Naïve Bayes for Natural Language Processing of Amazon Reviews

Nyssa Bulkes
Springboard Data Science Career Track
Capstone Project #1

Paying attention to what customers think about a product is critical to business success

- Customer reviews provide feedback to companies that can be used to tailor a product and improve consumer-product opinions (and ideally, future sales)

People love to talk, tell others what they think

- While a positive review can lead to more sales; a negative review provides an opportunity to improve

Product review data can offer valuable insight into consumer-product interactions

- By leveraging reviews as textual data, companies can directly use this data to better understand how a product is being received and tailor its production, as needed

The questions:

- What's the best way to quantify textual data?
- Do we have to focus on all the words? Can we ignore some? What's the best way to model textual data?

The dataset:

- Publicly available at Kaggle
- 3150 verified product reviews (text) and corresponding ratings (1-5 scale; 5=high)

The objective:

- To utilize EDA to explore a text-based dataset
- To build a Naïve Bayes classifier to characterize textual data
- To expand on this approach by incorporating term frequency inverse-document frequency

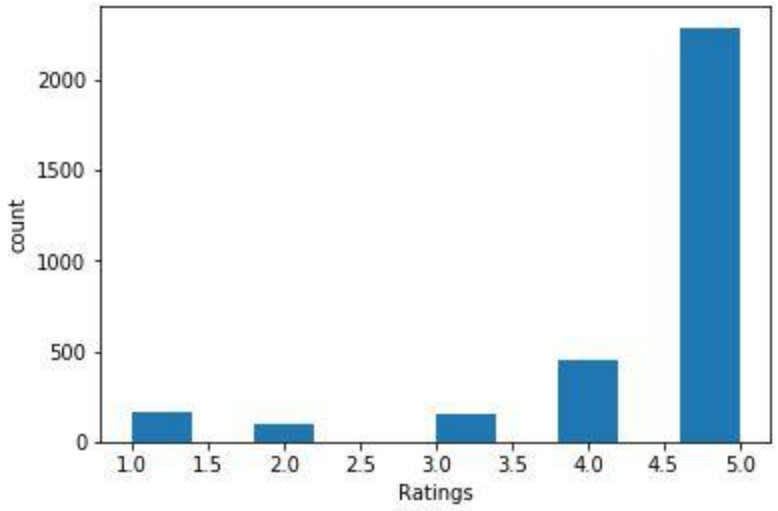
Exploratory Data Analysis

- Observations with no text in ‘verified_reviews’ dropped
- Calling the .describe() method on the dataset reveals there are 3150 unique observations
- Remaining data suggest that ‘verified_reviews’ and ‘feedback’ columns will be most helpful
- Data ranging between May 16, 2018 and July 31, 2018

	rating	date	variation	verified_reviews	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1

Finding trends

- Histogram plot of data in the ‘ratings’ column reveals most of the observations in the dataset are positive reviews
  - Is Amazon Alexa *really* that amazing? *Or*, are positive reviewers more likely to leave a review than negative or moderately-satisfied customers?



- From this data we cannot confidently say that the product is highly satisfactory, overall
- What we *can* ask, however, is, **for these mostly positive reviewers, what aspects of Alexa products impacted their opinion the most?**

– As an initial pass, a generic wordcloud shows the words that appear most often in the entire set of reviews

– This visualization, however, clearly suggests that we should remove some frequently occurring words that might not be so informative in our analysis of what makes Amazon Alexa products so appealing…

– Next, we remove frequently-occurring proper nouns (i.e. “Amazon”, “Alexa”, “Echo”, and also subset the data to include just the positive reviews (i.e. rating of “3” on 1-5 scale or greater)

– This revised cloud confirms that good reviews frequently contain words expressing highly positive sentiment (i.e. “great”, “love”)
– The visualization also suggests that music, sound, and speaker components of Alexa products are frequently discussed in the positive reviews

Naïve Bayes approach to text analysis

- Next, using scikit-learn, we'll:
  - import CountVectorizer for the vectorization step
  - import train_test_split to create training, testing datasets out of the reviews
  - Import MultinomialNB to construct a multinomial Naïve Bayes classifier to model the reviews

- Using the default parameters of MultinomialNB:
  - Training dataset accuracy score: 0.84
  - Testing dataset accuracy score: 0.76

Tuning hyperparameters

- Using alpha=0.1 instead of the default alpha=1 for classifier:
  - Training dataset accuracy score: 0.91
  - Testing dataset accuracy score; 0.78

TF-IDF approach to text analysis

- Term frequency inverse-document frequency: metric of how frequent or rare a term is within a corpus
  - Term-frequency: The word's frequency in the corpus
  - Inverse-document frequency: How rare the word is in the corpus, measured by the ratio of the word's count over the number of words in the corpus, logarithmically scaled
- For this alternate approach, using scikit-learn, we'll:
  - import TfidfVectorizer for the vectorization step
  - retain train_test_split to create training, testing datasets out of the reviews
  - retain MultinomialNB to construct a the classifier
- Using the default parameters of MultinomialNB:
  - Testing dataset accuracy score: 0.71

Tuning hyperparameters

- Using alpha=0.05 instead of the default alpha=1:
  - Testing dataset accuracy score; 0.78

Summary

- Initial EDA and visualization steps showed that we can quantify, visualize all of the words in a corpus; however, filtering out highly frequent words can clarify data insights
  - By further subsetting the data, we can more relevant answer for the question at hand (i.e. What do positive reviewers talk about the most?)

- Both approaches to classifier construction demonstrate that, while these data can be modelled with a fair amount of accuracy (>75%), tuning hyperparameters allows for optimal customization and fit of a model to the specific dataset at hand