***Sentiment Prediction using Amazon Product Reviews***

DATA 690: Natural Language Processing (NLP)

Professor Dr. Tony Diana

Authors:
Demarcus Wirsing, Diego Burgos, Himanshu Ghritalhre.

## Abstract

Our main question for this project is to see if it is possible to build a robust Sentiment Analysis machine learning system to determine if the reviews are positive or negative. Using Amazon review data, we were able to deploy a machine learning system that learned from all the positive, neutral, and negative reviews, and fine tune the algorithm to avoid bias sentiment. Two approaches were done to optimize the hypothesis. The first approach was done by trying to figure out if calculating sentiment for the text based on ratings already present in the dataset and reviews would perform better. The second approach was done by using polarity functions from TextBlob to carry out or derive negative and positive sentiment for the text of the reviews and then running the model and testing its accuracy. The model that scored the highest accuracy was obtained from fine tuning the Support Vector Machine classifier using the second approach

## Introduction

Sentiment analysis is a series of methods, techniques, and tools about detecting and extracting subjective information, such as opinion and attitudes, from language. Historically, its interest is stemmed from the desire to further understand other's opinions. Sentiment analysis has had a 50-fold growth between 2005-2016. It has become one of the fastest growing research areas in computer science that will continue to grow in the future (Mantyla, 2016).

Amazon reviews can be a very important factor when trying to make the decision to purchase something. One scathing review can be enough for someone to avoid the product entirely. Obtaining the most desired qualities of a product can have a great impact for its manufacturers. This information can be used for them to address any concerns or flaws so that customers will give more favorable reviews, which can result in a sales boost. For this project we are implementing Sentiment Analysis with Amazon review data to find out what traits of a certain product influence customer reviews the most. We try to find the most optimal way to address these questions:

**Research Questions**

1. Which product categories have lower reviews or which products have higher reviews?
2. Which products should be kept, or which one can be dropped from Amazon's product roster (which ones are junk)?
3. Can we associate positive and negative sentiments with each product in Amazon's Catalog?
4. By using Sentiment analysis, can we predict scores for reviews based on certain words?

**Hypothesis**

Using Amazon review data, our group will deploy a Sentiment Analysis model that will be able to determine whether the review is positive or negative.

## Literature Review

One quality source of information when trying to solve any problem is to look at previous works. It is important to find older articles on sentiment analysis as it is a good way to see how others got to their solution and how it can be applied to our problem.

Although Ahmed Kabir's work is mostly related to the programming language R, his

methodology gave a good idea on what one needs to do to conduct sentiment analysis. Kabir's (2020) steps are isted below:

1. Data collection
2. Data Measurement
3. Data analysis
5. Data collection (Raw Data)
6. Data preprocessing
7. Data Cleaning
8. Term Document Matrix
9. Word cloud
10. Obtain score
11. Sentiment analysis
12. Visualization

The process can be a long and complex process and being able to categorize each step in the process like that makes it easier to comprehend. Our group also had some issues dealing with how to exactly approach pre-processing text. Kabir provided some good insight on how and when data preprocessing should be done.

One article that we felt was relevant to our problem was Anad Joseph Daniel's article on sentiment analysis using Amazon review data. He found that using traditional lexicon-based approaches resulted in major accuracy issues while using machine learning required labeled data. Daniel tries to find a way to work around this by proposing a hybrid sentiment analysis framework that overcomes the flaws of both approaches. His team was able to deploy a novel tunicate swarm algorithm-based feature reduction that is integrated with the proposed hybrid approach to bypass the scalability issues that come from using large feature sets, reducing the feature set size to 43% while still retaining an accuracy of 93%. This approach improves the scalability; reduces the computation time and exchanges (Daniel, 2021). Tables 1 and 2 shows Daniel's results using performance metrics such as recall, precision, F1-score, feature size and computation time. Although our approach to our problem is done without Daniel's methods, it gives great insight on how important it is to test multiple models to find the most optimal one.

Figure 1:

Table 6.4: Performance measures comparison of proposed hybrid model with TSA and without TSA on Amaz data

| Products | With TSA | | | | Without TSA | | | |
|---|---|---|---|---|---|---|---|---|
| | Accu racy | Preci sion | Recall | F1 score | Accu racy | Preci sion | Recall | F1 score |
| Furniture | 93 | 94.4 | 97.6 | 96 | 91 | 93.9 | 97.3 | 95.6 |
| Toys | 86 | 92.5 | 95.3 | 93.9 | 84 | 89.8 | 91.6 | 90.7 |
| Electronics | 89 | 93 | 96.2 | 94.6 | 88 | 89.3 | 95.1 | 92.2 |
| Camera | 85 | 87.5 | 93.1 | 90.3 | 82 | 85.4 | 92.8 | 89.1 |

