Product Feature Analysis Based on Customer Reviews

Sanjeev
sasa6450@colorado.edu
CSCI 6502-001
Student ID:109799826

Swapnil Sethi
swapnil.sethi@colorado.edu
CSCI 6502-001
Student ID:110238249

Aishwarya Satwani
aishwarya.satwani@colorado.edu
CSCI 6502-001
Student ID: 109876452

Abstract

The project is focused on showcasing the NLP techniques to perform Product feature analysis on the customer reviews of Amazon products in a category. Given reviews for a product, we essentially identify its features/attributes and then extract opinion about those product features and classify them as positive or negative. We will analyze the opinion expressed for each product feature and observe the good and bad features from a customer's perspective. To extract the key features of the product we will employ Latent Dirichlet Allocation (LDA) topic modeling technique and perform sentiment analysis for each of the acquired feature.

Keywords: dataset,amazon, feature, domain,vectors,NLP, Python, Visualization.

ACM Reference Format:
Sanjeev, Swapnil Sethi, and Aishwarya Satwani. 2022. Product Feature Analysis Based on Customer Reviews. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The internet has revolutionized the way we buy products. We've seen the popularity of sentiment analysis grow exponentially, and we've seen binary classification for product reviews expand exponentially during the same span of sentiment analysis's popularity. With this project, we try to take a step further and work on developing a model that not only results in the success or failure of the product, but also how it might improve its market position. The goal is to gather customer feedback and utilize it to acquire a holistic view of the product's acceptance. We analyze the feedback at a granular level and present the business with a picture of which product features work and which don't. This will enables them to recognize their product weaknesses and develop a strategy to address them.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Related Work

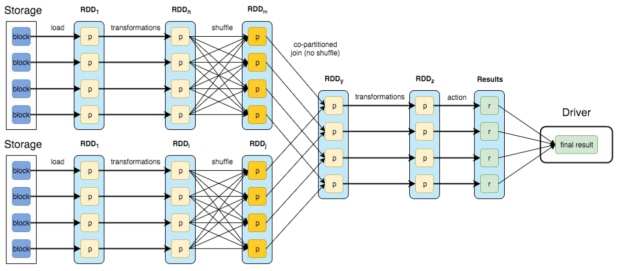
A lot of work has already been done in the area of Opinion Mining (analyzing what has been said positively or negatively). A more complete overview of opinion analysis approaches may be found in reference [9]. We are attempting to improve the Opinion Mining approach given in reference [1] by providing automatic opinion extraction using more advanced linguistic approaches.

3 Design and Implementation

The project is divided into many subtasks, each of which was analyze with the help of a few IEEE articles referenced in the reference section.

There are five stages to the project. Starting with dataset selection, infrastructure setup, extracting often discussed product attributes, identifying opinion signal words, and finally mapping product features and opinions to detect which opinion words match to which product feature. A feature vector is generated for each review at the completion of the third job, with one entry per product feature that might be larger than 0 (if the product feature has been positively remarked on) or less than 0 (if the product feature has been negatively remarked on). The fourth exercise, which was completed utilising the visualisations on the Tableau dashboard, involved analysing the recovered feature vectors. This reflects the general opinion of product features.

3.1 Data



One of Amazon's most well-known offerings is Amazon Customer Reviews (a.k.a. Product Reviews). Millions of Amazon customers have posted over a hundred million reviews to express their thoughts and describe their experiences with products on Amazon.com in the nearly three decades since the first review in 1995. As a result, Amazon Customer Reviews are a valuable source of information. We looked at other datasets on the market and landed on the Amazon customer review dataset, which includes reviews and product

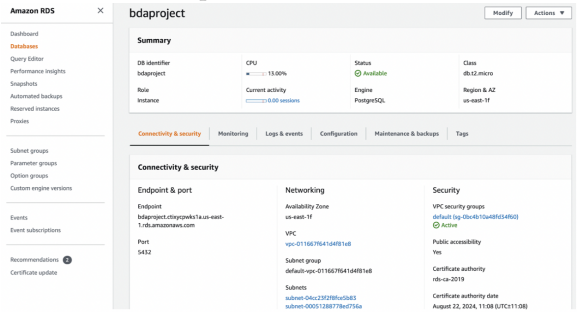
information. The dataset includes product parent, product title, product category, star rating, review headline , and review body.

