Classification of Amazon Cell Phone Reviews

John Galpin

UMBC DATA 602

12/15/2019

Introduction

Amazon provides an extensive marketplace where all kinds of products can be found. For every product you can find a review section near the bottom of the page. This dataset is derived from those reviews. The buyers provide reviews that comes with a subject, body, and numeric rating from 1-5. Providing us with prelabeled data that can be set up for various tasks including sentiment analysis, classification or regression. Below I will be sharing my findings after exploring and performing analysis on the dataset. Then I share my methods and results for classification of the individual ratings.

Dataset Description

The dataset comes with two csv files. One contains the individual reviews from different users on each product. The other contains reference information about the product. When combined the shape of the dataset is over 80,000 rows and more 17 columns.

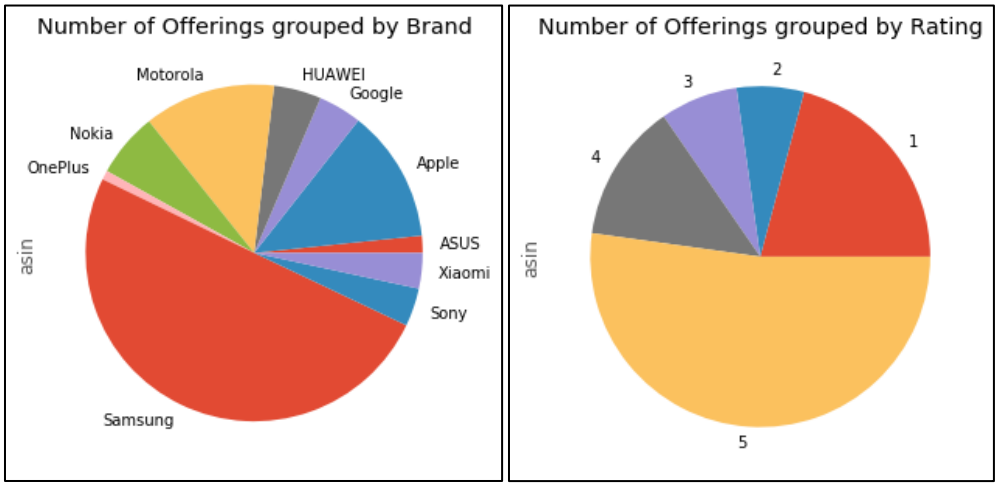
Reviews: ASIN(Product Number), Reviewer Name, Rating(1-5), date, verified review(indicator), title(subject), body, and the number of helpful votes

Items: Phone Brand, Product Title, URL, Image URL, Avg. Rating, Page URL, # of Total Reviews, and the Price

