviews, including food/taste, experience, location, and value, with Latent Dirichlet Allocation (LDA), and assigned positive or negative sentiment to each extracted aspect. Due to the lack of votes on "useful," Luo and Xu used combinations of machine learning algorithms including 1. Naive Bates and logistic regression; 2. Naive Bayes and Support Vector Machine; 3. Support Vector Machine accompanied by a Fuzzy Domain Ontology algorithm, to solve the binary classification of review helpfulness problem based on emotion data and best performing features. Lee, Hu, and Lu (Lee et al., 2018) used three categories of features to investigate a review's helpfulness: review quality, review sentiment, and review characteristics. The authors used four classification techniques including decision trees, logistic regression, random forest, and Support Vector Machine, and concluded that review characteristics are good indicators of review helpfulness while review quality and review sentiment are poor indicators of review helpfulness.

In summarization, much of the literature using Yelp review data used machine learning models to extract non-textual features such as location, experience, and reviewer data to predict reviews' helpfulness. Within research on review texts, most research was conducted to predict the star ratings of the review. Therefore, this present research attempts to fill a gap in the absence of literature that predicts review popularity from textual data based on reaction upvotes.

3 Methodology

Two models were used to train and test the dataset including Logistics Regression for binary classification and Naive Bayes for multi-class classification.

3.1 Logistic Regression

A Logistic Regression model uses a logistic function to model a binary dependent variable with two possible values of 0 and 1. The Yelp review data was first vectorized into a sparse matrix with each review as a bag of words as well as their counts converted to a binary variable. In other words, the vote counts of "useful," "funny," and "cool" were summed into one single category of "popularity", and then converted to a binary variable. All numbers above the mean number of votes were labeled as popular, and all numbers below the mean number of votes were labeled as unpopular. This is due to the fact that the distribution of the number of votes received were assumed to be normal, therefore, mean and median were approximated the same. Finally, the normalized review text data was used as predictors to train the Logistic Regression model. Train and test data was split into 80% and 20%.

3.2 Naive Bayes

A Naive Bayes classifier, assuming that the presence of a particular feature in a class is unrelated to the presence of other features, predicts the probability of different classes based on various attributes. The Naive Bayes classifier was implemented to test out the type of votes that would be the most indicative of a popular review. For this reason, rows that were labeled as unpopular were eliminated from the Naive Bayes model dataset as they do not contribute enough meaningful information. The vote counts were also converted into a binary variable by labeling the highest-count type as 1, and the remaining two reaction types as 0. The same corpus used for the Logistic Regression model was also used as predictors for the Naive Bayes classifier. Train and test data was split into 80% and 20% respectively as well.

4 Experiments

4.1 Dataset

The dataset is a 7.72 MB Yelp review comma separated file from Kaggle, last updated in 2018 by the author Omkar Sabnis. The reviews' dates span from 2007 till 2012 and consist of a single domain of only restaurant review data. It contains 10,000 reviews and 10 columns, with columns corresponding to features of the review: business ID, date, review ID, the star rating, the body of review text, type, user ID, and the three reaction votes: cool, useful, and funny. There are no annotations on the datafile.

4.2 Preprocessing

Unnecessary columns on the csv file such as business ID and review ID were removed, leaving the four fields that are only of concern to this project:

the body of review text and the three reaction vote counts. The preprocessing choices were made specifically tailored to the nature of Yelp. Because Yelp is a free-form review platform that allows users to type their opinions with a large degree of freedom, there might be cases where users excessively use capitalization or punctuation marks to express their opinions such as frustration and dislike towards a business. Therefore, words in a review were separated by space and punctuations were removed to generate a bag of words for each review. Preprocessing choices such as case-folding, stemming, and stop-words removal were also implemented so that they can be toggled on and off to test out the differences in the final results. A design choice was also made to eliminate words that only occur once throughout the reviews because they are most likely typos.

4.3 Evaluation Metric

Both Logistic Regression and Naive Bayes models were evaluated on their accuracy, precision, recall, F1 scores, as well as their confusion matrices. The results were also compared to a maximum-likelihood baseline that simply labels all reviews as unpopular as it was the majority label of the reviews. Features such as stemming and case-folding were toggled on and off to test out their impacts on test results. The Logistic Regression model slightly outperformed the baseline model by correctly predicting more popular reviews. From the Naive Bayes classification, the reaction vote of "useful" has the highest precision, recall, and F1 scores to be the most indicative of popularity.