	product_id	product_title	star_rating	review_headline	review_body	review_date
0	B003EM6AQE	Panasonic ErgoFit In-Ear Earbud Headphone	5	Great In-Ear Headphones!	I LOVE THESE HEADPHONES! I am totally satisfied with the sound and the fact that they stay in my ears! They are great...and I love the color of course. :)	2015-08-31

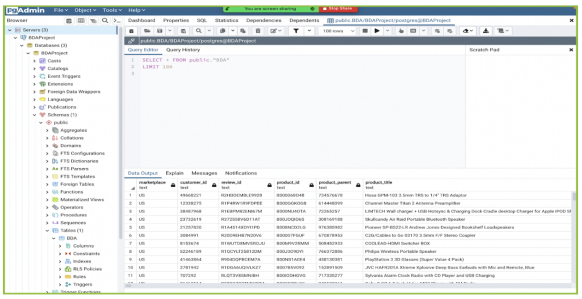
3.2 Infrastructure Setup

Considering the vast dataset for each product on Amazon, handling the data and its scalability is the tricky part. For our current project, we used headphone as our main product for reviewing and analysing. This would result in processing data of the size upto 1.8 GB.

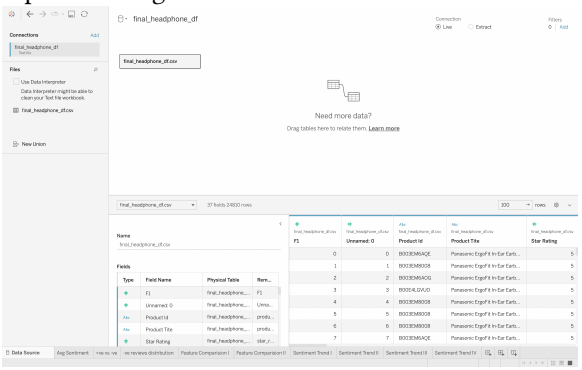
1. **AWS:**
   We setup the RDS database on AWS for the readability of the data as a table which was the successful attempt. We dealt with different dataset and we setup the project by properly setting up the endpoints and the configuration.We used the RDS databse of AWS in order to setup the dataset we have for processing the data. The following figure gives you an overview of RDS database setup in AWS.



2. **postgreSQL:** We made use of postgreSql by creating a table matching the columns according to the avaliable dataset for better view and filtering of the products. The following figure gives you the setup where we cleaned up the dataset having null values to have a proper dataset and imported it in the postgreSql tool.This made us to view the tables and analyse the data further in an easier way . The first 100 rows of the dataset chosen can be viewed here which we will be working on further for feature extraction and binary classification.

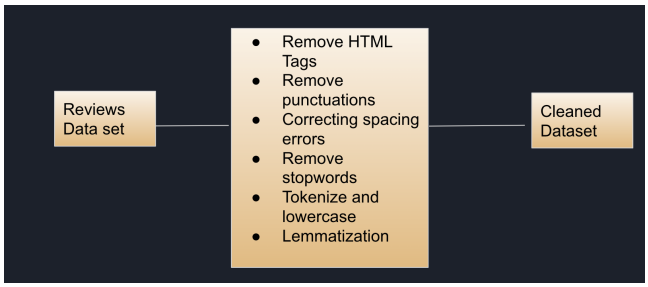


3. **Tableau :** Once the analysis was done ,we made use of Power Bi as a tool for the visualisation as shown in the image below.The usage of power BI helped us in getting different form of visualisations and we were able to filter out the parameters as required and it not only helped in visualisation but also gave us a room for improvement of our methods and approaches for topic modeling.

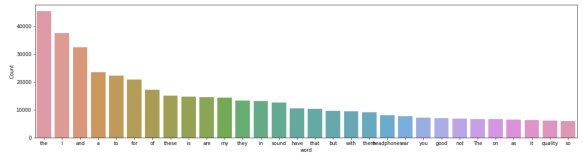


4. **Spark:**
   The goal of Spark was to create a new framework, optimized for fast iterative processing like machine learning, and interactive data analysis, while retaining the scalability, and fault tolerance.We made use of spark to load and work with such a large dataset since spark is great for consistent data and provides an interface for programming clusters with implicit data parallelism. Spark Streaming unifies disparate data processing capabilities, allowing developers to use a single framework to continually clean and aggregate data before they are pushed into data stores.

