Nick Videtti

nvidetti@syr.edu

Assignment 6

Multinomial Naive Bayes, Bernoulli Naive Bayes, and Decision Trees on Text Data

SciKitLearn’s MultinomialNB, BernoulliNB, DecisionTreeClassifier, CountVectorizer, and TfIdfVectorizer

5/24/2023

| Restaurant Reviews and Their Validity - Part 2 | |
| --- | --- |
| **Introduction** | Restaurants are a big part of most social lives. They are often where friends go to meet up after a long week of work, where families go to experience something new, where couples go for a romantic date night, or where a bachelor gets his favorite meal for takeout on Tuesday nights. Regardless of the person or lifestyle, food is an integral piece of American culture, and the restaurant is at the forefront of food culture. This makes a restaurant’s reputation as valuable as any factor towards its success. Having a strong and consistent customer base with a large number of “regulars” will make any restaurant successful, regardless of even something as important as the restaurant’s quality.    All that said, new customers need to come from somewhere, and the reputation of a restaurant often drives that through reviews. It is extremely common to look up a restaurant online before deciding to try it out for the first time, and usually an exceptional rating with mostly exceptional reviews are what it takes to convince prospective customers to go to that new restaurant rather than choosing another tried and tested option.      Sadly, with the continued advancement of technology and the “good old fashioned” phenomenon of people not being truthful, many reviews are fake and/or false. This can unfortunately make a good restaurant fail over time by bringing down its prestige and hurting its reputation.  Putting all of the pieces together, this study will try to find exactly how to identify reviews by whether they are positive or negative and whether they are truthful or fallacy. It will also be attempted to identify both positive/negative sentiment and truthfulness at the same time. 92 restaurant reviews were supplied with their corresponding sentiment (positive or negative) and truth (true or false) and will be used to conduct some experiments. Part of these reviews will be used to make predictions based on the words that are in each review, and the remaining reviews will be used to see how accurate those predictions are. |
| **Analysis** | These analyses will be done using SciKit Learn’s Multinomial Naive Bayes classifier (sklearn.naive\_bayes.MultinomialNB), Bernoulli Naive Bayes classifier (sklearn.naive\_bayes.BernoulliNB), and Decision Tree Classifier (sklearn.tree.DecisionTreeClassifier). Numeric data are needed in order to use all of these algorithms, and the Bernoulli algorithm specifically will need boolean/binary numerical data. This will be addressed by converting each review into a vector using SciKit Learn’s vectorizers available in sklearn.feature\_extraction.text. Both CountVectorizer and TfIdf Vectorizer will be used to create both binary and non-binary document vectors, resulting in four sets of vectors that all will be tested using each of the 3 algorithms, where possible.  Before that can be done, the first step is to pull in the data, which comes in the form of a CSV file. This file contains three fields: one for truth called “lie” (either t or f), one for sentiment (either n or p), and the text for the review itself. These data will be read into a Pandas DataFrame. After that, four more columns will be created: lie\_sentiment, words, cleaned\_words, and cleaned\_review. “lie\_sentiment” will be “lie” concatenated with “sentiment” (will take values of either “tp”, “tn”, “fp”, or “fn”). “words” will be a list of the words from the review, generated using NLTK’s word tokenizer. “cleaned\_words” will be “words” with all word phrases converted to lowercase. Also, each list in “cleaned\_words” will contain only word phrases that do not contain numbers and will exclude any word phrases that fall into NLTK’s English stopwords. Also, “cleaned\_words” will be filtered to only phrases that are between 3 and 13 characters long. Once “cleaned\_words” is good to go, “cleaned\_review” will be created using the cleaned words with each list being combined into a single string. This will be the version of the review that is used for the remainder of these analyses.  