5 Discussion and Conclusion

Overall, Logistic Regression proves to be a good model to predict the popularity of Yelp review votes, as shown in Table 3, 6, and 9. The Logistic Regression model after stemmer slightly outperforms all other parameter combinations. Compared to the maximum-likelihood baseline, the confusion matrix (Table 2) shows that the Logistic Regression model predicts more popular reviews correctly, and the prediction model also has a 0.06 increase in precision of predicting unpopular reviews and 0.50 increase in precision of predicting popular reviews. The baseline model has a recall of 1 for unpopular reviews due to its nature of all-unpopular labeling, while the Logistic Regression model's recall does not decrease by much.

After stemming, both the baseline model and Logistic Regression model have a 0.01 increase in precision of unpopular prediction compared to before stemming. As a result, the Logistic Regression model after stemming has the highest precision in predicting unpopular reviews out of all the parameter combinations in the research. However, the precision of popular reviews decreased by 1 percent, while the recall of popular review and accuracy remained the same as the Logistic Regression model prior to stemming. As a result, stemming does not significantly impact the Logistics Regression model in predicting review popularity. This might be due to the diversity in language used in Yelp reviews, and thus stemming does not bring about a significant impact in word frequency counts.

Case-folding was also toggled on and off to test out the impact of upper- and lower-case usages in prediction of review popularity. As shown in Table 7 and 8, the Logistic Regression model without case-folding does a surprisingly good job in predicting the popularity of reviews based on the precision of popular and unpopular reviews and the accuracy score. This indicates that case-folding might not be a necessary parameter in predicting review popularity, and in fact, it might be worthy to investigate the impact of having upper- and lower-case letters in the review texts on reaction votes.

Results from the Naive Bayes shows that the precision, recall and F1 of the reaction vote "useful" are all more than double higher than those of "cool" and "funny." Additionally, as displayed in table 11, the precision, recall and F1 score of "useful" has a 1-percent increase after stemming. This might suggest that stemming does have an advantage in predicting review popularity among the votes that are thought to be useful, indicating that the "useful" votes might share more similarities in the usage of content words in language than other reviews. However, we should also interpret the statistics with caution considering the size of "useful" is also more than triple of the sum of the "cool" and "funny" counts. Thus, there might not have been enough sample size for "cool" and "funny" to be accurately classified.

6 Limitation

There were several limitations for this research. Besides the textual data of review itself, the factors determining the popularity of a review would also depend on the location, restaurant popularity, reviewer popularity, and reviewers' characteristics such as identity, occupation, and whatever they choose to disclose on their profiles. As a result, in order to develop a comprehensive model to predict reviews' popularity, we need machine learning models that also take into account these characteristics, on top of our NLP models, such as the examples referred to in our literature review.

Another limitation comes from preprocessing, by removing punctuation from our texts, we might risk losing meaningful contents that are relevant to our popularity prediction. For example, a user that uses an excessive amount of punctuation might attract more popularity by invoking the funny reaction from readers.

The final limitation is our bag of words feature extraction method. By using a unigram model, we inherently lose our ability to capture relationships between two or more words that are closer in syntactic or semantic relationship: for example, a word and its modifier like "delicious pizza", or a word and its negation like "not clean."

As a result, future work can take on the direction to expand on this project to investigate the impact of case-folding in preprocessing with more review datasets to compare the prediction results between lowercasing text data versus leaving uppercase letters as they are, as well as investigating the removal of punctuation marks on the impact of predictions. Another potential direction is to research on different feature extraction methods such as bigram and trigram models, and whether they improve prediction results from the bag of words models.

7 Table and Charts

	Precision	Recall	F1	Size
Unpopular	0.66	1.00	0.80	1324
Popular	0.00	0.00	0.00	676
Accuracy			0.64	2000
Macro	0.33	0.50	0.40	2000
Weighted	0.44	0.66	0.53	2000

Table 1: Baseline for Logistic Regression to Classify Popular Votes Using Bag-of-words.

	Precision	Recall	F1	Size
Unpopular	0.72	0.79	0.76	1324
Popular	0.50	0.41	0.45	676
Accuracy			0.66	2000
Macro	0.61	0.60	0.60	2000
Weighted	0.65	0.66	0.65	2000

Table 2: Model Performance for Logistic Regression to Classify Popular Votes Using Bag-of-words.

