Multiclass Text Classification: Predicting Yelp Star Ratings with NLP

Stephanie Buchanan

August 25, 2021

1 Executive Summary

The goal of this project is to use natural language processing (NLP) techniques and machine learning algorithms to predict a Yelp star rating based on the text of the review given. This would be useful for classifying reviews that do not have a star ratings, for example small businesses, or movie reviews.

Three supervised machine learning algorithms were assessed: (1) Multinomial Naïve Bayes, (2) K Nearest Neighbors Classifier and (3) Support Vector Classifier. The one with the highest accuracy score was the support vector classifier with an accuracy score of 62% on the test set.

2 Data

The data was sourced from the Yelp Dataset available online. There are 4 main datasets on the site that include one for business, review, users and check-ins. The one that was used in this analysis was the reviews data set. This dataset included features for 'review_id', 'user_id', 'business_id', 'stars', 'date', 'text', 'funny', 'useful', and 'cool'. For this analysis, the 'stars' and 'text' dataset were used.

The data was first explored looking at the distribution of the count of reviews for each star rating category. This visualization is shown in Figure 1. It was observed that the higher star ratings of 4 to 5 had a larger count of reviews than the lower ratings of 1 to 3.

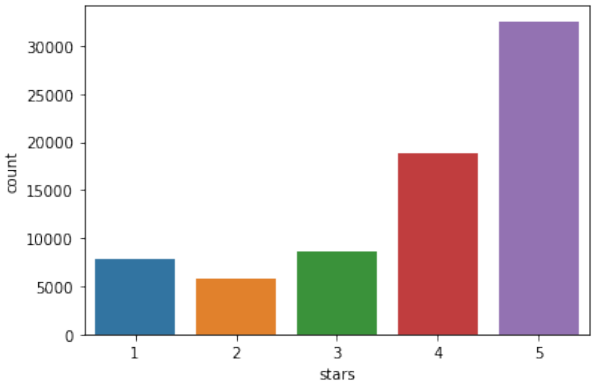

Figure 1: Count plot for number of reviews for each star rating category.

The length of reviews was also explored within each star rating category. This visualization is shown in Figure 2. As seen in the plot, the higher start ratings of 4 and 5 have longer reviews than the lower ratings of 1 to 3.

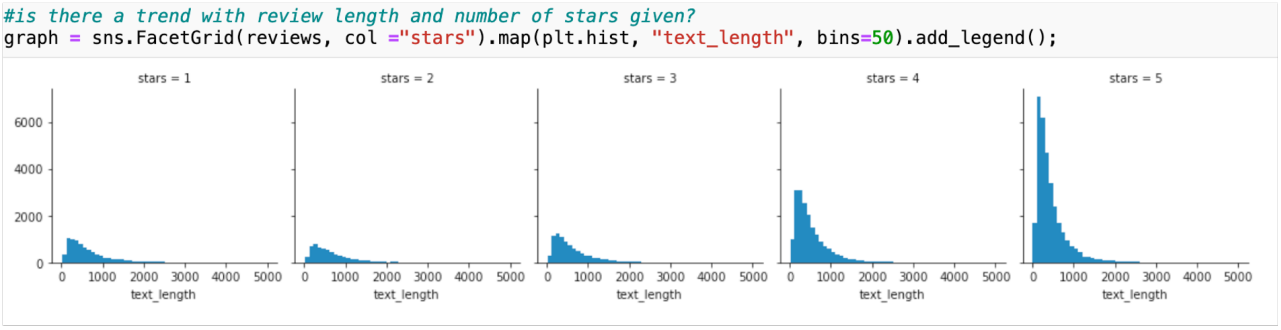

Figure 2: Histograms of review length for each star rating category.

The data was split into train, validate and test sets using a 70/20/10 split. Stratification was used during the split to get a good sampling of each star rating into each dataset.

The goal of the training dataset is for model building. Usually, it is the biggest proportion of the data in order to improve the model as much as possible. The training set can also be used to develop models with different hyperparameter settings.

The validation dataset is used primarily for model selection, and to fine-tune parameters of the model. The model does not do any 'learning' with this dataset, but the results of the tuning does update the hyperparameters.

The test dataset is used to get an unbiased evaluation of the final model fit on the training dataset. It is only used once the final model is selected to ensure the model is being evaluated on 'unseen' data.

3 Research Question

Is it possible to predict the star rating based on a customer's review? If a good predictive model can be generated with the review text, this would help classify reviews that do not have star ratings.

4 Methods

NLP is described as a branch of artificial intelligence with roots in computational linguistics[1]. These techniques help computers understand, analyze and manipulate human languages. For example, NLP makes it possible for a computer to read text, interpret it, determine the most common phrases, measure sentiment, and measure objectivity of the text.

