Amazon Alexa Reviews Classification using Sentiment Analysis

Capstone Project 2
Milestone Report 1

Catherine Somers

Problem Statement

The purpose of this project is to create a model to make predictions with regards to Amazon Alexa product reviews. The main focus will be using sentiment analysis in an attempt to predict the significance to the overall reception to Amazon Alexa enabled products. We can grasp some sort of insight to the positive and negative reviews given. From a business perspective, this is useful since a more informed decision can be made for updated versions of existing products. A different but related use to this data would be to help detect which features among the dataset are important in determining what is taken as positive feedback or negative feedback. Product managers typically use product reviews as a way to gauge how they could further improve future iterations of their existing product down the line. Each new version takes one existing thing and makes it better. In addition, while this report may focus directly on Amazon Alexa reviews, the techniques used here could also apply to any type of product review dataset with a star rating system for positive and negative feedback.

Data Acquisition and Wrangling

The first step in preparing for the analysis of Amazon Alexa reviews was to search for an appropriate dataset. Since sentiment analysis is something that is important from a business standpoint. Given the nature of how voice assistants have shaped the scope of how we live our lives, I wanted to explore the ways how these products have improved the quality of life. What made me decide upon specifically Alexa enabled devices was more to do with the interest in seeing what people had to say about Amazon's Alexa over the other competing voice assistants that exist.

I was able to find data for Amazon Alexa products that provides input data, star ratings, review date, product variant, and feedback for a number of various Alexa products. I looked on Kaggle as it is one of the largest platforms to find public data and an overall great place to find data to investigate for sentiment analysis. The data I have chosen to work was last updated 2 years ago, but can still help give a snapshot of what impressions customers got in the timeframe of the reviews.

The first step in working with this data from Kaggle was to download the data file. It is a tsv file that contains 3150 values for ratings on Amazon Alexa products from the last two years. Once I had the data file downloaded, I pulled it into a notebook the same way you would read in a csv file. After I converted the tsv into a pandas dataframe, I did some initial examining to see if there were any null values. To my surprise, there were

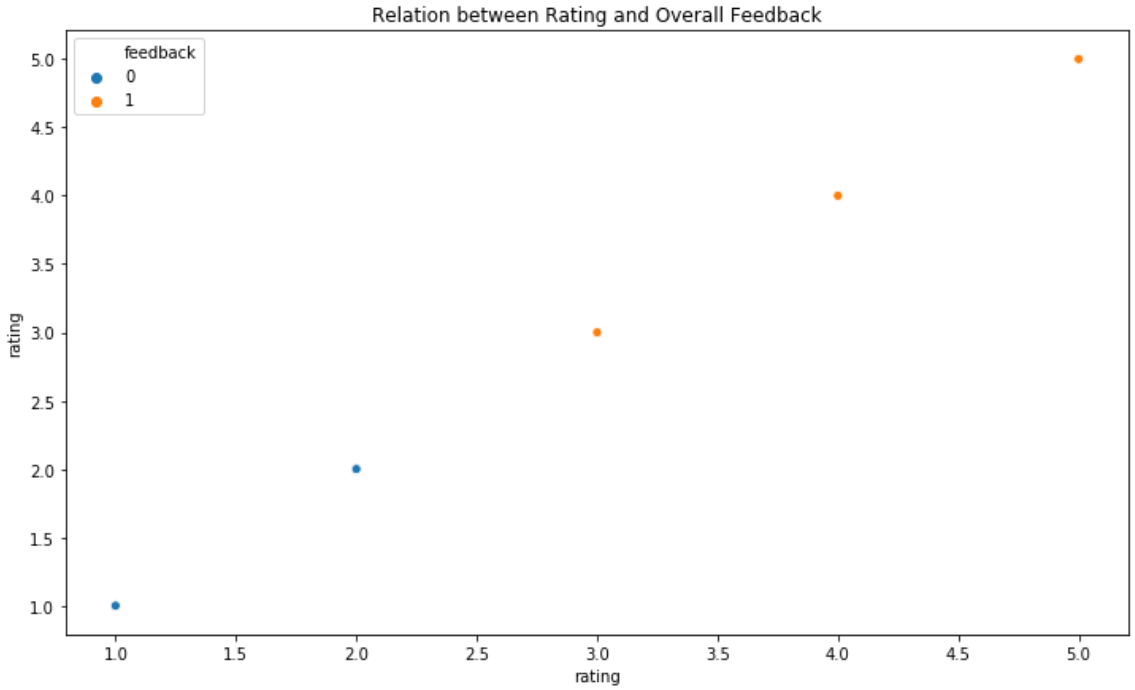
not any null values and the data acquired was fairly clean. The average rating for reviews across every product variation was 4.46. For the most part, it seems that most users have been happy with their devices.