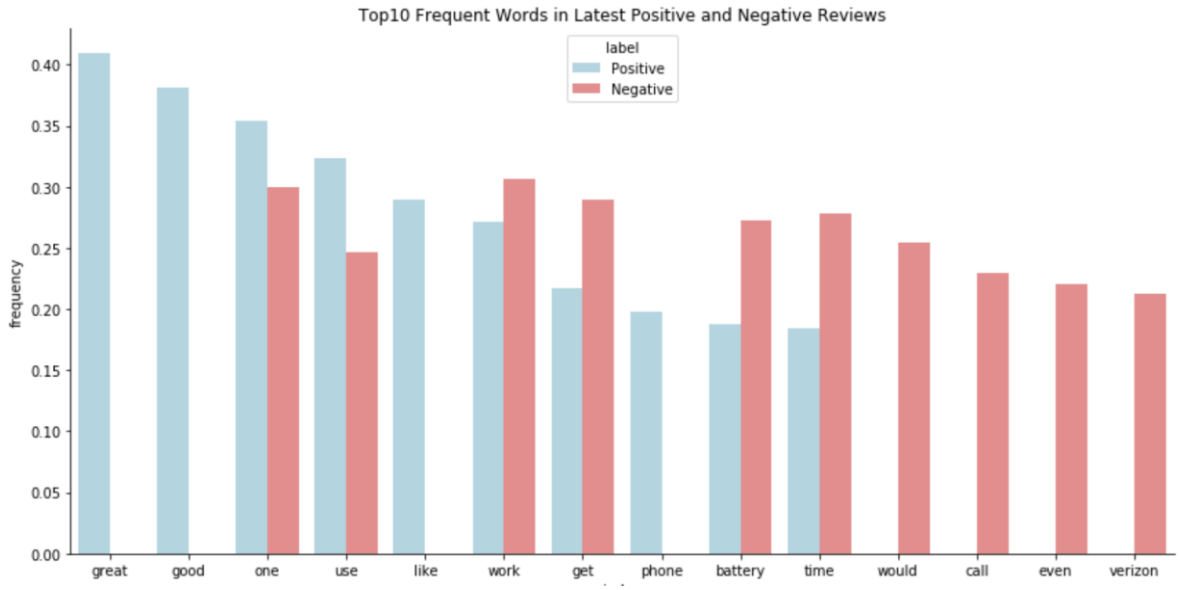
Cleaning

There are a few different things I did to clean the dataset. Most of the cleanup was done to the text portion of the data. First, I converted all of the text to lower case. Then tokenized into a list by splitting the text along the spaces. Afterward, I removed the punctuation and stop words from the text. Next, any reference of the brand name was removed from text. Lastly, I lemmatized the text. This process was done to the subject and body of each review. Note, for the sake of comparison, results for analysis on the subject are not used. Early on they proved to be valuable indicators. But I wanted to use the body review to gather the full sentiment instead of summary level. To have all of the information available in one dataset, I joined the items and reviews dataset along the ASIN number. Later, during the process I ran language detection on the dataset. While most of the dataset was in English (73,000 reviews). I wanted to stick to just English reviews and decided to filter out the other languages.

Exploratory Data Analysis

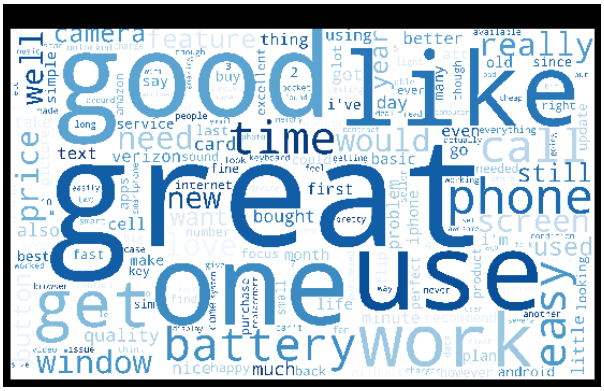


Initial finding yielded some important information for our classification task. The first being there are a disproportionate number of Samsung reviews than any other brand. Second, the rating of 5 makes up the majority of the available reviews. This is great news for cell phone makers. Bad for classification algorithms. This both skews the average rating for every brand and the weights of the classifier.