3.3 Data Wrangling(Data Cleaning)



1. **Preprocessing Text**
   The dataset before preprocessing consisted of word count equal to around 40000 and this had to be preprocessed to get a cleaned dataset which would be helpful in processing of data. Following is the image of the dataset before preprocessing.



The review text was in the most unstructured form and also various types of noise were present in it and the data is not readily analyzable without any preprocessing. The entire process of cleaning and the text standardization , making the reviews noise free and making the entire data ready for analyzing is known as text preprocessing . Below are the steps followed for text preprocessing.

a. Removing HTML Tags:
   The HTML tags do not typically add much value towards understanding and analyzing the reviews. so HTML words were removed from the text.

b. Removing Punctuations:
   Next step of text preprocessing would be text normalization and removing unnecessary and special characters.These may be special symbols or even punctuation that occurs in sentences. This step is often performed before or after tokenization.The reason for removing the punctuation is that it does not give any significance when we analyze the text and utilize it for extracting feature based on NLP

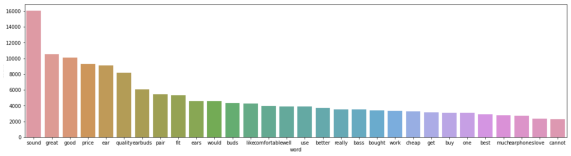
c. Correcting spacing Errors:
   Next step would of text preprocessing would be correcting the space errors in the reviews as adjusting the space would make the data analysing even more easier and understandable.

d. Removing Stopwords:
   Stopwords are those words that have little or no significance. They are removed from text during processing so as to retain words having maximum significance and context.Words like a, the,me and are those stopwords that has no significance during NLP

e. Tokenize and lowercase :
   Then the entire review is tokenized for easier understanding and the entire review is changed to lowercase so as to avoid the confusion between the words.

f. Lemmatization:
   The process of lemmatization is to get the word in the reviews to base form of the word.This is done because the root word also known as Lemma will be present in the dictionary.

The following figure shows the word count after preprocessing of data.



3.4 Topic Modeling :

Topic modeling is a technique for detecting the topics in a document/text that combines machine learning and natural language processing. It may determine the likelihood that a word or phrase belongs to a specific topic and group documents/text based on their similarity or proximity. This is accomplished by looking at the frequency of terms and phrases in the papers. Text summarization, recommender systems, spam filters, and other applications of topic modeling are among them.
Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), and Non-Negative Matrix Factorization (NMF) are some of the famous methods for extracting topic models. We're using Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA)
Steps -

1. Set up a Dirichlet distribution of documents in the topic space and select N topics from a multinomial distribution of topics over a document.

2. Set up the Dirichlet distribution of topics in the word space and choose N words from the multinomial distribution of words over topics for each of the previously sampled topics.

3. Maximize the probability of creating the same documents.

Following that, the algorithm above is mathematically defined as

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}; \alpha, \beta) = \prod_{i=1}^{M} P(\theta_j; \alpha) \prod_{i=1}^{K} P(\phi; \beta) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\phi z_{j,t}),$$

$\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ define Dirichlet distributions, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ define multinomial distributions, Z is the vector with topics of all words in all documents, W is the vector with all words in all documents, M number of documents, K number of topics and N number of words.

