Volgenau School of Engineering

Sentiment Analysis on Amazon Electronics Product Review: Final Report

Authors: Madhuri Ghadiyaram, Vineel Vishwanth Busi
Course Project Professor: Dr. Bin Duan
Course Name: Big Data Essentials
Course name and Section #: AIT 614 – DL1

Abstract

Sentiment Analysis, also known as Opinion Mining, is the systematic identification, extraction, quantification, and study of affective states and subjective knowledge using natural language processing and text analysis. Sentiment analysis is commonly used in marketing, customer service, and clinical medicine to analyze ratings and survey responses, as well as web and social media and healthcare materials.

Sentiment Analysis of product-based feedback is the aim of this project. The data for this project was gathered from "amazon.com" online product reviews. We intend to categorize analysis data at the review stage, with promising results.

Customers can order a range of electronic goods from Amazon and have them delivered to their homes. Buyers will check and rate the product quality they have ordered using the e-commerce platform. These reviews are a valuable feature for test research because they educate product providers about consumer perceptions of the product. Customers strongly rely on product feedback today, and they are the driving force behind the sales of numerous products and services.

We performed data preprocessing, stemming, lemmatization, TF-IDF and bag of words for this project. We have used sentiment analysis using logistic regression and decision trees on Amazon electronic review data generated by customers for a variety of electronic products. Sentiment analysis of reviews can aid product suppliers in obtaining knowledge about a product based on consumer feedback. The binary sentiment analysis is unable to distinguish between positive and negative terms in customer feedback.

Keywords: Sentimental Analysis, Product Reviews, Natural Language Processing, Binary Sentiment Analysis Model

1. Introduction

Sentiment analysis is a text classifier that analyzes and categorizes texts according to the user's preferences: positive, negative, or neutral. Text mining and statistical disciplines have recently become very interested in sentiment analysis of product reviews. Both consumers and product suppliers are finding e-commerce to be extremely demanding these days. More and more people are buying items online and reading reviews to determine the product quality or to get a sense of the product before purchasing. Companies must also understand how their commodity is perceived by the general public. This can aid in the improvement of product quality in favor of targeted consumers as well as the development of marketing strategies. Sentiment analysis is at the core of modern buying analysis for these purposes.

The aim of this project is to evaluate feedback over time in order to perform sentiment analysis on them. This research is based on the initial product review, which can be found at https://jmcauley.ucsd.edu/data/amazon. The Amazon product reviews dataset collection contains millions of product reviews from a variety of product categories, including food, automobile, books, clothes, electronics, mobile phones, and other products. However, we will concentrate our research on reviews of electronic products, especially speakers. We'll sort the reviews into positive and negative categories, and then enhance our study.

2. Objectives

The project's main aim is to derive sentiment from user reviews on Amazon's electronic products. The aim is to use Natural Language Processing techniques to classify product reviews into negative and positive categories and assess how many reviews are correctly labeled by the proposed model. Furthermore, the most reviewed items will be identified, as well as the associations between features and the goal attribute.

The following are the project's overall mission procedures for achieving the study's goal:

1. Using different visualization techniques, analyze and explore the data.
2. Using Natural Languages to preprocess the data Tokenization, transforming uppercase to lowercase, stopword elimination, punctuation mark removal, stemming, and lemmatization are examples of processing techniques.
3. Expanding contractions and removing accents.
4. Execution To find word density, feature engineering uses natural language processing techniques such as bag-of-words, TF-IDF, Ngram, and other methods.
5. Three models are used to build the model.
6. Sentiment analysis, which involves identification of named entities, word similarity, and other factors.
7. Positive and negative feedback are categorized.
8. For each model, a score prediction is made.
9. Compare and contrast the models produced with accuracy, precision, and precision.

3. Dataset Selection:

In this Project we used two datasets

1. Product Reviews complete dataset:

The dataset we used is gathered from - https://jmcauley.ucsd.edu/data/amazon/ . The original data is in Json format. The electronics dataset consists of review and production information. Reviews (ratings, text, and helpfulness votes), product metadata descriptions, category details, price, brand, and image features), and links (also viewed/also purchased graphs) are all included in this dataset. The sample dataset is as below –

```
review_df.head()
```

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	AO94DHGC771SJ	0528881469	amazdnu	[0, 0]	We got this GPS for my husband who is an (OTR)...	5	Gotta have GPS!	1370131200	06 2, 2013
1	AMO214LNFCEI4	0528881469	Amazon Customer	[12, 15]	I'm a professional OTR truck driver, and I bou...	1	Very Disappointed	1290643200	11 25, 2010
2	A3N7T0DY83Y4IG	0528881469	C. A. Freeman	[43, 45]	Well, what can I say. I've had this unit in m...	3	1st impression	1283990400	09 9, 2010
3	A1H8PY3QHMQQA0	0528881469	Dave M. Shaw "mack dave"	[9, 10]	Not going to write a long review, even thought...	2	Great grafics, POOR GPS	1290556800	11 24, 2010
4	A24EV6RXELQZ63	0528881469	Wayne Smith	[0, 0]	I've had mine for a year and here's what we go...	1	Major issues, only excuses for support	1317254400	09 29, 2011

The dataset consists of 1,689,188 reviews and 17 variables.

Dataset Description:

The dataset has 1,689,188 reviews submitted by the users since years. There are also text reviews and many ratings.