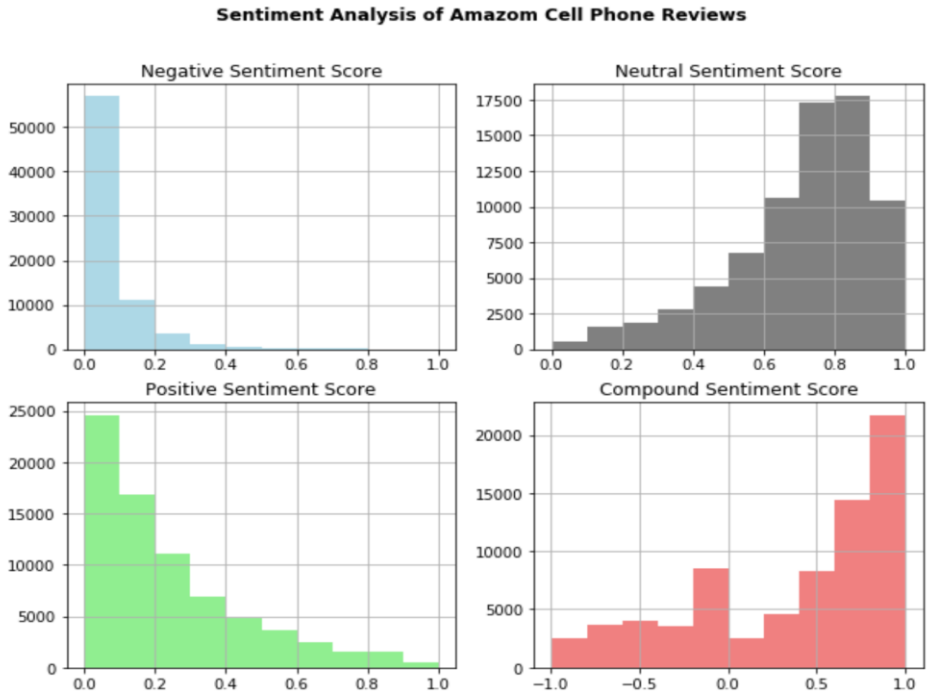


After getting some general properties of the dataset I wanted to dig a little deeper. I needed some more information to help me with the classification task. I derived a new column from ‘rating’ called ‘positivity’ which grouped the ratings into three buckets. Any rating under 3

would be -1, rating 3 would get a neutral 0, and over 3 would be assigned 1. I wanted to see what words stuck out in each of the positive and negative ends. I took a thousand reviews of both positive(blue) and negative(red). Using a word cloud, I plotted the words by frequency. Casual observation tells me the good adjectives are highly present in the positive reviews. Negative reviews results are less definitive.

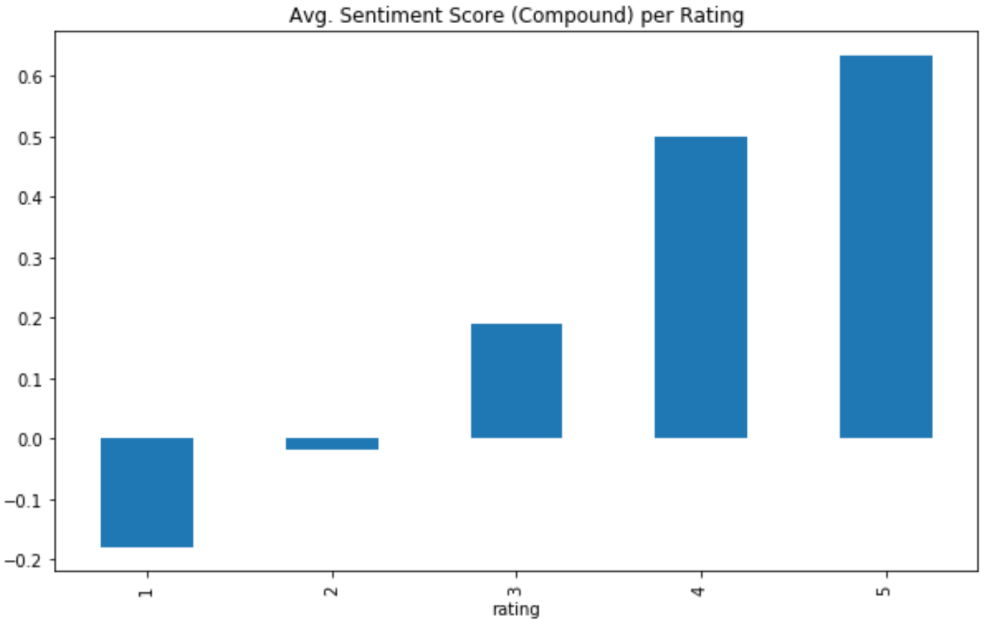


Sentiment Analysis



As part of my analysis I wanted to see how sentiment alone would factor into a classification task. To do sentiment analysis I used VADER **(**Valence Aware Dictionary and

Sentiment Reasoner). VADER is a lexicon and rule-based sentiment analysis tool. It is specifically set up to analyze sentiments expressed in social media. A particular advantage is that it incorporates commonly used slang and it does not have to be trained. VADER assigns a positive, negative and neutral score for each statement. In this case the cleaned text from the body of the review. These scores are combined into a compound score from negative one to positive one.