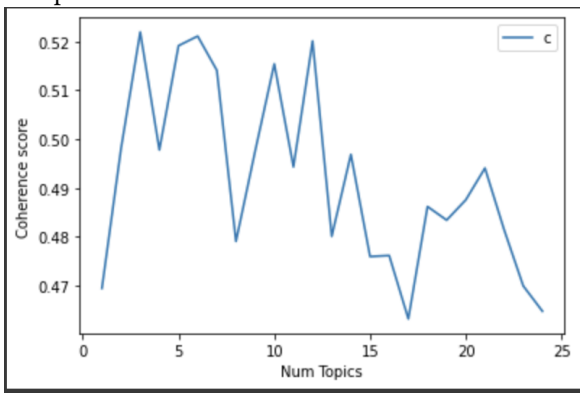
```
[(0,
  '0.141*"ear" + 0.071*"fit" + 0.055*"bud" + 0.036*"come" + '
  '0.026*"comfortable" + 0.025*"noise" + 0.021*"stay" + 0.021*"size" + '
  '0.016*"ipod" + 0.016*"wear"'),
 (1,
  '0.076*"design" + 0.020*"slightly" + 0.020*"free" + 0.018*"earpiece" + '
  '0.018*"backup" + 0.018*"length" + 0.016*"open" + 0.015*"positive" + '
  '0.015*"level" + 0.014*"mostly"'),
 (2,
  '0.056*"buy" + 0.050*"use" + 0.050*"pair" + 0.045*"well" + 0.038*"one" + '
  '0.038*"would" + 0.033*"work" + 0.028*"go" + 0.020*"earphone" + '
  '0.020*"look"'),
 (3,
  '0.213*"sound" + 0.134*"good" + 0.104*"great" + 0.100*"price" + '
  '0.093*"quality" + 0.041*"really" + 0.040*"cheap" + 0.027*"comfortable" + '
  '0.015*"amazing" + 0.011*"low"'),
 (4,
  '0.049*"last" + 0.049*"month" + 0.042*"long" + 0.041*"first" + 0.031*"new" + '
  '0.031*"take" + 0.029*"wire" + 0.026*"gym" + 0.026*"every" + 0.024*"leave"'),
```

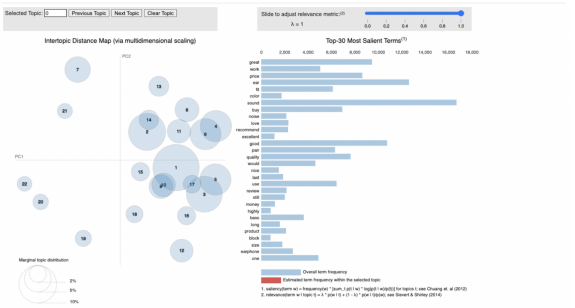
Coherence Score Coherence score indicates how human-interpretable the topics are.

CV Coherence Score CV generates content vectors for words based on their co-occurrences, then computes the score using normalised pointwise mutual information (NPMI) and cosine similarity.

Choosing the Best Coherence Score There is no single method for determining if a coherence score is high or low. The score and its value are determined by the data used to generate it. For example, a score of 0.5 may be sufficient in one scenario but insufficient in another. The only rule is that we must achieve the highest possible score.

We have used so-called elbow technique to accomplish a trade-off between the number of topics and the coherence score. The method entails plotting coherence score as a function of topic count. The idea behind this method is that we want to choose a point after which the diminishing increase of coherence score is no longer worth the additional increase of the number of topics.





3.5 Sentiment Analysis:

a. **Binary Classification**

We processed the reviews and extracted features for vectors to apply Logistic Regression along with calculating the weights. The following are the features we considered to extract the vectors.

x1	Count the number of positive lexicon	3
x2	Count the number of negative lexicon	2
x3	1 - If the word "no" is present 0 - Otherwise	1
x4	Count 1st and 2nd pronouns	3
x5	1 - If the word "!" is present 0 - Otherwise	0
x6	Log of the word count of the review	

We considered both positive and negative lexicons to count the respective features. Feature 3 is to check if the review contained the word "no" or not. Feature 4 is to count the number of pronouns and feature 5 would to check whether there exists a "!" in the review. Finally, feature 6 would take the log of the count of the words.

Logistic Regression:

Stochastic Gradient Descent was implemented using cross entropy as loss function. Keeping the initial weight as 0 and treating the bias term as additional weight by adding a dummy feature of 1 to each of the vectors. SGD randomly selects examples, predicting its values with the current set of weights using the training set to generate a loss/gradient and then updating the weights.