Attributes	Description	Examples	Type of Variable
reviewerID	**ID of the reviewer**	AO94DHGC771SJ, AMO214LNFCEI4, A3N7T0DY83Y4IG	Object
asin	**ID of the product**	528881469	Object
reviewerName	**name of the reviewer**	amazdnu, Amazon Customer, C. A. Freeman	Object
helpful	**helpfulness rating of the review**	[0, 0], [12, 15], [43, 45]	Object
reviewText	**text of the review**	We got this GPS for my husband who is an (OTR)...	Object
overall	**rating of the product between 1 and 5**	5,1,3,2	Float64
summary	**summary of the review**	Very Disappointed, Major issues, only excuses for support	Object
unixReviewTime	**time of the review (unix time)**	1370131200, 1290643200, 1283990400, 1290556800, 1317254400	int64
reviewTime	**time of the review (raw)**	06 2, 2013, 11 25, 2010, 09 9, 2010, 11 24, 2010, 09 29, 2011	Object

2. Product Metadata:

The dataset we used is gathered from - https://jmcauley.ucsd.edu/data/amazon/ . The original data is in Json format. This dataset includes electronics product metadata such as descriptions, category information, price, brand, and image features. The sample dataset is as below –

```
print ("Total data:", str(dfmeta.shape))
dfmeta.head()
```

```
Total data: (498196, 9)
```

	asin	imUrl	description	categories	title	price	salesRank	related	brand
0	0132793040	http://ecx.images-amazon.com/images/I/31JIPhp%...	The Kelby Training DVD Mastering Blend Modes i...	[[Electronics, Computers & Accessories, Cables...	Kelby Training DVD: Mastering Blend Modes in A...	NaN	NaN	NaN	NaN
1	0321732944	http://ecx.images-amazon.com/images/I/31uogm6Y...	NaN	[[Electronics, Computers & Accessories, Cables...	Kelby Training DVD: Adobe Photoshop CS5 Crash ...	NaN	NaN	NaN	NaN
2	0439886341	http://ecx.images-amazon.com/images/I/51k0qa8f...	Digital Organizer and Messenger	[[Electronics, Computers & Accessories, PDAs, ...	Digital Organizer and Messenger	8.15	{'Electronics': 144944}	{'also_viewed': ['0545016266', 'B009ECM8QY', '...	NaN
3	0511189877	http://ecx.images-amazon.com/images/I/41HaAhbv...	The CLIKR-5 UR5U-8780L remote control is desig...	[[Electronics, Accessories & Supplies, Audio &...	CLIKR-5 Time Warner Cable Remote Control UR5U-...	23.36	NaN	{'also_viewed': ['B001KC08A4', 'B00KUL8O0W', '...	NaN
4	0528881469	http://ecx.images-amazon.com/images/I/51FnRkJq...	Like its award-winning predecessor, the Intell...	[[Electronics, GPS & Navigation, Vehicle GPS, ...	Rand McNally 528881469 7-inch Intelliroute TND...	299.99	NaN	{'also_viewed': ['B006ZOI9OY', 'B00C7FKT2A', '...	NaN

The dataset consists of 498196 category information and 9 variables.

Attributes	Description	Examples	Type of Variable
asin	**ID of the product**	0132793040	Object
title	**Name of the product**	Kelby Training DVD: Mastering Blend Modes in A...	Object
Price	**Price in US dollar**	23.36	Object
imURL	**Url of the product image**	http://ecx.images-amazon.com/images/I/31JIPhp%..	Object
related	**related products**	We got this GPS for my husband who is an (OTR)...	Object
salesRank	**sales rank information**	{“electronics”:144944}	Float64
brand	**name of the brand**	RCA	Object
categories	**list of categories the product belongs to**	[[Electronics, Computers & Accessories, Laptop..	object

4. Data Preprocessing:

1. Performed data wrangling with Electronics product reviews and meta datasets in json files which were saved in different dataframes and then the two dataframes were merged together using left join with “asin” as common column. Final merged data frame is shown below:

```
product_reviews.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1689188 entries, 0 to 1689187
Data columns (total 17 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   reviewerID      1689188 non-null  object
 1   asin            1689188 non-null  object
 2   reviewerName    1664458 non-null  object
 3   helpful         1689188 non-null  object
 4   reviewText      1689188 non-null  object
 5   Rating          1689188 non-null  int64
 6   summary         1689188 non-null  object
 7   unixReviewTime  1689188 non-null  int64
 8   reviewTime      1689188 non-null  object
 9   imUrl           1687975 non-null  object
 10  description     1655511 non-null  object
 11  categories      1689188 non-null  object
 12  title           1643686 non-null  object
 13  price           1639882 non-null  float64
 14  salesRank       810070 non-null   object
 15  related         1662142 non-null  object
 16  brand           954251 non-null   object
dtypes: float64(1), int64(2), object(14)
memory usage: 232.0+ MB
```

2.We searched for missing values as the first stage of simple data preprocessing. We discovered that the title - 45502, reviewerName - 24730, description - 33677, price - 49306, related - 27046 are having more missing values.

```
##########################################
## CHECKING FOR MISSING VALUES
##########################################

product_reviews.isnull().sum()
```

```
reviewerID             0
asin                   0
reviewerName       24730
helpful                0
reviewText             0
Rating                 0
summary                0
unixReviewTime         0
reviewTime             0
imUrl               1213
description        33677
categories             0
title              45502
price              49306
salesRank         879118
related            27046
brand             734937
dtype: int64
```

To solve the missing values in brand we extracted the first word from the title column and replaced null values with brand name. Dropped missing values 'title', 'reviewerName', 'description', 'price','related' and 'salesRank'. The final data set consists of

```
product_reviews.isnull().sum()

reviewerID        0
asin              0
reviewerName      0
helpful           0
reviewText        0
Rating            0
summary           0
unixReviewTime    0
reviewTime        0
imUrl             0
description       0
categories        0
title             0
price             0
salesRank         0
related           0
brand             0
dtype: int64
```

The final dataset consists of 75564 rows and 17 columns.

3.In order to reduce time consumption for running models, only “speaker products” were chosen, and the following method was adopted.

1. Dataset with product title named “Speakers”, “speakers”,” speaker”,” Speakers” were extracted from merged dataframe. Final speakers’ dataset was 26341 rows and 16 columns.
2. We concatenated variables ‘reviewText’ and ‘summary’ and renamed the resulted variables as ‘review_text’.
3. We checked for are any duplicate rows in the dataset, we found two duplicate rows and We dropped the duplicate variables ‘reviewerName’ and 'unixReviewTime'.
4. The variable ‘Rating’ vales has been handled in a way as
    a. If value is greater than 3, we considered it as ‘Good’. There are 24181 records.
    b. If value is less than 3, we considered it as ‘Bad’. There are 2159 records.