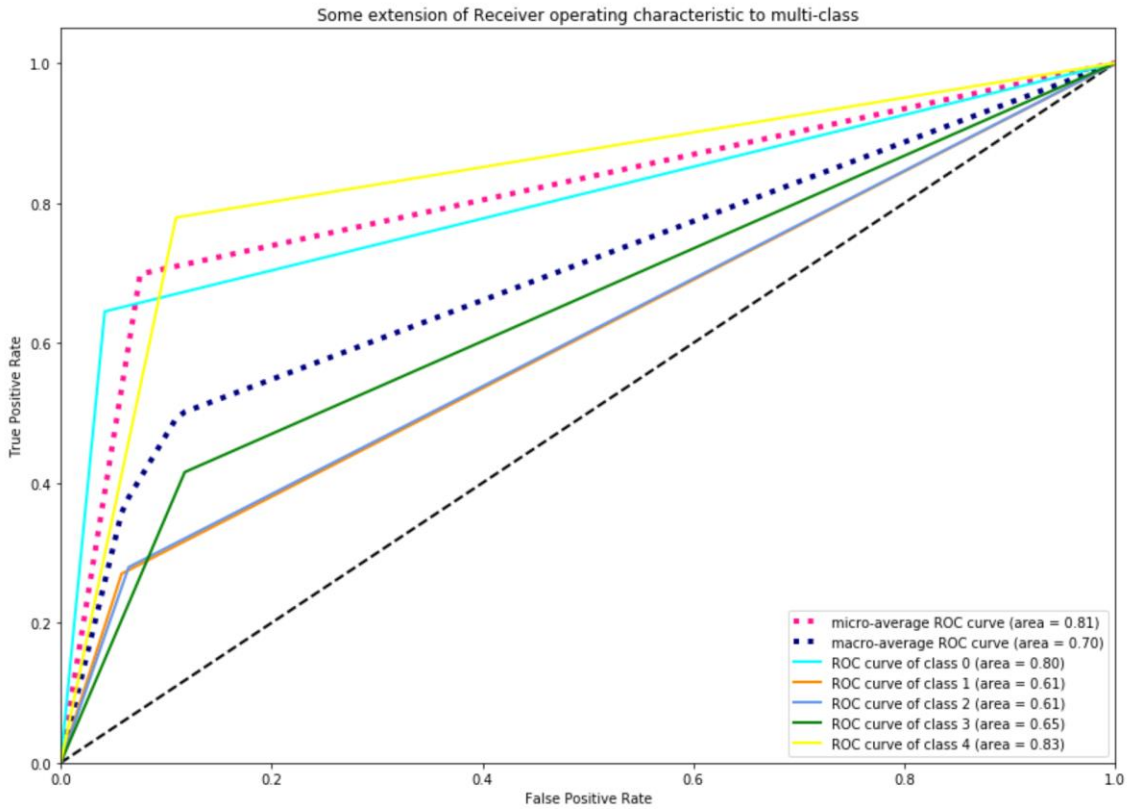


The correlation coefficient between sentiment score (compound) and rating is 0.6089. From the above graph on average there is a correlation between the compound sentiment score and each class. Next, we will test the sentiments predictive power.

Classification and Metric Analysis

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Time(Sec)**
AdaBoosted Tree(VADER)	60%	69%	60%	64%	46
SGDC Linear SVC(TF-IDF)	70%	92%	70%	79%	.66
RNN LSTM	90.91%	77.93%	63.46%	69.93%	5,531

For the classification task I wanted to pit word vectorization and sentiment analysis against each other. Both the VADER scores and TF-IDF were individually ran through the Boosted Tree and the SVC. The best score from those tests are provided in the table above. I defined three different models that I thought were best suited for the task. The AdaBoosted Tree ended up performing best on the sentiment analysis scores. For TF-IDF vectorization, the Stochastic Gradient Descent Linear SVC had the best result. Lastly, I ran the cleaned text through a LSTM Recurrent Neural Network.



LSTM RNN ROC Curve

The sentiment analysis did have some predictive power. Using the AdaBoosted Tree the prediction accuracy fell in line with the correlation score. Standalone the sentiment analysis score wasn't a strong predictor of review ratings. Word vectorization proved to be the much better setup for this task. Both SGDC and RNN performed significantly better on the classification task than the models that used the VADER scores. In the future I would use the full text provided (subject and body). I would also like to test combining sentiment analysis and word vectorization into one model.

Work Cited

Pandey, Parul. “Simplifying Sentiment Analysis Using VADER in Python (on Social Media Text).” *Medium*, Analytics Vidhya, 8 Nov. 2019, medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f.

“Text Classification with an RNN: TensorFlow Core.” *TensorFlow*, Google, 2020, www.tensorflow.org/tutorials/text/text_classification_rnn.

WenlingYao. (2019) Kaggle, Source Code. https://www.kaggle.com/yaowenling/amazon-cell-phone-review-nlp