Preliminary Results

Splitting the dataset into training and testing sets:

```
In [52]: random.shuffle(features_array)
         train = features_array[:int(len(features_array)*0.8)]
         print(len(train))
         test = features_array[int(len(features_array)*0.8):]
         print(len(test))

         77407
         19352
```

Getting the final weights:

```
In [56]: weights
Out[56]: array([ 13.64117361,  12.74986032,   0.        ,  11.9580739 ,
                  0.        , -54.76065747, 325.50999836])
```

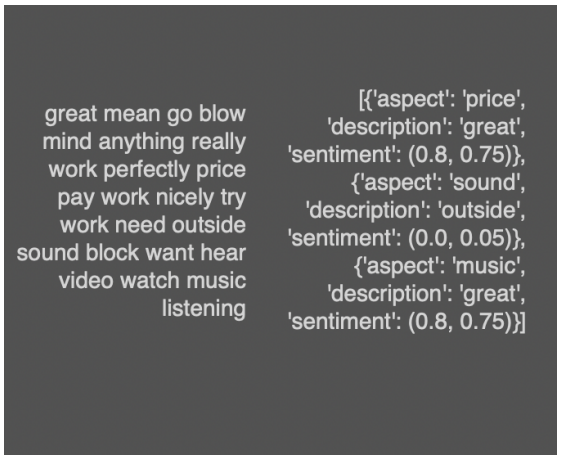
Lastly, getting the classified data:

```
In [64]: results = []
         count_pos = 0
         count_neg = 0
         for t in test:
             new_score = classprob(np.dot(weights, t[1:8]) + bias)
             if(new_score > 0.5):
                 results.append((t[0], "POS"))
                 count_pos += 1
             else:
                 results.append((t[0], "NEG"))
         print("Postive Reviews: ", count_pos, "\nNegative Reviews: ", count_neg)

         Postive Reviews:  14943
         Negative Reviews:  4409
```

b. **Aspect Analysis using TextBlob**

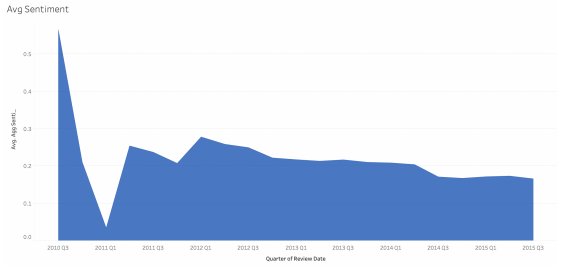
In typical sentiment analysis processes, we see a set of sentences or a paragraph being categorized as positive or negative that gain insights from an overall perspective. For our use-case however, we wanted our sentiment analysis to be aspect-wise, or in other words topic-based. Here we would find key-words that best describe our product features and perform opinion analysis that would evaluate the customer satisfaction for it. We aim to look for topics in a customer review that have some sentiment attached and extract insights on how to improve. The answer to this was to firstly employ parts-of-speech tagging (using a python library called Spacy). For each token inside our sentences, we can see the dependency thanks to spacy's dependency parsing and the POS (Part-Of-Speech) tags. We're also paying attention to the child tokens, so that we're able to pick up intensifiers such as "very", "quite", and more. This is important to correctly understand the adaptability of that particular feature under analysis. If we were to look at the sentence "These headphones have great sound quality". Then post topic modeling, we know that "sound" is one of the product features. When we apply parts of speech tagging, we can get "great" to be the adjective or rather the description of the feature "sound". Now, all that is left is to perform sentiment analysis on the description, for which we have picked the TextBlob library. TextBlob is a library that offers sentiment analysis out of the box. It has a bag-of-words approach, meaning that it has a list of words such as "good", "bad", and "great" that have a sentiment score attached to them. It is also able to pick up modifiers (such as "not") and intensifiers (such as "very") that affect the sentiment score. The results were quite impressive. Although, we did see a few drawbacks. The potential issue is that some descriptive terms or adjectives can be positive in some cases and negative in others, depending on the word they're describing. The default algorithm used by TextBlob is not able to know that cold weather can be neutral, cold food can be negative while a cold drink can be positive. We also realized, creating a vast set of labeled data would help us create more advanced classifiers with a higher amount of accuracy.