5. The columns asin is renamed as product_id , imurl as url, product_title as title, brand as brand_name.

6. A new column ‘rating_class’ has been included and values are inserted as per the ‘Rating’ score.

Descriptive statistics:

i. Number of reviews: 26340
ii. Number of unique reviewers: 21517
iii. Prop of unique reviewers: 0.817
iv. Number of unique products: 816
v. Prop of unique products: 0.031
vi. Average rating score: 4.329

Columns were renamed for clarity purpose.

The Dataset is as below –

```
product_reviews_sp2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26340 entries, 1420 to 1689182
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   reviewer_id       26340 non-null  object
 1   product_id        26340 non-null  object
 2   reviewer_name     26340 non-null  object
 3   helpful           26340 non-null  object
 4   rating            26340 non-null  int64
 5   unix_review_time  26340 non-null  int64
 6   reviewTime        26340 non-null  object
 7   url               26340 non-null  object
 8   description       26340 non-null  object
 9   categories        26340 non-null  object
 10  product_title     26340 non-null  object
 11  price             26340 non-null  float64
 12  salesRank         26340 non-null  object
 13  related           26340 non-null  object
 14  brand_name        26340 non-null  object
 15  review_text       26340 non-null  object
 16  rating_class      26340 non-null  object
dtypes: float64(1), int64(2), object(14)
memory usage: 3.6+ MB
```

7. Dropped unnecessary columns 'rating', 'reviewer_name', 'salesRank', 'unix_review_time', 'reviewTime', 'url', 'description', 'categories', 'product_title', 'price', 'related', 'brand_name', 'rating_class_num', 'HelpfulnessNumerator', 'HelpfulnessDenominator' and 'review_length' in the dataset