The main platforms used in the analysis of Yelp ratings dataset was the sklearn package and the Natural Language Toolkit[2] NLTK provides a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

TfidfVectorizer transforms text to vectors with weights that represent the tf-idf score, it is a CountVectorizer followed by TfidfTransformer. The input is the corpus, or collection of documents, and the output is a matrix with columns as the unique words, and the values in the row represent the TD-IDF value.

The TD-IDF value is a statistical measure that evaluates how relevant the word is in a document. It is calculated by multiplying how many times the word appears in a document and the inverse document frequency (IDF).

The IDF is expressed as: $idf = log\frac{|D|}{|d:t_i \in D|}$, where D is the number of documents in the corpus, and $d : t_i \in D$ is the number of documents in which the word appears. If the word t_i appears in every document of the corpus, the IDF is equal to 0. The fewer documents the word t_i appears, the higher the IDF value.

One important parameter in the TfidVectorizer is the 'ngram_range'. An 'n-gram' is defined as a sequence of n words in some text. The parameter was initialized at (1,1) to look at only unigrams, and tuned later using the validation dataset in the analysis.

The Naive Bayes Classifier Algorithm applies Bayes' theorem with the "naive" assumption of conditional independence between every feature pair. Naive Bayes is used a lot in text classification and calculates the probability of each class for a doucment and assigns the class with the highest probability.

KNN and SVM are two commonly used supervised machine learning algorithms. Both of these algorithms can be used for classification of text or images. KNN is a non-parametric algorithm,

[1] source: https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html

[2] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.

which means that the method does not assume anything about the form of the mapping function other than the observed patterns that are close likely have similar output variables. The non-parametric property of the KNN algorithms makes it more versatile.

SVM is also non-parametric model and starts by transformation of the data from a low dimension into a higher dimension. The next step is to separate the two classes to finally return the proper label for the test samples. It is a typically a binary classification technique, but can also be used in multi-classification applications. To use for a multi-class dataset, the parameter 'decision_function_shape' is set to 'ovo'. The algorithm relies on the training set to predict a hyperplane separating the classes in an n-dimensional space.

KNN starts with a 'K' that is chosen by the user and this value tells the algorithm how many neighbors should be considered for classification. In the second step of the algorithm, the model checks the distance between the target example and all the other observations in the dataset. The distances are then added to a list and sorted. The last step is to check the sorted listed and the labels for the top K values are returned. Different distance metrics can be used such as Euclidean, Manhattan and Minkowski.

The main limitations when using KNN is choosing the wrong value to K to start and the large time investment to run. If the wrong value of K is selected, the predictions that are returned can be very off. The main limitations to SVM are the extended training period needed to tune the hyperparameters and kernels, and the time investment to run when the values are not linearly separable. SVM also depends on the assumption that there exists a hyperplane that separates the data points.

5 Results

The three models, 1) Multinomial Naïve Bayes, (2) K Nearest Neighbors Classifier and (3) Support Vector Classifier, were implemented using the default hyperparameters using the training dataset. The accuracy of the model was computed and these results are shown in Figure 3.

As shown in the figure, the model with the best training accuracy was the Support Vector Classifier. This model was therefore tuned with a cross-validation folds of 3 and using the validation set in a GridSearchCV.

model	accuracy, Train
MultinomialNB	0.6145
KNeighborsClassifier	0.452
SVC	0.9222

Figure 3: Accuracy results of the different models with the train set.

The hyperparameters tuned were the 'C' and the 'kernel' parameters for the classifier and the 'min_df' and 'ngram_range' for the TfidfVectorizer. The grid search returned the best parameters being the following values: C = 1.0, kernel = 'linear', 'min_df' = 1, 'ngram_range' = (1,2). This

resulted in an training accuracy of 62.2% and a test accuracy of 62.07%. There does not seem to be over-fitting in the model as the accuracies for the train and test sets are close, but the accuracy could be improved.

6 Conclusions

NLP was used in this study to determine if the star rating of a business on Yelp can be predicted using the text in the review. The Support Vector Classifier had the highest initial training accuracy score when compared to Multinomial Naïve Bayes and K-Nearest Neighbors algorithms using default initialization values. The cross-validated test accuracy for the Support Vector Classifier model was 62.07%, meaning cross-validated accuracy scores should have been computed before selecting a model to tune.

To improve the model performance, it could be considered to apply the TfidfVectorizer on separate star rating subsets instead of the whole corpus. This could improve the model since the higher star rating reviews were longer meaning more words in those documents could influence off the term weights. Evaluating the model across different business types – hotel, restaurant, medical, etc – could also be considered since the words used in each type could change. Selecting another model altogether, such as the Multinomial Naïve Bayes model, is recommended.

7 GitHub

The jupyter notebook for this analysis is saved on my GitHub Portfolio along with the mini-projects completed throughout the term.