		Predicted	
		Unpopular	Popular
Actual	Unpopular	1046	278
	Popular	399	277

Table 3: Confusion Matrix for Logistic Regression to Classify Popular Votes Using Bag-of-words.

	Precision	Recall	F1	Size
Unpopular	0.67	1.00	0.80	1331
Popular	0.00	0.00	0.00	669
Accuracy			0.67	2000
Macro	0.33	0.50	0.40	2000
Weighted	0.44	0.67	0.53	2000

Table 4: Baseline for Logistic Regression to Classify Popular Votes Using Bag-of-words.

	Precision	Recall	F1	Size
Unpopular	0.73	0.78	0.75	1331
Popular	0.49	0.41	0.44	669
Accuracy			0.66	2000
Macro	0.61	0.60	0.60	2000
Weighted	0.64	0.66	0.65	2000

Table 5: Model Performance for Logistic Regression to Classify Popular Votes Using Bag-of-words with Stemming.

		Predicted	
		Unpupolar	Popular
Actual	Unpopular	1012	319
	Popular	395	274

Table 6: Confusion Matrix for Logistic Regression to Classify Popular Votes Using Bag-of-words with Stemming.

	Precision	Recall	F1	Size
Unpopular	0.64	1.00	0.78	1288
Popular	0.00	0.00	0.00	712
Accuracy			0.64	2000
Macro	0.32	0.50	0.39	2000
Weighted	0.41	0.64	0.50	2000

Table 7: Baseline for Logistic Regression to Classify Popular Votes Using Bag-of-words.

	Precision	Recall	F1	Size
Unpopular	0.70	0.79	0.74	1288
Popular	0.50	0.38	0.43	712
Accuracy			0.64	2000
Macro	0.60	0.58	0.58	2000
Weighted	0.63	0.64	0.63	2000

Table 8: Model Performance for Logistic Regression to Classify Popular Votes Using Bag-of-words with Case-sensitive Features.

		Predicted	
		Unpopular	Popular
Actual	Unpopular	1005	283
	Popular	441	271

Table 9: Confusion Matrix for Logistic Regression to Classify Popular Votes Using Bag-of-words with Case-sensitive Features.

	Precision	**Recall**	**F1**	**Size**
Cool	0.28	0.12	0.17	93
Funny	0.36	0.18	0.24	133
Useful	0.78	0.91	0.84	700
Accuracy			0.73	926
Macro	0.47	0.40	0.42	926
Weighted	0.67	0.73	0.69	926

Table 10: Model Performance for Naive Bayes to Classify Type of Votes Using Bag-of-words.

	Precision	**Recall**	**F1**	**Size**
Cool	0.14	0.05	0.07	100
Funny	0.41	0.22	0.28	120
Useful	0.79	0.92	0.85	706
Accuracy			0.74	926
Macro	0.45	0.40	0.40	926
Weighted	0.67	0.74	0.69	926

Table 11: Model Performance for Logistic Regression to Classify Type of Votes Using Bag-of-words with Stemming.

	Precision	**Recall**	**F1**	**Size**
Cool	0.20	0.08	0.12	97
Funny	0.35	0.21	0.26	111
Useful	0.80	0.92	0.85	718
Accuracy			0.74	926
Macro	0.45	0.40	0.41	926
Weighted	0.68	0.74	0.71	926

Table 12: Model Performance for Logistic Regression to Classify Type of Votes Using Bag-of-words with Case-sensitive Features.

References

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *CoRR*, abs/1605.05362.

Pei-Ju Lee, Ya-Han Hu, and Kuan-Ting Lu. 2018. Assessing the helpfulness of online hotel reviews: A classification-based approach. *Telematics and Informatics*, 35(2):436–445.

Yi Luo and Xiaowei Xu. 2019. Predicting the helpfulness of online restaurant reviews using different machine learning algorithms: A case study of yelp. *Sustainability*, 11(19).

Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, page 913–921, USA. Association for Computational Linguistics.

Boya Yu, Jiaxu Zhou, Yi Zhang, and Yunong Cao. 2017. Identifying restaurant features via sentiment analysis on yelp reviews.