8. The final dataset consists of 26340 rows and 7 columns.

```
df4.info()
```

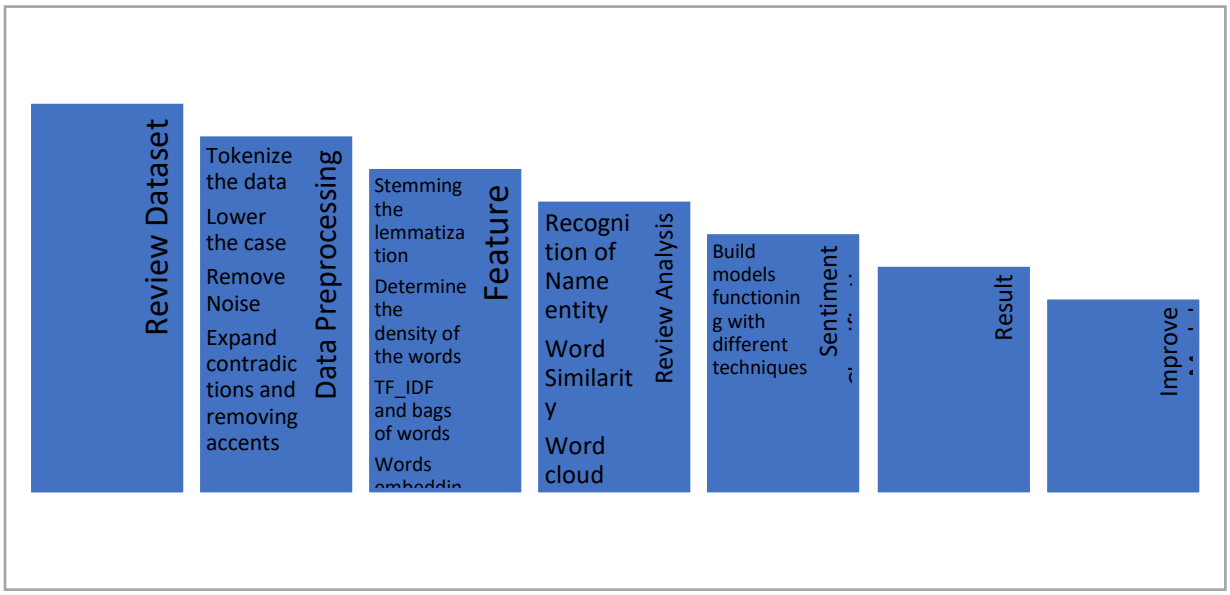
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26340 entries, 1420 to 1689182
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   reviewer_id   26340 non-null  object
 1   product_id    26340 non-null  object
 2   review_text   26340 non-null  object
 3   rating_class  26340 non-null  object
 4   year          26340 non-null  int64
 5   clean_text    26340 non-null  object
 6   token         26340 non-null  object
dtypes: int64(1), object(6)
memory usage: 2.9+ MB
```

Preprocessing of Review_text:

After that, we tokenized the words using a tokenizer from the NLTK library. Unstructured text also contains a lot of noise, particularly when it's used for web or screen scraping. HTML tags are a good example of components that don't contribute anything to the interpretation and analysis of text and should be excluded. For HTML tag cleanup, we used the BeautifulSoup library. Standard expressions were used to exclude special characters, which are non-alphanumeric characters that add little meaning to text and cause noise (regex).

Square brackets, URLs, and numbers were also omitted. Stopwords were also eliminated. Stopwords are unimportant words that do not alter the semantic sense of the text if they are removed from sentences. The accent has been withdrawn (It transliterates any unicode string into the closest possible representation in ascii text). Shortened forms of words or syllables are known as contractions. They're made by removing one or more individual letters from words. A contraction is often made up of more than one letter. Text standardization is aided by converting each contraction to its extended, original form. We used a regular collection of contractions from the CONTRACTION MAP library to expand contractions.

5. Proposed System Architecture:



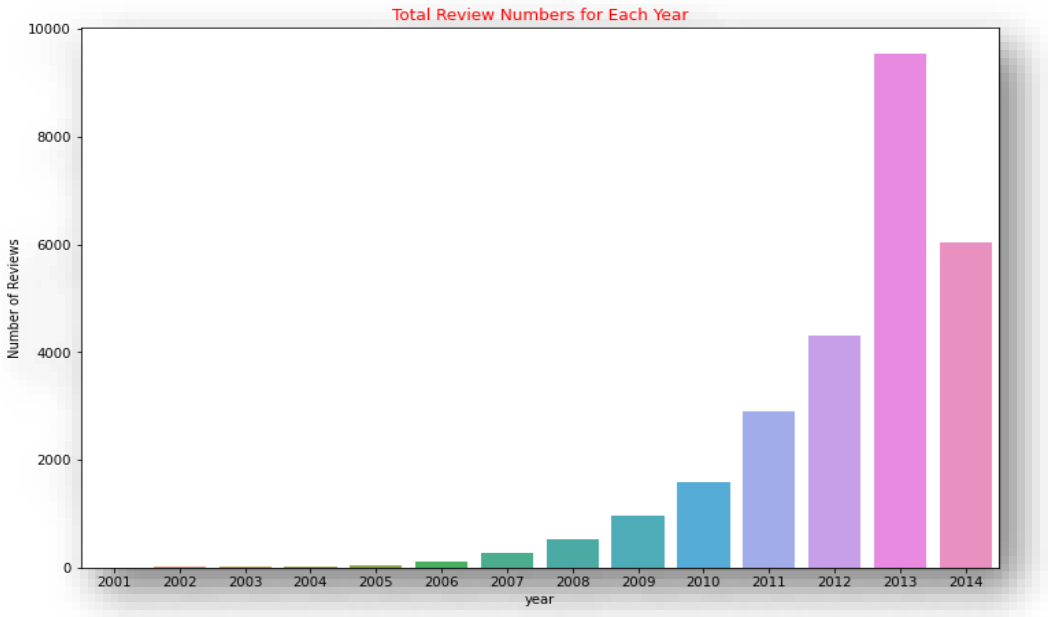
The proposed system has the following components:

1. Review Dataset: The dataset is obtained from https://jmcauley.ucsd.edu/data/amazon/ and this dataset has Amazon products reviews. Here we considered only Electronics section and from this only Speakers.

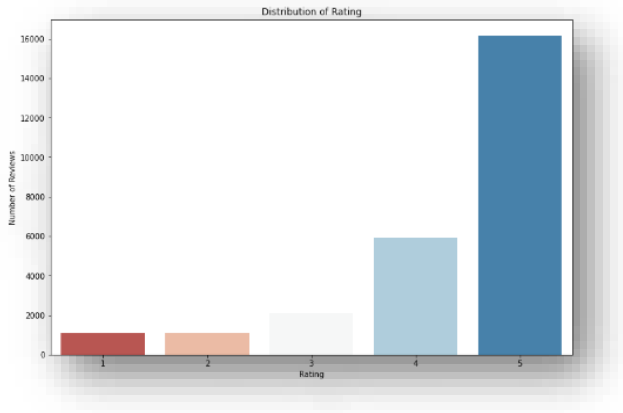
2. Exploratory data analysis:

