Naïve Bayes for Natural Language Processing of Amazon Reviews

Nyssa Bulkes

Capstone #1 – Springboard Data Science Career Track

Introduction

When thinking about buying something, product reviews offer insight before money ever leaves a customer's wallet. Reviews can tell a potential customer what quality and functionality they can expect and also what could be improved, and as a result, customer reviews can directly impact a company's sales. However, in a world where data is abundant, review data can do more than just inform future customers about what to buy. Rather, review data can provide honest feedback to companies looking to improve their bottom line. Paying attention to what customers think about a product is key to defining any successful business. Rather than mindlessly crank out a product without considering what a target demographic might think about the product, savvy businesses will attend to how their customer base already perceives what they're purchasing and use that as a metric of how the product will perform in the future. To do this, analyzing reviews written on a company website and treating it as data, companies can learn directly and without sugar-coating how well a product is doing in a given market. This textual feedback can be directly leveraged back into the company as business insight and can inform an organization how they might tailor and improve a product to lead to better future sales. By deriving insights from customer reviews, companies can better understand what people think about a product and, further, what concrete steps they can take to better meet the needs of their consumers.

Natural language processing (NLP) involves applying computational algorithms to textual data to characterize it without needing a human to read through the text manually to derive meaning. Although these algorithms require data to be represented numerically, NLP facilitates automatic analysis of textual data through various approaches (i.e. tokenization, vectorization). Further, by utilizing algorithms to analyze text, we can learn other things about

the data other than what it means. For example, by using informative data visualizations, we can use tools such as word clouds to display the most frequent words appearing in a given text. We can also apply techniques from the domain of sentiment analysis to determine whether a text is positive or negative and, further, which words in that document make it more likely to be categorized as positive or negative. Through the years, NLP has matured to not only automate analysis of text but to use that automation and the power of computing to derive meaningful insights about a text that would be cumbersome and tedious if a human were to do it on her own.

In this paper, I utilize a publicly available dataset to achieve three objectives:

1) to utilize EDA to explore a text-based dataset
2) to build a Naïve Bayes classifier to characterize textual data
3) to expand on the Naïve Bayes approach by incorporating term frequency inverse-document frequency to explore other ways in which text data can be analyzed.

From an exploratory perspective, I present an introductory approach using NLP to understand what text data can tell us about a product while still remaining critical and observant of caveats that come with analyzing a dataset of reviews (i.e. What does the distribution of reviews tell us about the reviewers as a population?). In the next section, I discuss how the data in more specificity and describe how they were cleaned and handled.

Method & Analysis

Data acquisition & cleaning

The data were originally posted on Amazon.com from customers who were reviewing the company's line of Amazon Alexa and Echo products (linked here). The dataset was obtained

from Kaggle, publicly available at no cost. Participants are, presumably, Amazon customers or individuals who have interacted with an Amazon product following purchase. To be eligible to leave a review, according to Amazon's website, a person needs to have spent at least $50 on a credit card at Amazon in the previous 12 months. On their "Community Guidelines" page, it further specifies that individuals in the same household are not allowed to post multiple reviews of the same product. From this, we could assume that each observation comes from a unique household, but it is unclear how this guideline is policed. It is possible that Amazon prevents users who share an Amazon account from posting multiple reviews, but people from the same household might hold different accounts. Ultimately, while this is stipulated on the company's website, it is taken with some caution, specifically as it pertains to characterization of the observations as completely independent of one another.

To clean the data, rows with missing values or no text review were dropped. A total of 3150 observations, or "verified reviews", were included in further analyses. To explore the data descriptively, the .info() and .describe() methods were used. This exploration showed that the reviews span May 16, 2018 and July 31, 2018. Interestingly, there are 16 variations of Alexa represented in the data; however, these variations seem to represent mostly aesthetic differences in fabric used for the device, and this dimension is not explored further.

Data visualization

Next, data were grouped by numeric rating (scale of 1-5, 1=low; 5=high) and, to get a sense of the data's distribution, I plotted a histogram (**Figure 1**).