To perform feature engineering later in the model, the year, the month, and day of the week were extracted into separate columns. Estimating the review length is also an important feature for classifying text in Natural Language Processing (NLP). As a result, a new column has been added to the data called review_length.

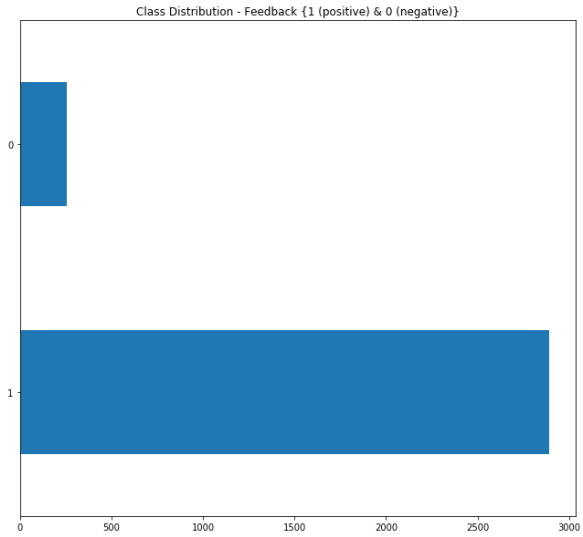
Data Exploration and Storytelling

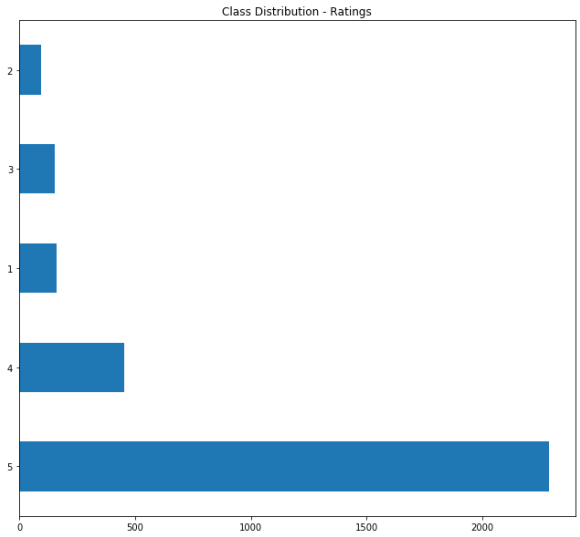
Upon completing the previous step, it was next time to explore the dataset through visualizations. I was hoping to find some patterns among the product variation and overall feedback. I also wanted to find out more about the relationship between the rating and overall feedback.

I began looking into the relationship between rating and feedback by plotting in a scatterplot comparing both. In order for feedback to be considered positive, its rating needs to be greater than 3.