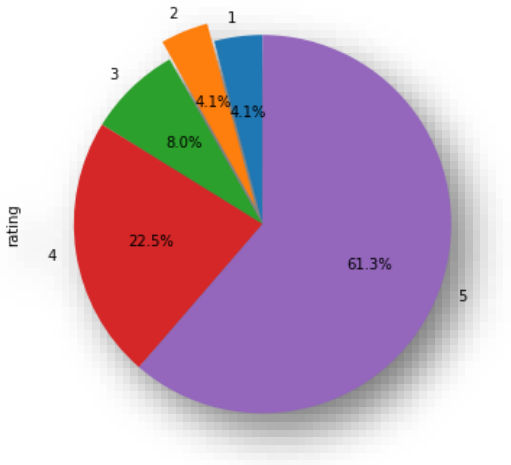
Data is analyzed using exploratory data analysis (EDA). Using various visualizations, summarize the key characteristics.

After Data Preprocessing, we performed EDA and found few interesting facts and ran down below.

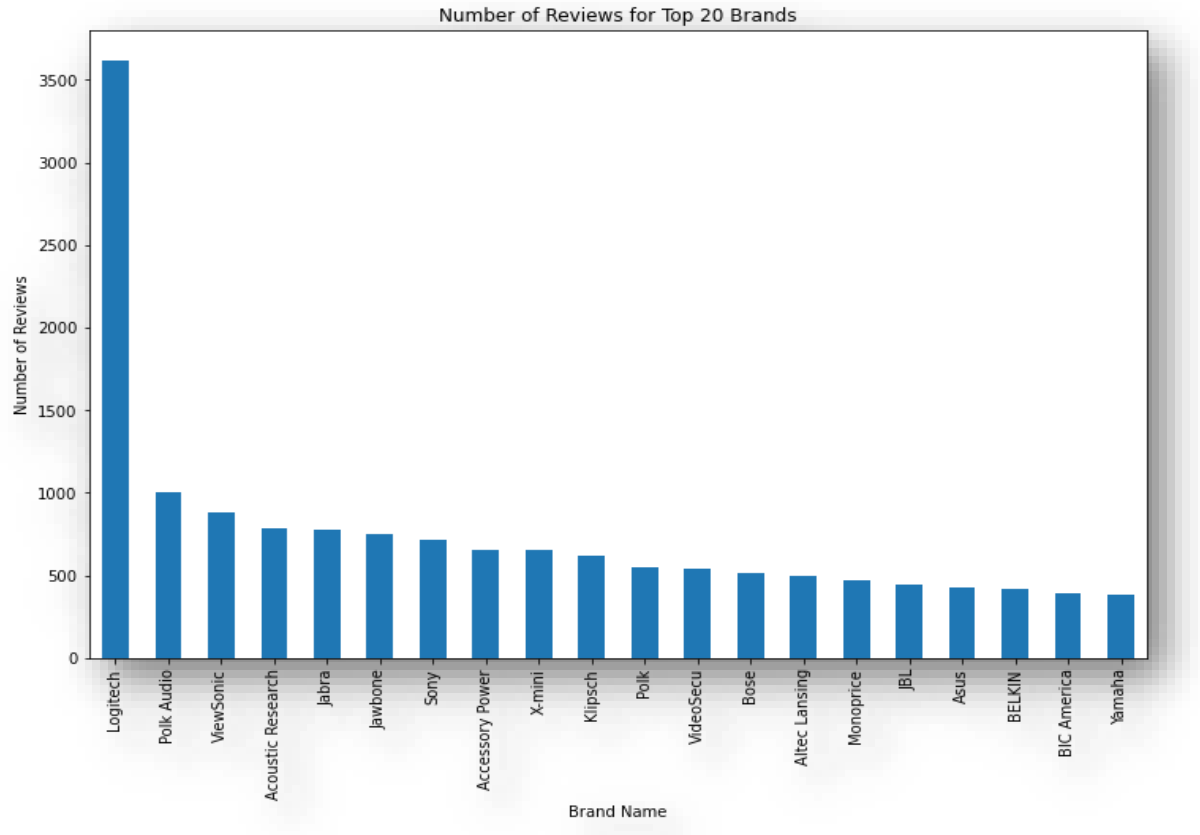


The above statistical graph speaks about the total number of reviews each year. Here we could see that the reviews started in 2001 with 1 count and have higher reviews in 2013 with 9551 reviews. 2014 year is halfway when this dataset is extracted.

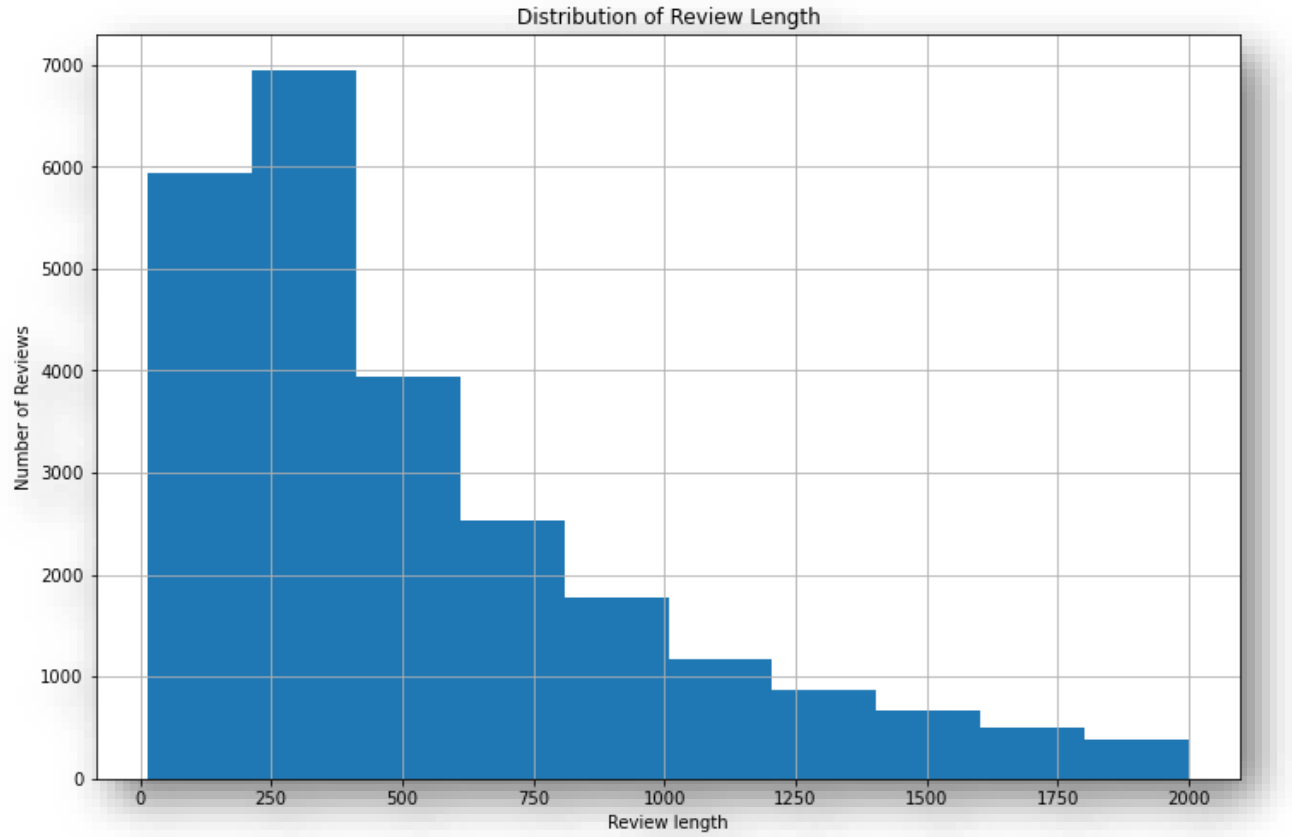




The above graph speaks about the distribution of Ratings. The 5start ratings are high with 16159 while the 1star ratings count are 1083 and 2 star are 1076.



The above graph says about the number of reviews for top 20 brands in the dataset where we can see Logitech speakers are getting more reviews while Yamaha speakers has got less.



This graph talks about the length of reviews given. Most of the reviews are approximately 250 characters in length.

3.Data Pre-Processing: As mentioned under preprocessing of review_text.We used the NLTK library to tokenize the data, then we lowered the case, removed noise, expanded contradictions, and removed accents.

4. Feature Engineering:
Feature engineering is the process of creating features for machine learning algorithms using domain knowledge of the data. If performed correctly, feature engineering improves the predictive ability of machine learning algorithms by generating features from raw data that aid in the learning process. Feature Engineering is both a science and an art.
Text stemming was used in feature engineering to aid in the reduction of derived words to their word stem, base, or root form. The stem does not have to be the same as the original expression. Stemming can be done in a variety of ways, including using a lookup table or using suffix-stripping algorithms. Since the conversion isn't useful, these depend on chopping off's', 'es', 'ed', 'ing', 'ly', and so on from the end of the terms. However, stemming aids in text standardization. The sample from the new dataset shown below will be used to implement all of the data mining steps listed above.

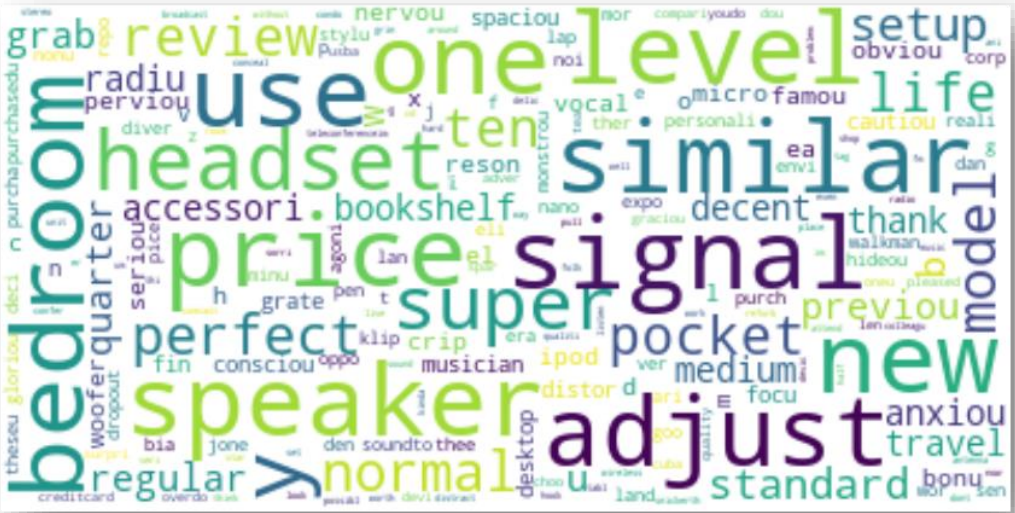
```
product_reviews_sp2['review_text']
```