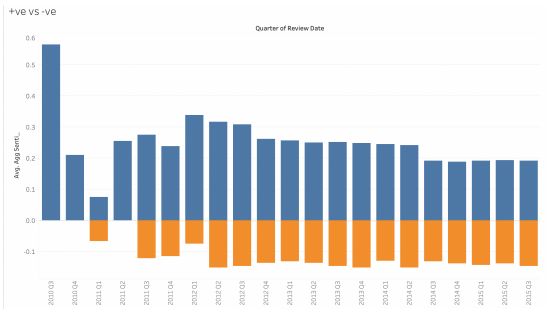


3.6 Visualisation :

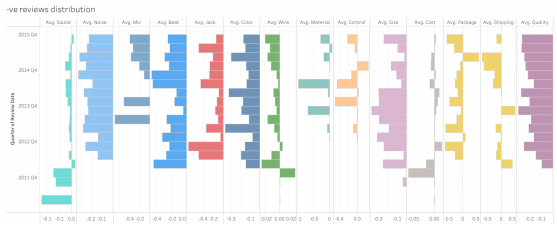
The results accumulated from the processed model are then analyzed from a business and customer perspective using BI tool Tableau. The findings are demonstrated in the form of charts and graphs:

- We first created visualizations to get a general overview of the product through its reviews. The following plots indicate the evolution of the products over the years across 4 quarters.

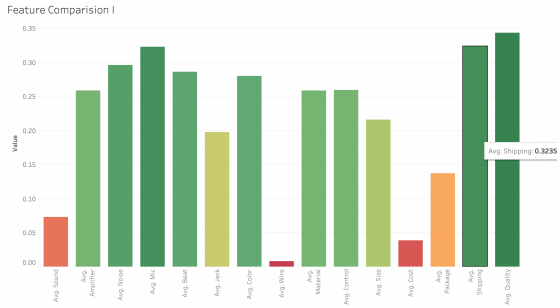




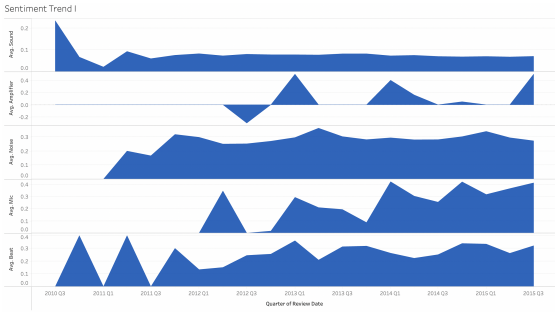
- We then went ahead with a more detailed view of our model. The following graph is a filtered set of reviews whose average sentiment analysis across different features is negative in nature. This is further categorized into average scores in different quarters over the years for each of the product features.

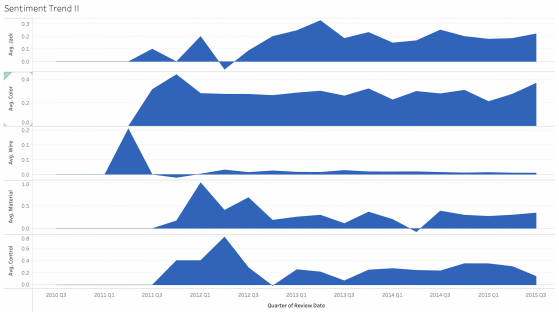


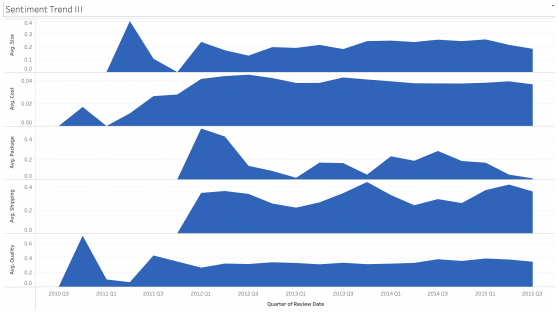
- Now coming to the graphs that would be more helpful from the business perspective in our opinion. These plots suggest the degree of improvement each of these product features could use. For example, the customers rated the cost and jack of the headphones as the bottom two aspects of the product.

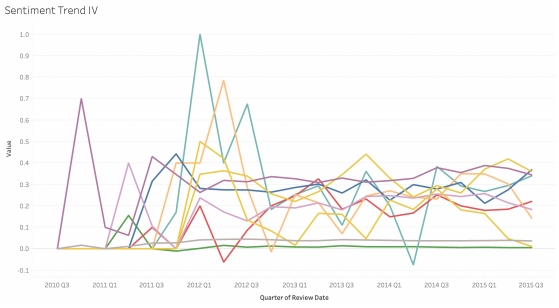


- Now the final visualizations we wanted to cover were the trends. When it comes to product development, it is crucial to pay attention to the trend of the product over a period of time. This gives the business owners a bigger picture and calls for reasoning for certain patterns observed. Following plots would be a helping hand in this situation. Since we have considered 15 product features, we have split these plots into 3 with trends for 5 features each.