Once the “master” DataFrame just described is created, the next step is to prepare data for the classifier models. This will require subsets of the “master” DataFrame for each analysis, one for “lie”, one for “sentiment”, and one for “lie\_sentiment”. Each of these will contain its corresponding label with the cleaned reviews from the “cleaned\_review” column. For each of these subsets, they will need to be converted into four different document vector DataFrames using a combination of Pandas, SciKit Learn’s CountVectorizer, and SciKit Learn’s TfidfVectorizer. For the Multinomial Naive Bayes and Decision Trees, each will use both binary and non-binary vectors for each vectorizer, but Bernoulli Naive Bayes will only use the binary vectors since the algorithm itself requires as such. Each of these DataFrames will go through a 10-fold cross validation using Multinomial Naive Bayes, Bernoulli Naive Bayes, and Decision Trees. Then, the minimum accuracy, maximum accuracy, and average accuracy of the cross validation will be recorded.  Due to the large number of dataframe-vectorizer-algorithm-measure combinations, 90 to be exact, a function will be created and then will be iterated through three times, once for each dataframe. This way the number of combinations to code is 30 and the function will just need to be called 3 different times.  Once this is complete, all measure results will be printed, aggregate confusion matrices will be printed, and the best performing model (based on the average accuracy) for each DataFrame will be printed. For the confusion matrices, code testing showed that sometimes testing data did not include all possible classifications (specifically for the 4-class lie\_sentiment data), which would break the code. This was fixed, but the solution may result in some of the confusion matrices not including results from all ten folds. |
| **results** | The results of the analyses described were as follows.  ***“Master” DataFrame - First 10 Rows and First Few Columns***    ***“Master” DataFrame - First 10 Rows and Last Two Columns***    ***lie Data - First 10 Rows***    ***sentiment Data - First 10 Rows***    ***lie\_sentiment Data - First 10 Rows***    ***lie Data - Multinomial Naive Bayes Model Results***    ***lie Data - Bernoulli Naive Bayes Model Results***    ***lie Data - Decision Tree Model Results***    ***lie Data - All Results and Best Model(s)***    ***sentiment Data - Multinomial Naive Bayes Model Results***    ***sentiment Data - Bernoulli Naive Bayes Model Results***    ***sentiment Data - Decision Tree Model Results***    ***sentiment Data - All Results and Best Model(s)***    ***lie\_sentiment Data - Multinomial Naive Bayes Model Results***    ***lie\_sentiment Data - Bernoulli Naive Bayes Model Results***    ***lie\_sentiment Data - Decision Tree Model Results***    ***lie\_sentiment Data - All Results and Best Model(s)***    For the “lie” and “sentiment” data, a baseline target accuracy to surpass with two classifications is 50%. For the four-classification “lie\_sentiment” data, being over 25% is the baseline target. That said, even the best “lie” models had 54% average accuracy, which implies the models are not too much better than a simple guess and are possibly worse than an educated guess. The “sentiment” models performed much better, with the best model having an 82% average accuracy. The best “lie\_sentiment” model had an unimpressive average accuracy of 42%, but that does still beat the 25% baseline target that was suggested. Overall, it seems sentiment is the easiest to predict, but the usefulness of that only goes so far if the truthfulness can’t be predicted with significant accuracy. |
| **conclusions** | Much like the first part of this study, the results suggest that it is actually quite difficult to tell whether a review is truthful or not based on the words in the reviews. It was however much easier to tell whether the review was positive or negative. As far as predicting both at the same time, it was somewhere in between, but still not very impressive. It was actually found that guessing whether a review is truthful is likely more effective than any results that came from this study.  These are disappointing findings. While sentiment is the easiest to predict, that will have limited significance if it can not be paired with at least moderately accurate ways to determine truthfulness. Also, the sentiment of a review is almost always already known through some sort of supplemental rating, and even if it is not known, it is very easily determined through reading.  Although this has already been investigated twice now, there is still more work to be done. The end goal is still to find a way to identify whether or not a review is fallacy based on the words in the review. While it seems unlikely that this goal will be attained, there are many other ways to approach this problem, and some more of those ways will need to be explored before meaningful results hopefully occur. |