```
1420       Is it just me Im shop around for wireless spea...
1421       work OK 25 star work ok not worth the price ta...
1422       one broke other complet fuzz I purchas these s...
1423       pretti good speaker I got these for christma o...
1424       good enough that I want to buy a third pair I ...
                                 ...
1689178    simpli stellar super sonic speaker system can ...
1689179    excel bluetooth speaker with lot of bell and w...
1689180    best sound speaker at thi price rang My short ...
1689181    impress sound stylish excel price the creativ ...
1689182    whi thi will Be amazon top sell portabl blueto...
Name: review_text, Length: 26340, dtype: object
```

The main approach we took was to simplify representation, also known as "bag-of-words," which ignores grammar and word order while keeping the word simple and counting the number of times each word appears. The tally by constructing a matrix of tokenized word counts from text documents, Vectorizer improves the usability of the "bag-of-words" process. We have used the Tf-idf vectorizer to measure the word density.

5. Review Analysis: We used the package spacy to predict the tokens in a sentence after Feature Engineering, and we used the displacy visualizer to visualize all of the feedback. We found the noun, verb, pronoun, and adjective using the dependency parse of the coarse POS tag, as well as the dependency tag.

In addition, we built a word cloud. We did this by converting the summary text to lower case and then tokenizing it into sentences.



6.Sentimental Classification:

Sentiment analysis is the method of analyzing consumer sentiment using natural language processing, text analysis, and statistics. The best companies are aware of their customers' feelings—what they're doing, how they're doing it, and what they mean. Tweets, articles, reviews, and other places where people mention your brand will reveal customer sentiment. Sentiment Analysis is the domain of using software to grasp these feelings, and it's a must-know for developers and business leaders in today's workplace.

Advances in deep learning, like many other fields, have pushed sentiment analysis to the forefront of cutting-edge algorithms. To extract and categorize the sentiment of words into positive, negative, or neutral categories, we now use natural language processing, statistics, and text analysis.

Sentiment analysis is carried out using algorithms that identify words as positive, negative, or neutral using text analysis and natural language processing. This enables businesses to obtain a better understanding of how their customers feel about their brand.

In order to perform machine learning on text documents, for the binary model, we build two versions. The first version is sentimental analysis with vader_lexicon and two versions of Logistic Regression,Random forest and Naive Baye's Models such as using the techniques of word count with unigram, Tf-idf.

7.Result: we compared all the versions of binary models and chose the best model that has the highest accuracy.

6. DATA ANALYTICS APPROACHES:

For all data analytics modelling, we split the original dataset into 80% training set and 20% test set and then text corpus was transformed into numeric vectors to apply Scikit-learn. We performed Logistic Regression, Naïve Bayes and Random Forest Classifier using TF-IDF Vectorization and Bag of words.

Logistic Regression:

Logistic Regression is a Machine Learning algorithm that is used to solve classification problems. It is a predictive analysis algorithm that is based on the probability principle.
The supervised learning classification algorithm logistic regression is used to estimate the likelihood of a target variable. Since the existence of the target or dependent variable is dichotomous, there are only two classes.

Random Forest:

Random forest is a learning algorithm that is supervised. It creates a "forest" out of an ensemble of decision trees, which are normally trained using the "bagging" process. The bagging method's basic premise is that combining different learning models improves the overall outcome.

Naïve Bayes:
It's a classification method based on Bayes' Theorem and the presumption of predictor independence. A Naive Bayes classifier, in simple terms, assumes that the existence of one function in a class is unrelated to the presence of any other feature.

1. Bag of words:

The Bag of Words model is one of the most basic but effective methods for extracting features from text documents. The Bag of Words or "Bag of n-grams" representation refers to this particular technique (tokenization, numbering, and normalization). The aim of this model is to transform text documents into vectors, with each document resulting in a vector that represents the frequency of all distinct words found in the document vector space for that particular document. We used the CountVectorizer to transform our set of text documents into a matrix of token counts after fitting it to our training data.

Logistic Regression:

Since Logistics Regression works well for high dimensional sparse data, we trained the Logistic Regression classifier on train_reviews feature matrix. Then, using y_test, we calculated the accuracy to be 93.68%.