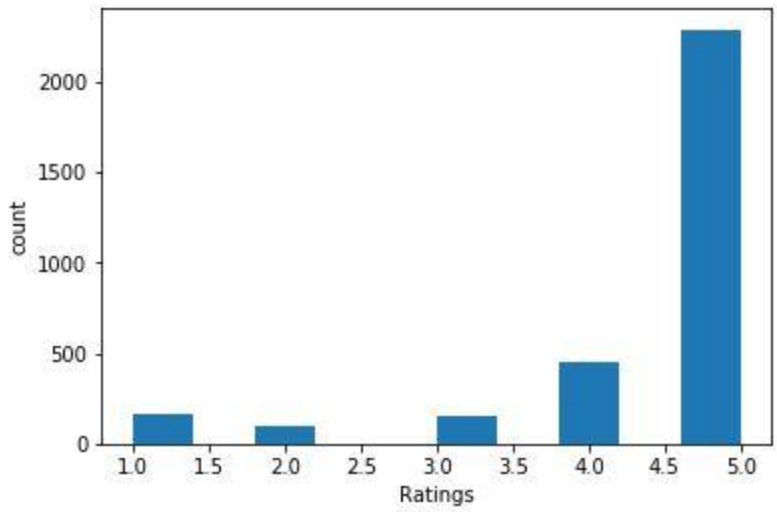

Figure 1. Distribution of Amazon Alexa product ratings (1=low; 5=high)

By plotting the data and grouping the data by rating, we see prominent left skew, or negative skewness in the data. This is corroborated by the counts, which tell us that the majority of the reviews, 2286 of 3150, are positive. However, we should note that, with this data, we don't know whether Amazon Alexa products are really that satisfactory. It is possible that the people electing to leave a review are more inclined to leave a review when it's positive. With this in mind, however, the nature of the data is unchanged—the majority of the available reviews are positive. We can still ask, for example, of the people who post positive reviews, what are the pivotal issues, product-wise? Of the people who left reviews, what are the most common topics in the text? We'll continue to explore this below.

In any case, the number of negative reviews is high enough—257 reviews with a 1 or 2 rating—that we can still look into opposing ends of the scale and explore what factors are likely to lead to either a positive or negative review. To facilitate looking at only the most helpful of words, I used stopwords to filter out proper nouns (i.e. ‘Alexa’ and ‘Amazon’), which were the most frequent but least helpful when considering things like sentiment and product components correlating with positive or negative ratings. To look more in-depth at the positive and negative

reviews, separately, I constructed word clouds to visualize the data and see which words and topics were coming up most frequently within each group. I joined all of the text together and characterized observations as positive when its rating was greater than 3 (**Figure 2**), and created the negative subset of reviews by joining the text from observations with a rating less than 3 (**Figure 3**).

Figure 2. Word cloud depicting the positive reviews (rating > 3).

The word cloud representing positive reviews confirms that good reviews frequently contain words expressing highly positive sentiment (i.e. “great”, “love”), and this is a good sanity check to confirm that ratings correlate with the types of words we would expect of a positive review. Further, the visualization also suggests that music, sound, and speaker components of Alexa products are frequently discussed in the positive reviews. We could posit that people might use the device for or at work, but it could also be that people are saying that the device just “works” or works well.

Figure 3. Word cloud depicting text from negative reviews (rating < 3).

Relatedly, the word cloud representing negative reviews does not exhibit any of the words with typically positive sentiment. In contrast, the negative cloud contains words like “really”, “even”, and “still”, words that suggest persistence or emphasis, and we can believe that these words might be used to emphasize the negative sentiment associated with those reviews. Interestingly, the same components discussed in the positive reviews—music and speaker—appear in the negative reviews. This suggests that while some of the customers enjoyed these components of their Amazon Alexa purchase—we could even argue that it was the majority of customers, as there were more positive than negative reviews—some customers had a different experience with these same features.