Chart showing how Daniel's model performs against various products

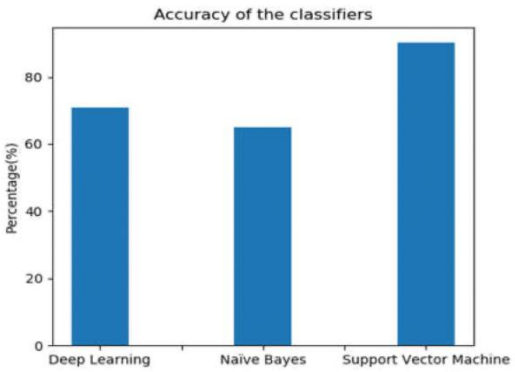
Figure 2:

| Techni ques | Performance measures | | | | | |
|---|---|---|---|---|---|---|
| | Accur acy | Preci sion | Recall | F1 score | Com- putation time | Feature size re- duction (%) |
| Proposed OBL-TSA | 95 | 95.4 | 96.6 | 96 | 15 sec | 43 |
| TSA | 93 | 94.3 | 97.6 | 96 | 22 sec | 40 |
| PSO | 89 | 91.1 | 96.5 | 93.7 | 35 sec | 38 |
| GA | 90 | 92.1 | 96.5 | 94.3 | 38 sec | 35 |

Chart comparing the results of the various techniques used by Daniel

Saurav Pradha similarly approaches this problem using Twitter data. She uses the same methods used to prepare the text for amazon review data such as stemming, lemmatization to compare the popularity of two products, Google Now and Amazon Alexa. Due to the unstructured nature of Twitter data, Support Vector Machine performed the best with 90.3% accuracy (Figure 3). Seeing how Support Vector Machine performed well with the unstructured data given, her work gave us a good case to use for Amazon review data to see how it compares to other models (Pradha, 2019).

Figure 3:




Bar plot showing which classifier performed the best in Pradha's analysis.

Classification of sentiment is usually done in two or three classes. Some Amazon ratings may have multiple classes. This can result in making the model biased and reduce the accuracy. Anirban Mukherjee (2019) used oversampling to reduce the class imbalance of the dataset before training the model. Using a variant of neural networks such as simple RNN, GRU, LSTM, and Bidirectional LSTM to find out which model performed the best. His results (Figure 4) show that the Bidirectional LSTM performed the best.

Figure 4:

| | Accuracy | | | | | |
|---|---|---|---|---|---|---|
| Dataset | Simple RNN | | GRU | | LSTM | |
| | Train | Test | Train | Test | Train | Test |
| I | 55.92% | 56.67% | 65.51% | 60.21% | 63.47% | 60.73% |
| II | 65.92% | 67.17% | 67.88% | 67.07% | 67.8% | 66.88% |

Arniban's results showing the training and testing accuracies of simple RNN, GRU, LSTM, and Bidirectional LSTM on two datasets

## Methodology

### 1.Data Exploration/Cleaning

The dataset that was used for this project was a list of over 34,000 Amazon product reviews obtained from Kaggle. Each entry contains a review score, the text of the review, and the information regarding the product itself. The dataset was found to have an average review score of 4.51 with low standard deviation. We observed that most reviews were positive from the 2nd quartile onwards. The average number of helpful reviews (reviews.numHelpful) is 0.5 but with a high standard deviation. The range of most reviews had a helpful count of 13. The review with the highest helpful count was 621. The most frequently reviewed products had an average review rating in the 4.5-4.8 range with little variance. Although there is a slight inverse relationship between the name frequency level and average review ratings for the first 4 names are rated between 4.5 – 4.8, which is considered good overall reviews.

### 2. Training/Test Data

The dataset was split into training and test sets to train a sentiment analysis classifier. Because most reviews were positive with 5-star reviews, we needed to do a stratified split on the reviews score to ensure that we don't train the classifier on imbalanced data. To fix this, we used sklearn's Stratified ShuffleSplit class to remove all samples that have NAN in review score, then converting all review scores to integers.

### 3. Sentiment Analysis/Text Preprocessing

Spelling correction is a useful pre-processing step because this also will help us in reducing multiple copies of words. For example, "Analytics" and "analytcs" will be treated as different words even if they are used in the same sense. The text was further modified with further edits by making the text lowercase, removing text in square

brackets, removing punctuation, removing words containing numbers, and getting rid of punctuation and nonsensical text.

Text is turned into numerical feature vectors using the Bag of Words strategy. This is done by assigning a fixed integer ID to each word occurrence. To implement the Bag of Words strategy, we used SciKit-Learn's CountVectorizer to perform the following:

Tokenization was performed to break sentences into words. Stopwords to filter filler words such as 'the' and 'are.' Occurrence counting was done to build a dictionary of features from integer indices with word occurrences. Feature Vector was implemented to convert the dictionary of text documents into a feature vector.