```
# Call the modeling function for logistic regression with countvectorizer and print f1 score
modeling(LogisticRegression(multi_class = 'multinomial', solver = 'newton-cg',
                            C = 0.1, n_jobs = -1, random_state = 42))

# Assign y_pred to a variable for further process
y_pred_cv_logreg = y_pred
```

```
f1 score: 0.9368778879137922
```

Random Forest:
we trained the Random Forest on train_reviews feature matrix. Then, using y_test, we calculated the accuracy to be 88.21%.

```
# Call the modeling function for random forest classifier with countvectorizer and print f1 score
modeling(RandomForestClassifier(n_estimators = 100, random_state = 42))

# Assign y_pred to a variable for further process
y_pred_cv_rf = y_pred
```

```
f1 score: 0.8821639649917258
```

Naïve Bayes:

we trained the Naïve Bayes on train_reviews feature matrix. Then, using y_test, we calculated the accuracy to be 90.02%.

```
# Call the modeling function for naive bayes with countvectorizer and print f1 score
modeling(MultinomialNB())

# Assign y_pred to a variable for further process
y_pred_cv_nb = y_pred
```

```
f1 score: 0.9024898627972953
```

2. TF-IDF Model:

Term Frequency-Inverse Document Frequency (TF-IDF) is that combines two metrics: term frequency and inverse document frequency. The TF-IDF score was added to our Bag of Words model to help us concentrate on more meaningful words. TF-IDF weights terms based on how uncommon they are in our dataset, excluding words that are overly common and simply add to the noise.

Logistic Regression:

We applied the tf–idf vectorizer with ngram_range = 3(trigram) to the multinomial method, and fit it to our training data, and then we trained the Logistic Regression classifier. Next, we predicted using y_pred_tfidf_logreg, and computed the accuracy as 87.98.%.

```
# Call the modeling function for logistic regression with TF-IDF and print f
modeling(LogisticRegression(multi_class = 'multinomial', solver = 'newton-cg
                            tfidf_vect_train, tfidf_vect_test)

# Assign y_pred to a variable for further process
y_pred_tfidf_logreg = y_pred
```

```
f1 score: 0.8798521901204276
```

Random Forest:

We applied the tf–idf vectorizer with ngram_range = 3(trigram) with n_estimator = 100 and fit it to our training data, and then we trained the Random Forest classifier. Next, we predicted using y_pred_tfidf_rf and computed the accuracy as 90.22%.

```
# Call the modeling function for random forest classifier with TF-IDF and pr
modeling(RandomForestClassifier(n_estimators = 100, random_state = 42),
         tfidf_vect_train, tfidf_vect_test)

# Assign y_pred to a variable for further process
y_pred_tfidf_rf = y_pred
```

```
f1 score: 0.9022406588318006
```

Naïve Bayes:

We applied the tf–idf vectorizer with ngram_range = 3(trigram) with multinomialNB() and fit it to our training data, and then we trained the Naive Bayes classifier. Next, we predicted usingy_pred_tfidf_nb, and computed the accuracy as 88.03%.

```
# Call the modeling function for naive bayes with TF-IDF and print f1 score
modeling(MultinomialNB(), tfidf_vect_train, tfidf_vect_test)

# Assign y_pred to a variable for further process
y_pred_tfidf_nb = y_pred
```

```
f1 score: 0.8803180096189991
```

3. Sentiment Analysis using Vader_lexicon:

We used NTLK and SpaCy Libraries in these.spaCy is a powerful and advanced library that is gaining huge popularity for NLP applications due to its speed, ease of use, accuracy, and extensibility. It's built for production use and provides a concise and user-friendly API. spacy has a number of different models of different sizes available for use, with models in 7 different languages (include English, Polish, German, Spanish, Portuguese, French, Italian, and Dutch), and of different sizes to suit our requirements. we installed the library en_core_web_lg, which includes 685k unique vectors with 300 dimensions.
VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically tuned to sentiments expressed in social media. It

evaluates the text of a message and gives you an assessment of not just positive and negative, but the intensity of that emotion as well. It uses a dictionary of terms that it can evaluate Negations,Contractions,Punctuation and Slang. The advantages of using vader is it doesn't require any training data and it is fast enough to be used with streaming data.
For applying sentiment analysis we used two columns clean_text and ratings.we converted the ratings column as binary variables 1 as positive and 0 as negative. In this dataset we found that there are 22076 positive reviews and 4264 negative reviews are there and then We used sentimentIntensity analyzer to score all the comments and then We used the **polarity_scores()** to obtain the polarity indices for the sentences. The compound score is used by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). This is the most useful metric if you want a single unidimensional measure of sentiment for a given sentence.