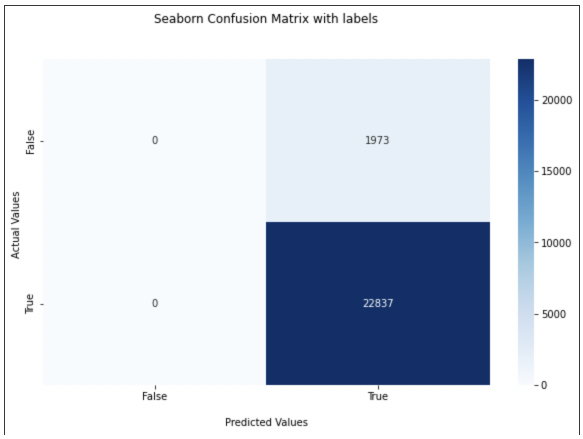






4 Evaluation

We contrasted the aggregate Aspect sentiment with the available Ratings data for evaluation. Ratings of three and up were deemed positive, while ratings of one to three were considered bad. First, We aggregated sentiments for similar features like Cost, Amount, Price in one feature Cost. Secondly, for comparison we calculated aggregate sentiment for all feature. For the analysis, we created a Confusion matrix. We were able to obtain a 92% accuracy rate.




```
Precision: 0.920
Recall: 1.000
Accuracy: 0.920
F1 Score: 0.959
```

5 Conclusion:

In this project, the amazon customer dataset was picked and the reviews were analysed and a category of Headphone was selected for further analysis and out of the reviews for the headphone , topic modeling was performed in order to extract few salient features the customers are talking about the headphone and a decision was made on picking up the right topics for the sentiment analysis. The analysis helped in giving the business the right features which has to be improved in that particular headphone and also giving them a feedback on what features are performing better in that particular brand.

6 Future work :

a. The analysis can be done using different methods in order to minimize the effect of the matching words.
b. The feature extraction can be done by overcoming dataset limitations and try to do the analysis on real data.
c. The topic modeling can be more independent and avoid manual feature extraction.
d. The NLP model can be evaluated against other existing solutions in the market.
e. Real-time review analysis. We hope to create a model that would perform analysis dynamically when a user posts a review. This would give the business and customers real-time evaluations.

7 Team Contributions:

Sanjeev:

a. Performed POCs on various platforms for setting up the infrastructure.
b. Final setup of the infrastructure for the Project
c. Researched on different available datasets for the analysis
d. Performed data wrangling(Data cleaning) of the dataset for creating a processed dataset ready for the models.
e. Manual feature extraction after the topic modeling.

Swapnil :

a. Researched on different available datasets for the analysis.
b. Implemented Topic modeling to extract top product features that were to be analyzed.
c. Created Visualization charts to depict the results and trends of the product feature sentiments.

Aishwarya :

a. Researched on different available datasets for the analysis
b. Added Spark as we were working with a large dataset.
c. Implemented NLP models on the reviews i.e. Binary classification and then Topic-based sentiment analysis for product feature analysis.

8 References

1. https://ieeexplore-ieee-org.colorado.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=5333919
2. https://ieeexplore-ieee-org.colorado.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=8713212
3. https://ieeexplore-ieee-org.colorado.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=9325646
4. https://ieeexplore-ieee-org.colorado.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=8389573
5. https://ieeexplore-ieee-org.colorado.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=9604613

6. https://arxiv.org/pdf/2005.11313.pdf
7. https://ieeexplore.ieee.org/document/9220511
8. https://blog.k2datascience.com/batch-processing-apache-spark-a67016008167
9. https://www.cs.cornell.edu/home/llee/omsa/omsa.pdf https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270 https://www.sciencedirect.com/topics/computer-science/logistic-regression http://curtis.ml.cmu.edu/w/courses/index.php/Pang_et_al_EMNLP_2002

9 Appendix:

Honor Code:

On my honor, as a University of Colorado Boulder student I have neither given nor received unauthorized assistance.