Our training sample had 22665 training samples and 9636 distinct words. Along with longer documents, we saw higher average count values on words that carry very little meaning, overshadowing shorter documents that have lower average counts with same frequencies. TfidfTransformer is implemented to reduce this redundancy. Term Frequencies are used to divide the number of occurrences for each word by total number of words. Term Frequencies Times Inverse Document Frequency (Tfidf) downscales the weights of each word, assigning less value to unimportant stop words.

## Different Models

To find the most optimal model, we deployed and tested different models to see which one performs the best. They are:

1. ***Multinomial Naive Bayes***

Multinomial Naïve Bayes is most suitable for word counts where data is typically represented as word vector counts while also ignoring non-occurrences of a feature i. It is a simplified version of Bayes Theorem, where all features are assumed conditioned independent to each other (the classifiers), P(x|y) where x is the feature and y is the classifier

2. ***Logistic Regression Classifier***

Logistic regression is a traditional machine learning model that fits a linear decision boundary between the positive and negative samples.

3. ***Support Vector Machine Classifier***

Support Vector Machines is very suitable for classification by measuring extreme values between classes, to differentiate the worst-case scenarios so that it can classify between Positive, Neutral and Negative correctly.

4. ***Random Forest***

Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, it predicts the final output.

5. ***Decision Tree Classification***

Decision tree classification is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules, and each leaf node represents the outcome.

## Visualization

To do something more we decided to create a data application using the new python library called Streamlit to show analysis in the Amazon dataset and machine learning prediction of sentiment values based on product reviews. The data application is a five hundred plus line of code in the form of a python file, this can be running on any operating system and will be presented in the user's browser. In this application there are three sections: Home, exploratory data analysis, and Machine Learning Models.

Each section can be accessed by the widget on the left-hand side of the page, you will have to click on the widget to show the drop-down list of the pages you would like to go to. The Home section will introduce you to our project as well as a look into our project's proposal, to access the proposal you will have to check the box to make it appear and uncheck to make it disappear.

Going over to the next section which is the Exploratory Data Analysis section you can see that this page is what the name entails. This page will display all data analysis from what we gathered on the data set. Just to describe the page here we are looking at the number of rows in the set, seeing how many products are in the set itself, etc. In the final section of the application, you have the Machine Learning (ML) Model. In this section, we are displaying our ML model creation in two approaches, one being we will also try to figure out if calculating sentiment for the text based on rating (already present in the dataset) and reviews is better. Using Streamlit to build this application we found to be more useful and interactive than that of using the conventional Jupyter notebook.

## Results

After doing all our analysis we were able to get these answers to our questions:

1. ***Which product categories have lower reviews or which products have higher reviews?***

   Amazon Basics AAA and AA Alkaline Batteries have the highest number of reviews and Amazon Basics Nylon CD/DVD Binder has the Lowest number of reviews.

2. ***Which products should be kept, or which one can be dropped from Amazon's product roster (which ones are junk?)***

   Based on the bar graph for Names of products, we see that certain products have significantly more reviews than other products, which may indicate a higher sale in those specific products. We also see that the products have a "right tailed" distribution which can also suggest that certain products have higher sales which can correlate to the higher product frequencies in the reviews. We also took the log of the product to normalize the data, in order display an in-depth picture of each products, and we see that the distribution still follows a "right tailed" distribution. This answer the first question that certain (products) have better sales, while other products have lower sale, and in turn dictates which products should be kept or dropped.

3. ***Can we associate positive and negative sentiments with each product in Amazon's Catalog?***

   Based on analysis from the dataset this was achievable by sub-sectioning the dataset based on the product name and then running that dataset with the filtered product name through our sentiment analysis models. From the analysis the product that had to most data return was the "AmazonBasics AAA Performance Alkaline Batteries". This product provided more positive sentiment than any other sentiment category.

4. ***By using Sentiment analysis, can we predict scores for reviews based on certain words?***

   We were able to provide the sentiment analysis along with scores for each review however we were unable to derive the sentiment analysis and score based on the exact word. What we were able to show was the frequency of the words used in each review.

## Report based on EDA and above analysis

From the analysis above in the classification report, we saw that products with lower reviews are not significant enough to predict these lower rated products are inferior. On the other hand, products that are highly rated are considered superior products, which also performs well and should continue to sell at a high level.

As a result, we need to input more data to consider the significance of lower rated products, to determine which ones should be dropped from Amazon's product roster. The good news is that despite the skewed dataset, we were still able to build a robust Sentiment Analysis machine learning system to determine if the reviews are positive or negative. This is possible as the machine learning system was able to learn from all the positive, neutral, and negative reviews, and fine tune the algorithm to avoid bias sentiment. In conclusion, although we need more data to balance out the lower rated products to consider their significance, however we were still able to successfully associate positive, neutral, and negative sentiments for each product in Amazon's Catalog.