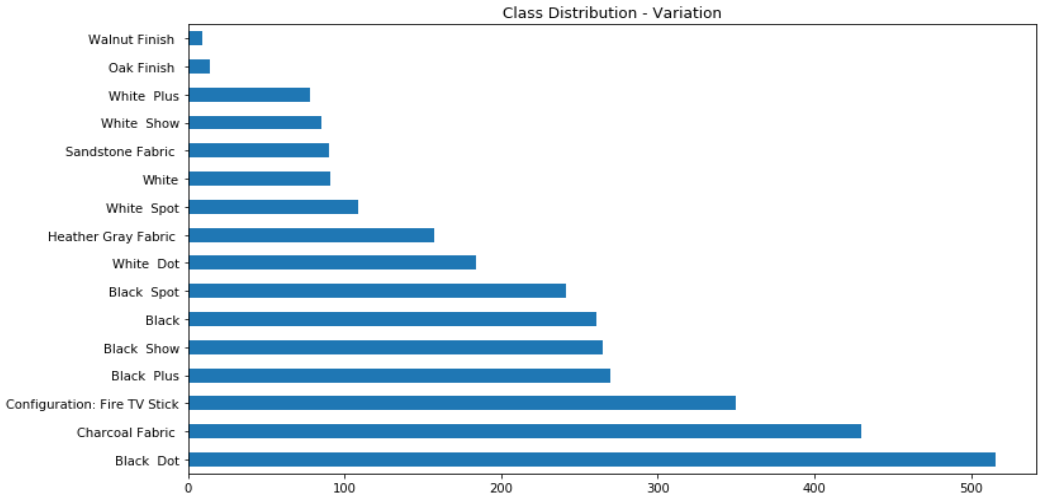


This gives more of a general overview on the correlation between rating and feedback. As we see here, the rating value range for negative feedback is 2 and below. Rating value range for positive feedback is 3 and above. I felt like this didn't give me as much insight as I would like so I wanted to look at feedback and ratings in their own respective bar plots to see how feedback and ratings are distributed in the dataset.





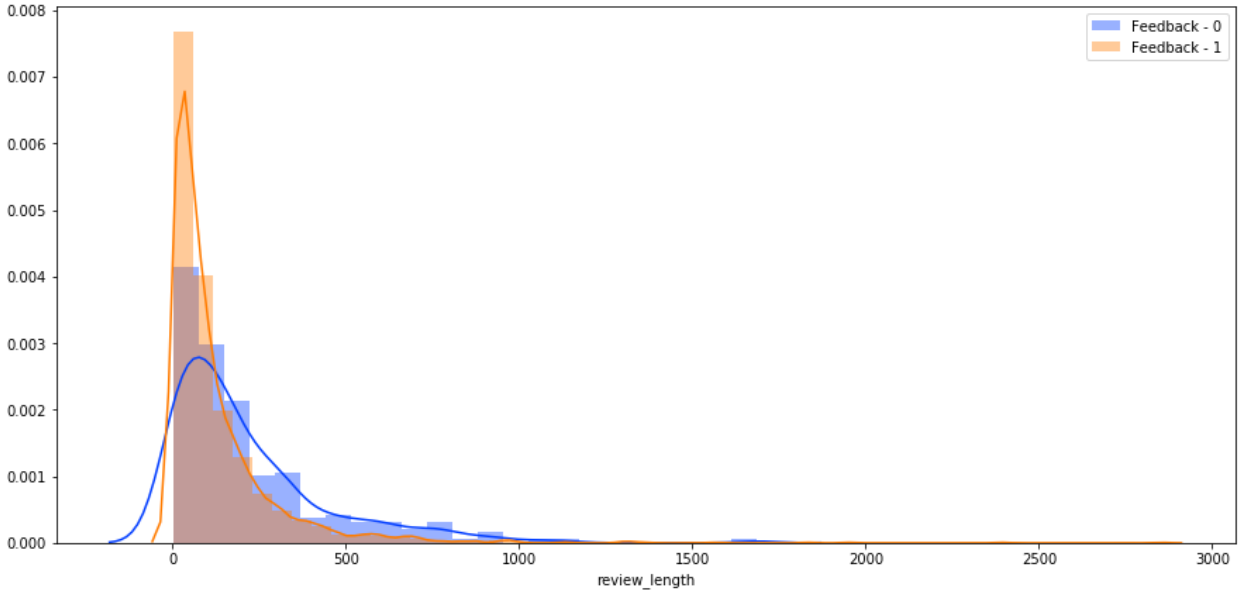
Looking at both cases here we see that the distribution among feedback and ratings is highly skewed on the positive side. For the most part, products have been well received by the customers. What may help in the future when it comes to reviewing classification models is stratifying the data to avoid class imbalance. This is something I need to keep in mind working with this kind of data and for future cases.