CountVectorizer for vectorization

Next, to analyze the textual data, I imported CountVectorizer, train_test_split and MultinomialNB from the scikit-learn library. The CountVectorizer was used for the vectorization step of the text analysis. Train_test_split was employed to create training and testing datasets from the reviews. MultinomialNB was used to construct a multinomial Naïve Bayes classifier to model the text data from the reviews. At the onset of model deployment, the default parameters

from MultinomialNB were employed. Without a-priori reasons to use other parameters, this project was intended as an exploratory approach of the different ways in which text data could be modelled. With the default parameters, the accuracy of the training dataset had a score of 0.84; the testing dataset accuracy was 0.76. Not bad. However, we know that hyperparameters can be tuned to improve model performance, and alpha is one of these parameters. The default alpha value used in MultinomialNB is 1. Instead, with alpha=0.1, the training dataset accuracy improved to 0.91 as did the accuracy for the testing dataset at 0.78. At this point, we see that while default parameters of the model provide a well-performing model, tuning the parameters and entertaining alternative settings can improve model performance with a simple keystroke.

TfidfVectorizer for vectorization

Next, term frequency-inverse document frequency (TF-IDF) was employed as an alternative approach to analyzing text data. TF-IDF is a metric of how frequent or rare a term is within a given set of text data. Namely, the math underlying the concept entails the term frequency, or the frequency of a word in a given corpus, and also the inverse-document frequency, or how rare the word is in the corpus, measured by the ratio of the word's count over the total number of words in the corpus, logarithmically scaled.

To use this approach, from scikit-learn, TfidfVectorizer was imported for use in the vectorization step. Train_test_split was used to again divide the data into training and testing datasets. Finally, MultinomialNB was retained to construct the classifier. Ultimately, all of the key steps were retained from the Naive Bayes approach, except for the vectorization step. Using the default parameters of MultinomialNB with TfidfVectorizer, the testing dataset accuracy score

was 0.71. This time, after trying out different alpha settings, alpha=0.05 led to an improved testing dataset accuracy of 0.78.

Discussion

I undertook this project to learn more about how NLP could be used to analyze text data. One of the main advantages to using automated text analysis approaches is the kind of information made available for analysis. For example, using word clouds, the size of the words in the visualization demonstrates the prominence of the word in the dataset. This is a very useful first step in exploring and understanding a set of textual datapoints, as it uses numeric, word-frequency data to visually depict the most frequent topics. One of the most important steps in any successful EDA is to better understand the data being worked with, and while word clouds are not a precise measure of any model, they are a helpful tool in understanding data visually. A person could manually read through the data to get a sense of what it contains, and this would offer yet another dimension of what the data contains—although, albeit, with a lot of time-consuming effort that is likely to skim over or skip potentially useful information. Using automated methods only, we are afforded a clear, initial view of what sorts of topics are important to Amazon Alexa customers.

Further, by subsetting the data into positive and negative reviews and ignoring the moderate or neutral reviews, we can focus in on one of the original research questions (i.e. What do positive reviewers talk about the most?). We cannot say with certainty that Amazon Alexa is a truly amazing, unflawed product, because we cannot divorce product positivity with selection bias—or the likelihood that a person writes a review when it will be positive relative to when it will be negative. For this reason, the histogram of the data is not all that helpful. However, it does tell us what kind of quantities we can expect in the different bins if we split the data into positive and

negative subsets. Other approaches to text data analysis could use more applications from sentiment analysis and delve even deeper to ask questions like, of the positive words, which were the top 10 positive words. More fine-grained analyses could be undertaken in future exploratory work.

Ultimately, both approaches to vectorization during classifier construction resulted in a fair amount of model accuracy (>75%). As latter steps demonstrated, however, tuning hyperparameters allowed for optimal customization and fit of the model to the particular dataset in question. As an exploratory approach to text data analysis, it was instructive to see the different model outputs when modifying just one of the default parameters specified in the classifier instantiation.