## Approach 1 v Approach 2

| Number | Model | Accuracy |
|---|---|---|
| 1 | Multinominal Naive Bayes | 90.2 |
| 2 | Logistic Regression Classifier | 92.9 |
| 3 | Support Vector Machine Classifier | 93.9 |
| 4 | Decision Tree Classifier | 93.6 |
| 5 | Random Forest Classifier | 95.1 |
| 6 | Fine tuning the Support Vector Machine Classifier | 95.9 |

**Figure 5 – Figure of how each model performed for Approach 1**

| Number | Model | Accuracy |
|---|---|---|
| 1 | Multinominal Niave Bayes | 83.30 |
| 2 | Logistic Regression Classifier | 95.10 |
| 3 | Support Vector Machine Classifier | 97.60 |
| 4 | Decision Tree Classifier | 96.90 |
| 5 | Random Forest Classifier | 95.60 |
| 6 | Fine tuning the Support Vector Machine Classifier | 97.35 |

**Figure 6 – Figure of how each model performed for Approach 2**

There are two approaches that we had to test our hypothesis. The first is to see if calculating sentiment for the text based on ratings will have better results. The second approach is to see if calculating sentiment for the text based on the polarity function from TextBlob would have better results. Figures 5 and 6 show how each model performed for each approach. The model that scored the highest accuracy was obtained from the Support Vector Machine classifier using the second approach. The model not only predicted the correct sentiment for the amazon products with an improved accuracy of 3.7 percent compared to approach 1. Based on the analysis and comparing the two approaches we decided the use approach 2 to test our hypothesis.

## Recommendation for Future Applications

Sentiment analysis can be applied for a broad range of operations. One thing that our team possibly thought of doing was to apply sentiment analysis results to detect reviews made from bot accounts. One common issue with Amazon products is inflated review scores from bot accounts. When you look up some appliances or tools

on Amazon you will get a long list of similar looking products. Many people used Amazon reviews to decide which product is better. People are more likely trust and buy something that is rated 5 stars results compared to 1 star. Unfortunately, Many Amazon products' quality may not match up to the reviews that the product gets. These reviews are generated by bot accounts who are made to give these products perfect reviews. This is how you can see people giving a seething critique of a product that is highly rated on Amazon's website. Being able to see if a product is propped up by bots may be the difference between choosing the to be able to perform such an analysis, it would be important to utilize the best approach and model that was researched in this project. Implementing the second approach with a fine-tuned Support Vector Machine will give us the highest accuracy scores as a foundation for bot detection.

## Conclusion

In conclusion the good news is that despite the skewed dataset, we were still able to build a robust Sentiment Analysis machine learning system to determine if the reviews are positive or negative. This is possible as the machine learning system was able to learn from all the positive, neutral, and negative reviews, and fine tune the algorithm to avoid bias sentiments. Although we need more data to balance out the lower rated products to consider their significance, however we were still able to successfully associate positive, neutral, and negative sentiments for each product in Amazon's Catalog.

## References:


D., A. J. D., & M., J. M. (2021). A Novel Sentiment Analysis for Amazon Data with Tsa Based Feature Selection. *Scalable Computing: Practice & Experience*, *22*(1), 53–66. https://doi-org.proxy-bc.researchport.umd.edu/10.12694/scpe.v22i1.1839

KABIR, A. I., AHMED, K., & KARIM, R. (2020). Word Cloud and Sentiment Analysis of Amazon Earphones Reviews with R Programming Language. *Informatica Economica*, *24*(4), 55–69. https://doi-org.proxy-bc.researchport.umd.edu/10.24818/issn14531305/24.4.2020.05

Mäntylä, M. V. (2018). The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers. *Computer Science Review*, 16-32.

Mukherjee, A., Mukhopadhyay, S., Panigrahi, P. K., & Goswami, S. (2019). Utilization of Oversampling for multiclass sentiment analysis on Amazon Review Dataset. *2019 IEEE 10th International Conference on Awareness Science and Technology (ICAST), Awareness Science and Technology (ICAST), 2019 IEEE 10th International Conference On*, 1–6. https://doi-org.proxy-bc.researchport.umd.edu/10.1109/ICAwST.2019.8923260

S. Pradha, M. N. Halgamuge and N. Tran Quoc Vinh, "Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data," *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, 2019, pp. 1-8, doi: 10.1109/KSE.2019.8919368.

Sakshisaku3000. (2020, July 13). Amazon product reviews using NLP. Retrieved May 04, 2021, from https://www.kaggle.com/sakshisaku3000/amazon-product-reviews-using-nlp