There are distinct bins here with each product type in the Alexa enabled line of devices show a pattern for the model preference. The Black Dot model is the most popular one out of all the different variations.

Application of Inferential Statistics

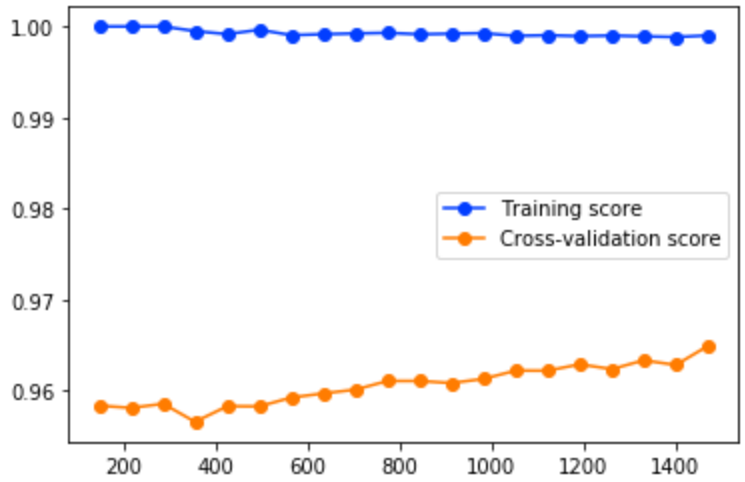
Once I explored the data visually, I began looking to perform statistical analysis on the review data. Exploring the relationship between review length and type of feedback. Looking at this plot, we see that customers with negative reviews write longer reviews.



For variation, one hot encoding was also used in my analysis. One hot encoding is a type of vector representation where all the vectors in a vector are 0, except for one, which has a 1 in its value. One important thing to note is that no matter how many dummy variables you end up is to make sure not to drop any one variable. Doing so may end up with a dummy variable trap. With that said, it means that there is the possibility of their being a case for perfect multicollinearity. This multicollinearity may occur when the independent variables in a regression model are correlated. The correlation becomes a problem due to independent variables needing to be independent. Coefficient estimates can swing based on which other independent variables are in the model. These coefficients can become very sensitive to small changes made to the model. Multicollinearity reduces the precision of the estimated coefficients. Due to this, we might not be able to trust any of the p-values to identify independent variables that are considered statistically significant.

I also performed K Fold Cross Validation with the data which gives us a good idea of how our selected model is performing on different chunks.

We decided to look at using both random forest classifier and gradient boosting classifier in our analysis here. The data was split up into a training set and a testing set first before getting started with random forest. One of the most important methods of random forest classifier in sci-kit learn is feature_importances. Using Random Forest Classifier, we looked at the top 10 features (not pictured) and created our y_pred. From there, the learning curve was looked at between the cross validation score and training score for the model.



From running the model with random forest, we got the following as the result.

Random Forest Classifier:
Accuracy Score: 0.9216931216931217
Precision Score: 0.9284940411700975
Recall Score: 0.9907514450867052
F1 Score: 0.9586129753914988
Confusion Matrix:
 [[14 66]
 [8 857]]

From running the model with gradient boosting, we got the following as a result.

Gradient Boosting Classifier:
Accuracy Score: 0.928042328042328
Precision Score: 0.9289558665231432
Recall Score: 0.9976878612716763

F1 Score: 0.9620958751393535
Confusion Matrix:
 [[14 66]
 [2 863]]

The models performed well using both methods and scored fairly high in each scoring criteria. From my analysis with the data so far, we can conclude that feature engineering is one of the most crucial steps when it comes to Natural Language Processing. Along with that switching over from a count vectorizer over to a TDF IF vectorizer made a difference in the performance of the F1 Score.

Application of Sentiment Analysis with BERT

Next Steps

Through the first stages of the project, I was able to investigate the relationship between the rating and feedback. Looking at both in two different cases seeing that both show to be highly skewed on the positive end of feedback. The next step in my analysis will be to perform sentiment analysis with BERT looking at the rating scores (1-5). First by splitting up the data into a training set and then a validation set. I will attempt to predict the sentiment of each class.