```
#accuracy score
accuracy_score(df['rating'],df['comp_score'])
```

```
0.8458238420652999
```

7. Experimental Results and Analysis:

In our case of this data, we must evaluate the classifier's output using appropriate criteria that take into account the class distribution and pay special attention to the minority class. As a result, we used the f1 score as my assessment measure, which is the harmonic average of precision and recall.

It's critical to comprehend the various types of errors that our model produces. A Confusion Matrix, which compares the predictions our model makes with the true mark, is a good way to visualize the detail. With this in mind, we used an uncertainty matrix in addition to our evaluation metric (f1 score).

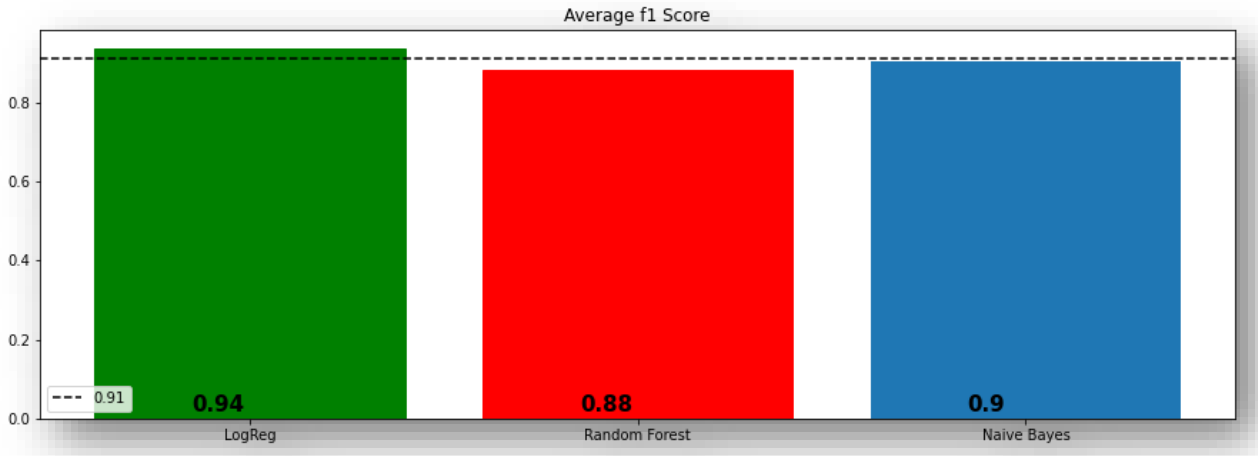
1. Bag of words Model: The binary system with bag of words the table below

	Logistic Regression	**Random Forest**	**Naive Bayes**
Test set	94.13%	91.97%	92.78%

Fig: Binary system with Bag of words

The winner is Logistic Regression, which has a score of 0.941344.

vectorizer	model	accuracy	class	precision	recall	f1-score	support
CountVect	LogReg	0.941344	bad	0.693811	0.497664	0.579592	428.0
			good	0.956662	0.980579	0.968473	4840.0
			average	0.935307	0.941344	0.936878	5268.0
	Random Forest	0.919704	bad	1.000000	0.011682	0.023095	428.0
			good	0.919628	1.000000	0.958131	4840.0
			average	0.926157	0.919704	0.882164	5268.0
	Naive Bayes	0.927866	bad	0.875000	0.130841	0.227642	428.0
			good	0.928517	0.998347	0.962166	4840.0
			average	0.924169	0.927866	0.902490	5268.0



2. TF-IDF Model: The binary system with TF-IDF the table below

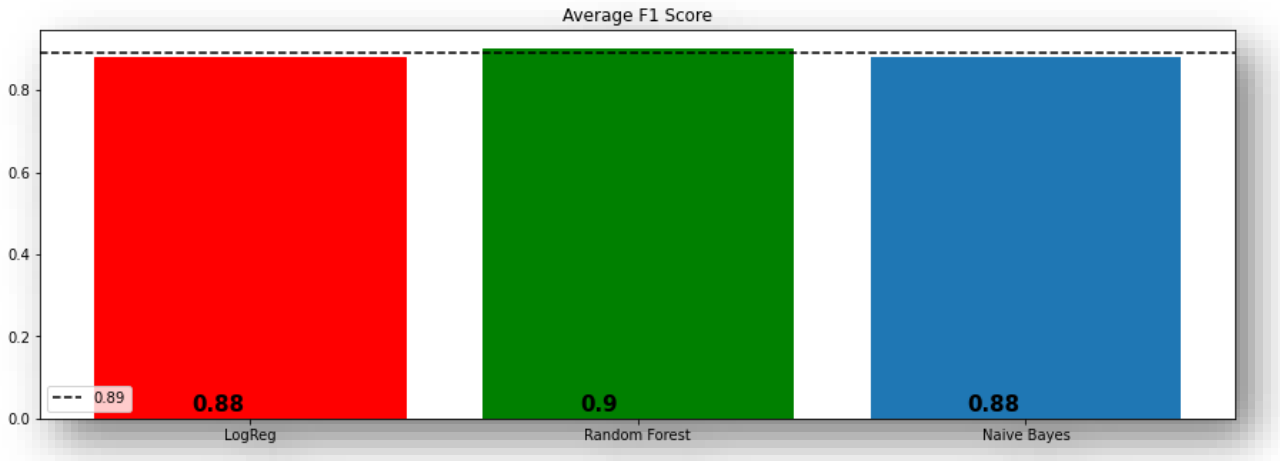
	Logistic Regression	Random Forest	Naive Bayes
Test set	91.87%	92.78%	91.89%

Fig: Binary system with TF-IDF

The winner for the TF-IDF is Random Forest with 0.927866.

Comparision Matrix of Models with TF-IDF Vectorizer

vectorizer	model	accuracy	class	precision	recall	f1-score	support
CountVect	**LogReg**	**0.918755**	**bad**	0.000000	0.000000	0.000000	428.0
			good	0.918755	1.000000	0.957657	4840.0
			average	0.844110	0.918755	0.879852	5268.0
	Random Forest	**0.927866**	**bad**	0.887097	0.128505	0.224490	428.0
			good	0.928352	0.998554	0.962174	4840.0
			average	0.925000	0.927866	0.902241	5268.0
	Naive Bayes	**0.918945**	**bad**	1.000000	0.002336	0.004662	428.0
			good	0.918929	1.000000	0.957752	4840.0
			average	0.925516	0.918945	0.880318	5268.0



3. The accuracy of the Vader lexicon is 0.8458

8. Conclusion:

Overall, we built a binary model and improved it to a multi-class model. To conclude, the final best result that comes after data modeling for bag of words is Logistic regression. It was used using a multinomial method which resulted in providing the best performance overall. For the TF-IDF, Random forest resulted in providing the best performance and are top models.

9. Future work:

In future work we can focus on improving the model to decide whether 5 stars recognition model is possible with machine learning algorithms. With the 5 stars recognition model, we may predict any review corpus into 5 stars-scale. We might use the techniques for improving the model accuracy such as hyperparameter tuning grid search, improving other metric scores if applicable and different models like support vector machine, gradient boost and so on.

References:

1. Introduction to sentiment analysis: What is sentiment analysis? (2021, April 28). Retrieved May 09, 2021, from https://algorithmia.com/blog/introduction-sentiment-analysis
2. Shekhar, A. (2019, December 06). What is feature engineering for machine learning? Retrieved May 09, 2021, from https://medium.com/mindorks/what-is-feature-engineering-for-machine-learning-d8ba3158d97a
3. Pant, A. (2019, January 22). Introduction to logistic regression. Retrieved May 09, 2021, from https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148
4. Machine learning - logistic regression. (n.d.). Retrieved May 09, 2021, from https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm
5. Donges, N. (n.d.). A complete guide to the random forest algorithm. Retrieved May 09, 2021, from https://builtin.com/data-science/random-forest-algorithm
6. Sunil RayI am a Business Analytics and Intelligence professional with deep experience in the Indian Insurance industry. I have worked for various multi-national Insurance companies in last 7 years. (2020, October 18). Learn naive BAYES Algorithm: Naive Bayes Classifier examples. Retrieved May 09, 2021, from https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
7. Machine learning - logistic regression. (n.d.). Retrieved May 09, 2021, from https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm