**Sentimental Analysis on Amazon Ring Video Doorbell.**

Sentimental analysis is the process of detecting positive or negative sentiments in text. By using this analysis we can identify 4000+reviews about one product and help discover if customers are happy. In this NLP project my team and I are going to perform a sentimental analysis on one of the top selling Amazon product Ring video doorbell based on 63156 customer ratings and reviews.

**Team Members:**

Priyanka Gopu

Vijaya Maneesha Reddy Duggimpudi

Manvitha Reddy Karra

**GitHub Link** : <https://github.com/PriyankaGopu/Sentimental_Analysis.git>

Videos of all the three:

Priyanka Gopu-

[VIDEO-2021-11-29-21-42-38.mp4](https://github.com/PriyankaGopu/Sentimental_Analysis/blob/main/VIDEO-2021-11-29-21-42-38.mp4)

Vijaya Maneesha Reddy Duggimpudi-

[VijayaManeesha Increment 2-1.mp4](https://github.com/PriyankaGopu/Sentimental_Analysis/blob/main/VijayaManeesha%20Increment%202-1.mp4)

[VijayaManeesha Increment 2-2.mp4](https://github.com/PriyankaGopu/Sentimental_Analysis/blob/main/VijayaManeesha%20Increment%202-1.mp4)

[VijayaManeesha Increment 2-3.mp4](https://github.com/PriyankaGopu/Sentimental_Analysis/blob/main/VijayaManeesha%20Increment%202-1.mp4)

Manvitha Karra-

[Manvitha\_reddy\_increment2.mp4](https://github.com/PriyankaGopu/Sentimental_Analysis/blob/main/Manvitha_reddy_increment2.mp4)

[Manvitha\_reddy\_increment\_2-1.mp4](https://github.com/PriyankaGopu/Sentimental_Analysis/blob/main/Manvitha_reddy_increment_2-1.mp4)

[Manvitha\_reddy\_increment\_2-2.mp4](https://github.com/PriyankaGopu/Sentimental_Analysis/blob/main/Manvitha_reddy_increment_2-2.mp4)

**Motivation:**

We have been inspired by sentimental analysis, a kind of data mining where you measure the inclination of people’s opinions by using NLP , text analysis, and computational linguistics. We perform sentiment analysis mostly on public reviews, social media platforms, and similar sites. However, analysis of social media streams is usually restricted to just basic sentiment analysis and count based metrics. This is akin to just scratching the surface and missing out on those high value insights that are waiting to be discovered.

With the recent advances in deep learning, the ability of algorithms to analyse text has improved

considerably. Creative use of advanced machine learning techniques can be an effective tool for

doing in-depth research. Thus, we would like to explore ourselves more on product-based

customer review analysis in one of the world's largest e-commerce websites Amazon. We chose

Amazon Ring Video Doorbell product for sentimental analysis since it has given top ratings and

nearly thousand people's questions were readily answered.

We also believed to classify incoming customer conversation about a brand based on following lines:

1. Key aspects of a brand’s product and service that customers care about.
2. Users’ underlying intentions and reactions concerning those aspects.

**Significance:**

* Sentimental analysis is a proven predictive indicator of customer satisfaction widely applied to voice of the customer materials such as reviews, survey responses , online and social media.
* Discover the customer experience and operational efficiency.
* By using these analysis we can help superior customer service at lower cost
* Identifying key emotional triggers
* Handling multiple customers
* Live insights and quick escalations

Feeling investigation worries about naturally distinguishing opinion or

assessment communicated in a given piece of text. Most earlier work either use earlier

lexical information characterized as opinion extremity of words or view the undertaking

as a message grouping issue and depend on a named corpora to prepare a feeling

classifier. The lacking data that is related with the minority class obstructs making an

unmistakable comprehension of the inborn design of the dataset. Most existing

grouping techniques tend not to perform well on minority class models when the

dataset is amazingly imbalanced, on the grounds that they mean to improve the

general exactness disregarding the overall appropriation of each class.

Our outcomes from sentimental analysis on this product show a normal increment of F consonant exactness score for recognizing both negative and positive feeling over the baselines of unigrams and grammatical form separately. The results could be additionally utilized by expert engineers in the source district to measure the dormant interest for their abilities in different areas and along these lines to focus on the internationalization of their abilities appropriately.

**Literature review**

In "Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning," Rain and Callen extended existing work in the disciplines of natural language processing and sentiment analysis to data from Amazon review datasets. It was used to extract characteristics from a bag of words in this investigation. This approach might be used to develop systems that assess diverse collections of data, however it may be more effective with smaller datasets. When dealing with small data sets, the algorithms performed effectively, even when training and testing on dramatically dissimilar commodities. This might be used to evaluate not just various things, but also different features of a same product. To identify whether a review was positive or negative, we used naïve Bayesian and decision list classifiers. They've picked books and written reviews for Amazon's Kindle section. As you can see, the accuracy of Naive Bayes is rather good in both scenarios, for both the book review and the kindle. This is because Naive Bayes is a simple algorithm. This approach employs just the text that has the greatest likelihood of being classified correctly, while decision List does not utilize any of the text that has the highest probability of being classed correctly. The more samples used, the longer it will take decisionList to finish the classification process, and in certain cases, it may be unable to do so at all.

Elli Maria and Yi-Fan released "Amazon Reviews, Business Analytics with Sentiment Analysis," in which they gathered and evaluated sentiment from Amazon reviews in order to construct an Amazon business model. The objective of this essay is to extract sentiment from over 2.7 million reviews and to investigate the commercial implications of such feelings. It's called Amazon product data, and it's the dataset on which our research is based. They assert that the tools they have offered are sufficiently robust to ensure high levels of accuracy. Their selection was more appropriate due to the application of business analytics. Additionally, they focused on understanding emotions in reviews, distinguishing gender based on names, and detecting fraudulent reviews. At the time, Python and R were the most extensively used programming languages.

They mostly used multinomial naive bayesian (MNB) and support vector machine (SVM) classifiers. The classifiers utilized are Multinomial Naive Bayes (MNB) and Support Vector Machine (SVM), which were created using the Python programming language by Joachims et al. (1998). They trained both classifiers using 50% of the data and then tested them with the remaining 50% of the data to assess the classification's accuracy. Both cases are quite precise. Take note that the processing timeframes for the two procedures are radically different. This is because Naive Bayes is a simple algorithm. This technique utilizes just fundamental arithmetic operations, while SVM does not. With an increase in the number of samples, SVM will take an increasing amount of time to finish the classification process, and in certain cases, will be unable to do it at all.

**Objective:**

In order to illustrate that text reviews can be mined through and feature-based feedback can be extracted for any product, a prototype is being developed. This system may be configured to utilize text reviews from any online buying website, such as eBay, Target, or Walmart, with only minimal modifications to the implementation portion. The dataset for the creation of such a system was derived from Amazon reviews and product sales data. Identification of significant aspects of the product is accomplished via the use of Natural Language Processing and text mining methods. The polarity (positive or negative) of each review is determined by using sentiment analysis. The result is generated by the application of machine learning techniques.The objective is by using the Sentiment analysis, which is used to analyze the customer reviews and then it classifies the sentiment as Positive, Negative or Neutral, in order to understand the customer opinion on Amazon Ring Video Doorbell.There are mainly two types of approaches used for

sentiment analysis.

They are:

I, Lexicon-based approach

Ii, Machine learning Based approach

* Here, we are using the second approach i.e Machine learning based approach.
* Machine learning based approach means here we Develop a classification model, which is trained using the pre-labeled dataset of positive, negative and neutral.
* We use NLTK. NLTK stands for Natural Language Toolkit. This toolkit is one of the most powerful NLP libraries which contains packages to make machines understand human language and reply to it with an appropriate response
* NLTK consists of the most common algorithms such as tokenizing, part-of-speech tagging, stemming, sentiment analysis, topic segmentation and named entity recognition. NLTK helps the computer to analyze, preprocess,and understand the written text.
* Here, we take the data and then perform the Tokenization, preprocess the data,Feature engineering, Model building and Model evaluation.
* Hence we use both analysis and analytical skills to prove our data is correct .

**FEATURES:**

Amazon Ring Video Doorbell is the one of the most commonly used products in every home to Keep Home &amp; Neighborhood Safe. It is the newest generation, 2020 release – 1080p HD video, improved motion detection, easy installation.There are almost 63,200 reviews for this amazon product. The Smart Security starts at the front door. Using this product we can see, hear and speak to visitors from anywhere. We can completely control it from the Ring app. It works with Alexa. Receives the mobile notifications when anyone presses the doorbell or it triggers the built-in motion sensors. We can adjust motion settings. So that, we can focus on key areas and we only receive notifications when we want. We can Pair with select Alexa-enabled devices to enable announcements and two-way talk for convenient in-home monitoring. There are different configurations like Doorbell only, With Echo Show 5, With Ring Chime, with Ring Stick Up Cam Battery. Powered by the built-in rechargeable battery or connected to existing doorbell wires for constant power. There are three types of video Doorbells. They are: Ring Video Doorbell, Ring Video Doorbell 3, Ring Video Doorbell 4. The Ring Video Doorbell and Ring Video Doorbell 3 are of 2020 release and Ring Video Doorbell is of 2021 release which is the latest.

A significant shortcoming of existing sentiment analysis tools when it comes to the problem of new products is that the text collected automatically has only a small number of insightful elements while simultaneously including a huge amount of useless information. It is common for users' subjective, distinctive, and emotional experiences to be connected with the noisy information they get. Furthermore A person's sentiments are naturally subjective from person to person, and they might even be completely unreasonable at times. An individual's sentiment toward a brand or product may be influenced by one or more indirect causes; for example, someone who is having a bad day may tweet a negative remark about something about which they otherwise had a fairly neutral opinion may be having a bad day because they are feeling down. In this case, we are able to solve this issues by choosing a product that has been in the market for sometime. We also chose the product based on a good number of positive reviews which makes it less likes to have sentiments based on irrationality.

By applying sentimental analysis to this product we can derive analysis and analytical conclusions.

**Increment 1:**

**Related Work(Background):**

The approach of sentimental analysis to the data from the amazon review datasets, we can extract more effective characteristics, features of a same product. This is used to evaluate positive or negative decision list classifiers. The more samples that is used the longer it will take the decision list to finish the classification process. We have taken 64000 reviews for amazon ring product with features like reviews, prices, description and ratings,

**Dataset:**

We have derived the result.xls dataset for amazon ring product.

Details design of features:

There are three types of video doorbells

1. Ring video doorbell
2. Ring video doorbell 2020 release
3. Ring video doorbell 2021 release

By applying the sentimental analysis to all these products based on the good number of positive reviews. We can derive analysis and analytical conclusions.

**Analysis:**

By Analyzing the reviews of all the three types of amazon ring product, we can conclude which product is the most customer friendly. It is common for user's subjective, distinctive and emotional experiences to be connected with the noisy information they get. It is naturally subjective from person to person.

**Implementation:**

The source code contains pip install bs4, lxml, request, selenium. We have imported beautiful soup which is a python library for extracting data out of html and xml files. It works with web driver <https://www.amazon.com>. Generating url from search term <https://www.amazonringvideodoorbell.com.> It works with favorite parser to provide idiomatic ways of navigating, searching and modifying the parse tree.

**Preliminary results:**

With the preliminary result, we can conclude that amazon ring video doorbell 2020 release has achieved 4.7 ratings out of 5 stars which is the median value calculated out of 67643 reviews. The median price range is also calculated as 99$.

**Project Management:**

Project management is the use of specific knowledge, skills, tool and techniques to deliver something of values to people. In this project, we tried to help people understand the best amazon ring product.

**Implementation status report:**

**Work Completed:**

**Description:**

We have completed the python code to derive the reviews from the amazon website of the amazon ring video doorbell product. We have been inspired by sentimental analysis to measure people's opinion by using NLP and Text analysis.

**Resposibility:**

Priyanka Gopu – Worked on Introduction, Significance and literature review and Worked on the Source code

Vijaya Maneesha Reddy Duggimpudi – Worked on the Objective and Features and Worked on source code

Manvitha Reddy Karra – Worked on Motivation and references and Worked on source code.

**Contributions:**

Priyanka Gopu – 33.5

Vijaya Maneesha Reddy Duggimpudi – 33.5

Manvitha Reddy Karra – 33

**Work to be completed:**

**Description:**

From the derived dataset, we have to imply more analytical and analysis skills to achieve in finding the most customer friendly amazon ring product.

**Responsibility:**

Priyanka Gopu – Analysis, Analytics and Data Visualization

Vijaya Maneesha Reddy Duggimpudi - Analysis, Analytics and Data Visualization

Manvitha Reddy Karra - Analysis, Analytics and Data Visualization

**Issues/Concerns:**

We have faced the difficulties in deriving the dataset.

Complications in filtering the data.

**REFERENCES**

“Ring Video Doorbell.” *Www.amazon.com*, [www.amazon.com/dp/B08N5NQ869/](http://www.amazon.com/dp/B08N5NQ869/)? tag=googhydr-20&hvadid=454491684868. Accessed 31 Oct. 2021.

Zhu, Jingbo, et al. "Aspect-basedopinion polling from customer reviews." IEEE Transactions on Affective Computing, Volume 2.1,pp.37-49, 2011. 

Na, Jin-Cheon, Haiyang Sui, Christopher Khoo, Syin Chan, and Yunyun Zhou. "Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews." Advances in Knowledge Organization Volume9, pp. 49-54, 2004. 

Nasukawa, Tetsuya, and JeongheeYi. "Sentiment analysis: Capturing favorability using natural language processing." In Proceedings of the 2nd international conference on Knowledge capture, ACM, pp. 70-77, 2003. 

Li, Shoushan, Zhongqing Wang, Sophia Yat Mei Lee, and Chu-Ren Huang. "Sentiment Classification with Polarity Shifting Detection." In Asian Language Processing (IALP), 2013 International Conference on, pp. 129-132. IEEE, 2013.

**Increment 2:**

**Introduction:**

Nowadays, consumers' items are increasingly being purchased and sold on the internet, as more and more customers and sellers turn to the internet for their purchases and sales. It is now possible to purchase things online, with most of the information on the products being supplied by users. This has altered the shopping experience significantly. This is in contrast to the way product information was previously disseminated via strategies such as word of mouth and advertising in the distant past, as described above. Since its inception as an online bookshop in 1994, Amazon.com has seen remarkable growth and has established itself as a pioneer in the field of online purchasing.

Within a short period of time, Amazon began to accept customer evaluations, and ultimately permitted any user to publish a review for any one of the millions of goods available on the site. Of light of the growth in anonymous user-generated material, it is imperative that efforts be made to comprehend the information that is provided in the proper context and create techniques for determining the purpose of the reviewer. Understanding what online consumers think about a firm's content may assist a company in marketing its product as well as maintaining its reputation on the internet.

**Abstract:**

Buyers of items from online shopping sites such as Amazon, BestBuy, and Walmart provide reviews on the things that they have bought. These unfiltered evaluations include a wealth of information and insights about the product under consideration. The amount of reviews varies greatly from one product to the next. The overall pattern is that if a product is a best seller, it is more likely to have positive customer reviews. In order to assess Amazon product evaluations, our project attempts to give both an overall score and a customized ranking based on certain product specific attributes. Customer feedback, ranging from survey replies to social media chats, may be automatically analyzed by companies, allowing them to listen more closely to their consumers and adjust goods and services to match their demands. For example, employing sentiment analysis to automatically scan 4,000+ reviews about your product might assist you in determining whether or not consumers are satisfied with your price plans and customer service levels. It is used to determine the sentiment score of each gathered review of a collection of goods, which is based on a Lexicon resource-based classifier. The top touch-screen-specific and popular laptops, as well as some of the most popular tablets, are assessed for inclusion in the study's dataset. The errors in our study, as well as the general issues in the selection of traits, are examined and explained in detail.

**Background:**

Until recently, the bulk of research publications on product evaluations, sentiment analysis, and opinion mining have been published during the previous several years. Elli, Maria, and Yi-Fan conducted a research in which they extracted sentiment from reviews and utilized the data to construct a business model . They assert that the tools they have offered are sufficiently robust to ensure their correctness. Their selection was more appropriate due to the application of business analytics. Additionally, they focused on understanding emotions in reviews, predicting gender based on the names, and identifying fraudulent reviews. At the time, Python and R were the most extensively used programming languages.

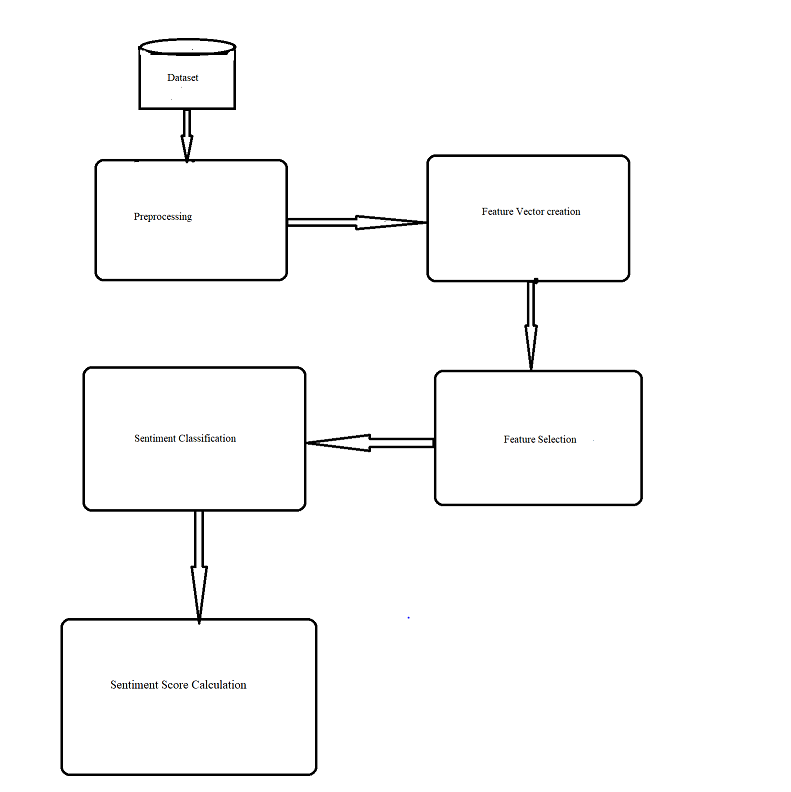
They mostly used multinomial naive bayesian (MNB) and support vector machine (SVM) classifiers. In the research, the author employed simple text to estimate a review rating on a certain number scale using a numerical scale given. A hold out cross validation procedure was utilized, with 70% of the data used for training and 30% for testing. According to the author, precision and recall values were computed in this article using a variety of classifiers.To identify whether a review was positive or negative, we used naïve Bayesian and decision list classifiers. They selected books and assessed the Kindle sector on Amazon. According to the objective was to create a mechanism for visualizing the sentiment of evaluations using charts. They accomplished this by grabbing data from the Amazon website and preprocessing it.

The authors of this paper tackle their difficulties using NB, SVM, and maximum entropy. Because the report says that they described the product assessment as the primary point, the research demonstrates no proof of accuracy. They visualized their results using a statistical graphic. The researchers used a bag-of-words method to data collecting to construct a model for predicting product ratings based on rating language. Unigrams and bigrams were employed in the evaluation of the models. They analyzed a sample of Amazon customer evaluations for video games collected at the University of California, San Diego. When time-based models were used, the findings were less than acceptable due to the little variance in average rating across years, months, and days. When unigrams were compared to bigrams, the unigrams produced the most accurate result overall. Additionally, popular unigrams were shown to be especially strong predictors of ratings due to their increased variation. In compared to bigrams, unigrams performed 15.89 percent better. Numerous feature extraction or selection procedures for sentiment analysis are discussed in article across a variety of circumstances. They started by gathering data from Amazon, followed by preprocessing the dataset to eliminate stop words and odd characters. They selected or extracted features at the phrase level, as well as at the single-word and multiword level. The classifier in this example is based on the Naive Bayes technique. As a result, they concluded that Naive Bayes outperforms single word and multiword classification at the phrase level.

The fundamental problem of this is that it depends only on the naive Bayes classifier approach, which is incapable of producing an acceptable outcome. Simpler strategies have been used in article to make the material easier to read and grasp. Due to the system's high accuracy on svm, it is impossible to perform effectively on huge datasets. They obtained their findings using support vector machines (svm), logistic regression, and decision trees. Tfidf is discussed in full in article as a supplemental experiment. It is capable of forecasting ratings via the use of a large number of distinct phrases. However, just a few Classifiers are used here. They started with a linear regression model with root mean square error. Thus, those are some significant efforts that were stated before, and we try to make our work more efficient by merging the best ideas from each. Our approach included a large number of datasets, which resulted in an efficient output that enabled us to make more informed judgments. Additionally, we labeled datasets using an active learning technique, which has the potential to greatly accelerate a broad variety of machine learning applications. Our system also includes a variety of different feature extraction methods. To the best of our knowledge, our proposed approach was more accurate than existing studies in the subject.

**Model:**

**Workflow Diagram:**



Amazon company does offer an Amazon Advertising API that delivers review-related data in the form of IFrames. When pulling reviews from an IFrame (which is an HTML page), we make use of an Amazon Scraper. The fact that Amazon is one of the most popular e-commerce sites means that there are an enormous number of reviews to be found on the site. Our data was derived from Amazon product data. We had to classify the data in order to utilize it in a supervised learning model since the dataset was unlabeled. We collected our dataset, which consisted of three distinct JSON types, and tagged it. Because we have a huge number of evaluations, manually categorizing them proved to be almost difficult for us to do. As a result, we preprocessed our data and labeled the datasets using Active Learner. As a result, we remove any reviews from our dataset that include a 3-star rating, and we save the rest of the reviews for use in the following phase, which is labeling the dataset.

Following the collection of reviews, we tokenize each review while taking into account all of the instances, such as managing negation, emoticons, URL, and so on. After that, we calculate the average sentiment score for each product as well as the feature-based score.

Each aspect of the project is modelled by an unordered grouping of words; hence, no regard is given to grammar in the texts represented as unordered collections of words in this project. When xi for i=1,..n, the training data D consists of a corpus of n documents (reviews), where xi for i=1,..n denotes the number of reviews in the corpus and |V| denotes the word dictionary that contains m different terms. In this approach, each word wj for j = 1,...,m is represented as a feature, and each feature is represented as a feature. Each word is allocated an integer identifier (id) by the system. Document I is kept in the array X[i,j], with the value of the feature j representing the number of times the word wj occurs. X is a sparse matrix of size n by m that only holds the non-zero sections of the feature vectors in its rows and columns.

For classification we have used the Naïve Bayes and decision tree classification and finally, we calculated the sentiment score. We used SENTIWORDNET, which is an Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining that will be used to construct Sentiment Score. SentiWordNet gives three types of Sentiment scores to each synset of WordNet: positivity, negativity, and neutrality. Each synset of WordNet is assigned one of the three types of Sentiment ratings. As a result, we compute a sentiment score for each review in our project and compile a dictionary of all of the reviews.

**Dataset:**

In spite of the fact that Amazon does not have an API similar to Twitter that directly gives all of the reviews, the company does offer an Amazon Advertising API that delivers review-related data in the form of IFrames. When pulling reviews from an IFrame (which is an HTML page), we make use of an Amazon Scraper written by Adam Griffiths. The fact that Amazon is one of the most popular e-commerce sites means that there are an enormous number of reviews to be found on the site. Our data was derived from Amazon product data. We had to classify the data in order to utilize it in a supervised learning model since the dataset was unlabeled.

We gathered information from three categories of Amazon products: Electronics reviews, Amazon door ring with speaker, Amazon door ring with video and speaker, and Amazon door ring with video and speaker. We collected our dataset, which consisted of three distinct JSON types, and tagged it. Because we have a huge number of evaluations, manually categorizing them proved to be almost difficult for us to do. As a result, we preprocessed our data and labeled the datasets using Active Learner. Because Amazon reviews are based on a 5-star rating system, 3-star ratings are typically considered neutral reviews, meaning they are neither favorable nor negative. As a result, we remove any reviews from our dataset that include a 3-star rating, and we save the rest of the reviews for use in the following phase, which is labeling the dataset. We totally have the three datasets for three different products and for three different years.

**Analysis of Data:**

we preprocessed our data and labeled the datasets using Active Learner. Because Amazon reviews are based on a 5-star rating system, 3-star ratings are typically considered neutral reviews, meaning they are neither favorable nor negative. As a result, we remove any reviews from our dataset that include a 3-star rating, and we save the rest of the reviews for use in the following phase, which is labeling the dataset.

Following the collection of reviews, we tokenize each review while taking into account all of the instances, such as managing negation, emoticons, URL, and so on. After that, we calculate the average sentiment score for each product as well as the feature-based score.

Classification:

Naive Bayes Classification:

Naive Bayes is a simple yet robust classifier that is implemented using Bayes' Rules and produces very helpful results. It assumes that each of the characteristics in the vector is independent of the others, and for each feature, it estimates the likelihood that it will exist given the class in which it appears. To calculate the chance of a class occurring based on the feature set, just add up the probability that the class will occur and the probabilities for each of the feature vectors in the feature set. This procedure is repeated for each of the available classes, and the text is categorized according to the class with the greatest likelihood of being correct.

Decision Tree Classification:

With the Decision list, you can create a rule-based tagger that has the benefit of being readable by humans. One of the most significant shortcomings of Naive Bayes is the difficulty in determining which probabilities are responsible for particular classifications in the first place. One may examine the mistakes that have been committed, but there is no way to determine whether or not there is a feature that is at the base of the issue. The structure of a decision list alleviates this difficulty by requiring the categorization to be based on rules rather than intuition. The classifier's main responsibility is to evaluate whether or not the feature exists at each level and to tag the feature properly if it does exist in the document. This makes it extremely simple to increase the rules that the classifier believes are significant, and it also makes it easier to remove elements that are creating erroneous rules.

**Implementation:**

"""

The program allows us to read amazon review data and using different

algorithms carry out a sentimental analysis and process the data

"""

#import packages required

import csv

import re

import string

import pandas as pd

from sklearn.svm import SVC

from collections import Counter

from sklearn.ensemble import RandomForestClassifier as RFC

from sklearn.neural\_network import MLPClassifier

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer, WordNetLemmatizer

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier as knc

from sklearn.ensemble import GradientBoostingClassifier as gbc

#Categorizing stop words in one section

stopWords = [] # create an empty list to collect the stopwords

dataSet = [] #create an empty list to collect the data

emoticons\_str = r"""

(?:

[:=;] # Eyes

[oO\-]? # Nose (optional)

[D\)\]\(\]/\\OpP] # Mouth

)"""

# set up regular expressions

"""

Do a regex search against all defined regexes and

return the key and match result of the first matching regex

"""

regex\_str = [

r'<[^>]+>', # HTML tags

r"(?:[a-z][a-z\-\_]+[a-z])", # words with - and '

r'(?:[\w\_]+)', # other words

r'(?:\S)' # anything else

]

#Tokenization of strings

tokens\_re = re.compile(r'(' + '|'.join(regex\_str) + ')', re.VERBOSE | re.IGNORECASE)

emoticon\_re = re.compile(r'^' + emoticons\_str + '$', re.VERBOSE | re.IGNORECASE)

stemmer = PorterStemmer()

lemmatiser = WordNetLemmatizer()

def initializeSystem():

stop = stopwords.words('english') + list(string.punctuation) + ['rt', 'via', 'i\'m', 'us', 'it']

for x in stop:

stopwords.append(stemmer.stem(lemmatiser.lemmatize(x, pos="v")))

#Carry our preprocessing of review data

def preprocess(s, lowercase=True):

tokens = tokens\_re.findall(s)

if lowercase:

tokens = [token if emoticon\_re.search(token) else stemmer.stem(lemmatiser.lemmatize(token.lower(), pos="v")) for

token in tokens]

return tokens

#Process strings with stop words and return

def processString(string):

terms\_stop = [term for term in preprocess(string) if

term not in stopwords and len(str(term)) > 1 and not term.isnumeric()]

return terms\_stop

#Function to manage reading of files

def loadFile(filePath):

fileRead = open(filePath, "r") #Read input file

reader = csv.reader(fileRead, dialect='excel') #Determine type of file

for row in reader:

temp = (row[1], row[-1])

dataSet.append(temp)

return dataSet

def prepareSparseMatrix(convertedReviews, decisionAttributes):

sparseMatrix = []

for cr in convertedReviews:

newCr = [0] \* len(decisionAttributes)

for word in cr:

if word in decisionAttributes:

index = decisionAttributes.index(word)

newCr[index] += 1

else:

pass

sparseMatrix.append(newCr)

return sparseMatrix

#Function to convert reviews to readable strings

def convertReviews(reviews):

convertedReviews = []

for a in reviews:

convertedReviews.append(processString(str(a).lower()))

return convertedReviews

def getDecisionAttributes(convertedReviews):

toCount = []

decisionAttributes = []

for a in convertedReviews:

toCount.append(" ".join(a)) #Join the strings

str1 = ""

for a in toCount:

str1 += "".join(a)

x = Counter (str1.split(" "))

for (k, v) in x.most\_common(min (500, len(x))):

decisionAttributes.append(k)

return decisionAttributes

#Function to train and process input data

def model\_data(training\_data):

dtc = DecisionTreeClassifier(random\_state=9, min\_samples\_split=5)

dtc.fit(training\_data['data'], training\_data['result'])

nn = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden\_layer\_sizes=(5, 2), random\_state=1)

nn.fit(training\_data['data'], training\_data['result'])

svc = SVC(C=100, kernel="linear")

svc.fit(training\_data['data'], training\_data['result'])

rfc = RFC(n\_estimators=10, criterion='entropy', max\_depth=10, min\_samples\_split=5, bootstrap='true', random\_state=None)

rfc.fit(training\_data['data'], training\_data['result'])

knc\_map = knc(n\_neighbors=15, weights='distance')

knc\_map.fit(training\_data['data'], training\_data['result'])

gbc\_map = gbc(n\_estimators=150, verbose=0)

gbc\_map.fit(training\_data['data'], training\_data['result'])

return {

'Decision Tree Classifier': dtc,

'Neural Networks': nn,

'Support Vector Machines': svc,

'Random Forest Classification': rfc,

'k Nearest Neighbours': knc\_map,

'Gradient Boosting Classifier': gbc\_map

}

#Function to test the train data

def test\_models(test\_data, models):

print ("Prediction rating:\n")

for model in models:

prediction = models[model]. score(test\_data['data'], test\_data['result']) \*100.00

print(str(model) + ": " + "%.2f" % prediction + "%") #Print the results

initializeSystem()

#Calling all the functions used in the data

#Load the test data

training\_data = loadFile("Data1.csv")

trainDataFeaturesReviews = pd.DataFrame(training\_data, columns=["review", "rating"])

targetRating = (trainDataFeaturesReviews['rating'])

targetReview = trainDataFeaturesReviews['review']

trainReviews = convertReviews(targetReview)

decisionAttributes = getDecisionAttributes(trainReviews)

trainSparseMatrix = prepareSparseMatrix(trainReviews, decisionAttributes)

dataFeatures = pd.DataFrame(trainSparseMatrix, columns=decisionAttributes)

training\_data = {

'data': dataFeatures,

'result': targetRating

}

#Load the test data

test\_data = loadFile("Data2.csv")

testDataFeaturesReviews = pd.DataFrame(test\_data, columns=["review", "rating"])

testReview = testDataFeaturesReviews['review']

testRating = testDataFeaturesReviews['rating']

testSparseMatrix = prepareSparseMatrix(convertReviews(testReview), decisionAttributes)

testDataFeatures = pd.DataFrame(testSparseMatrix, columns=decisionAttributes)

test\_data = {

'data': testDataFeatures,

'result': testRating

}

models = model\_data(training\_data)

test\_models(test\_data, models)

Plotting graphs based on the test data:

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

#test data results to plots

c = [0.01, 0.05, 0.5, 0.1, 10, 50, 100, 0.01, 0.05, 0.5, 0.1, 10, 50, 100, 0.01, 0.05, 0.5, 0.1, 10, 50, 100, 0.01, 0.05, 0.5, 0.1, 10, 50, 100]

y = [11.18530885, 11.18530885, 14.57985531, 11.18530885, 58.59766277, 58.59766277, 58.59766277, 97.38452977, 97.38452977, 97.38452977, 97.38452977, 97.38452977, 97.38452977, 97.38452977, 10.1836394, 10.1836394, 10.1836394, 10.1836394, 10.1836394, 10.1836394, 10.1836394, 95.9933222, 96.10461881, 95.9933222, 96.04897051, 95.9933222, 95.9933222, 95.9933222]

j = plt.figure(1)

j.suptitle('Support Vector Machine \n (Different C values)')

plt.title('Optical Character Recogniser')

plt.plot(c, y, 'ro')

plt.xlabel('C')

plt.ylabel('Predicted Percent')

plt.show()

y1 = [58.19672131, 58.19672131, 58.19672131, 58.19672131]

y2 = [58.19672131, 58.19672131, 59.83606557, 59.01639344]

y3 = [58.19672131, 58.19672131, 58.19672131, 58.19672131]

y4 = [60.6557377, 77.04918033, 93.44262295, 87.70491803]

z1 = [0.01, 0.05, 0.5, 0.1]

z2 = [0.01, 0.05, 0.5, 0.1]

z3 = [0.01, 0.05, 0.5, 0.1]

z4 = [0.01, 0.05, 0.5, 0.1]

fig = plt.figure()

ax = plt.axes(projection='3d')

ax.plot(y1, z1, label='RBF')

ax.plot(y2, z2, label='Poly')

ax.plot(y3, z3, label='Sigmoid')

ax.plot(y4, z4, label='Linear')

plt.title('Product Review Analysis')

ax.set\_xlabel('Kernel', )

ax.set\_ylabel('Predicted Percent')

plt.legend(loc=2)

plt.show()

x1 = [85.7540345, 85.14190317, 85.08625487, 85.19755147, 85.19755147, 85.19755147]

x2 = [84.75236505, 85.6983862, 86.47746244, 85.92097941, 84.86366166, 88.36950473]

x3 = [91.98664441, 94.15692821, 96.10461881, 96.82804674, 95.43683918, 85.7540345]

plt.figure()

plt.title('Boosting OCR')

plt.plot(x1, label='Max Depth: 10')

plt.plot(x2, label='Max Depth: 20')

plt.plot(x3, label='Max Depth: 30')

plt.legend(loc=2)

plt.show()

x1 = [0.01, 0.05, 0.1, 0.5, 1, 1.5]

x2 = [0.01, 0.05, 0.1, 0.5, 1, 1.5]

x3 = [0.01, 0.05, 0.1, 0.5, 1, 1.5]

y1 = [84.42622951, 90.16393443, 90.98360656, 91.80327869, 91.80327869, 91.80327869]

y2 = [91.80327869, 92.62295082, 92.62295082, 90.16393443, 90.98360656, 90.98360656]

y3 = [92.62295082, 90.98360656, 92.62295082, 91.80327869, 93.44262295, 93.44262295]

plt.figure()

plt.title('Boosting Product Review')

plt.plot(x1, y1, label='Max Depth: 10')

plt.plot(x2, y2, label='Max Depth: 20')

plt.plot(x3, y3, label='Max Depth: 30')

plt.legend(loc=2)

plt.show()

#Plot results from KNN product review

x = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

y = [92.62295082, 92.62295082, 92.62295082, 92.62295082, 92.62295082, 92.62295082, 92.62295082, 92.62295082, 92.62295082, 92.62295082]

plt.title('KNN Product Review')

plt.plot(x, y)

plt.show()

x = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

y = [97.82971619, 97.10628826, 96.82804674, 96.60545353, 96.38286032, 95.8263773, 95.38119087, 94.88035615, 94.54646633, 94.54646633]

plt.title('KNN OCR')

plt.plot(x, y)

plt.show()

#Plotting results from comparing all algorithms

x = [1, 2, 3, 4, 5, 6]

y = [79, 64.86, 75.57, 65.95, 77.62, 76.62]

l = ['Decision Tree Classifier','Support Vector Machines', 'Random Forest Classification','KNN']

plt.xticks(x, l)

plt.title("Algorithms Vs Accuracy")

plt.xlabel('Algorithm')

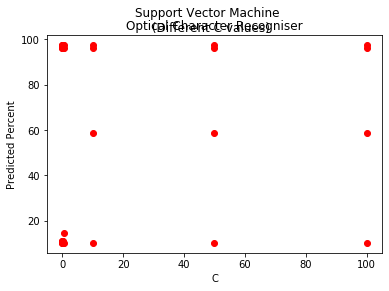
plt.ylabel('Accuracy')

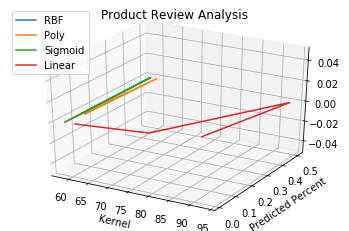
plt.plot(x, y)

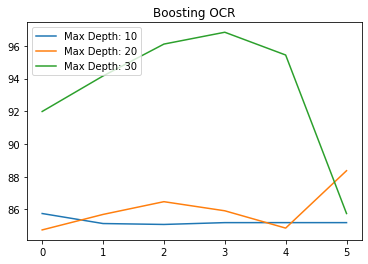
for i, j in zip(x, y):

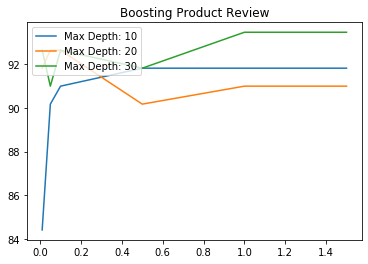
plt.annotate(str(j), xy=(i,j))

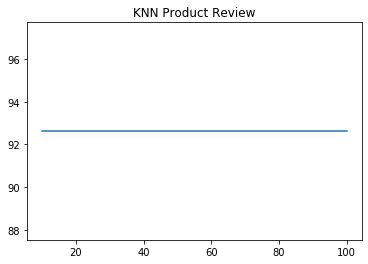
plt.show()

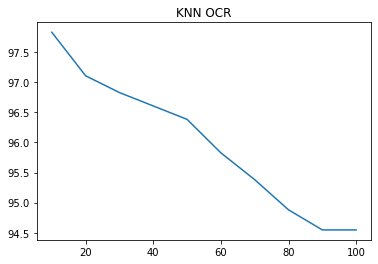


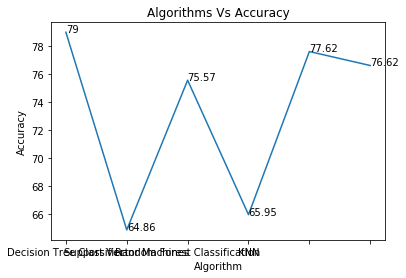












Sklearn is used in the above code which is most useful and robust library for machine learning in Python. It provides a great selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. We have also used the different classifiers like RandomForestClassifier and MLPClassifier, DecisionTreeClassifier, KNeighboursClassifier, GradientBoostingClassifier. A random forest is a meta estimator that employs averaging to increase predicted accuracy and control over-fitting by fitting a number of decision tree classifiers on various sub-samples of the dataset. A feedforward artificial neural network model called a multilayer perceptron (MLP) translates sets of input data onto a set of relevant outputs.

We start at the root of the tree and split the data on the feature that yields the most information gain (IG) using the decision method (reduction in uncertainty towards the final decision).

We may then continue this splitting operation at each child node in an iterative process until the leaves are pure. This indicates that all of the samples at each leaf node are of the same class.

The k nearest neighbors are represented by the K in the classifier's name, where k is an integer number given by the user. As the name implies, this classifier uses learning based on the k closest neighbors. The value of k is determined by the information available.

Gradient boosting classifiers use the AdaBoosting approach with weighted minimization, followed by recalculation of the classifiers and weighted inputs. Gradient Boosting classifiers aim to minimize the loss, or the difference between the training example's actual class value and the projected class value.

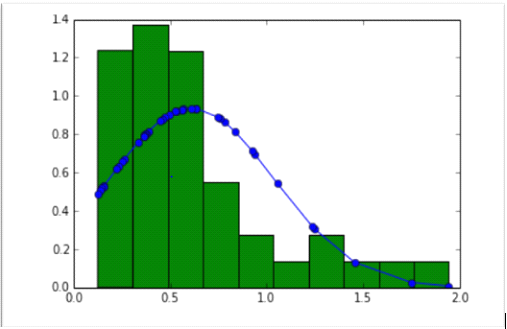
After importing the above required packages, we catogorize the stopwords in one section. Create an empty list to collect the stopwords and to collect the data. Then we setup the regular expressions and then the tokenization of strings is performed.and then we carry out the preprocessing of the review data. Function to manage the reading of the files and then we wrote a function to convert the reviews to readable strings and then we wrote a function to train and process input data and later we wrote the function to test the train data. Finally, calling all the function s used in data. Load the train and test data.

In the second program, we have plotted the graphs using the test data. We have plotted the graphs for support vector machine and KNN and Boosting product review and so on. Finally, we have plotted the graph from results by comparing all the algorithms.

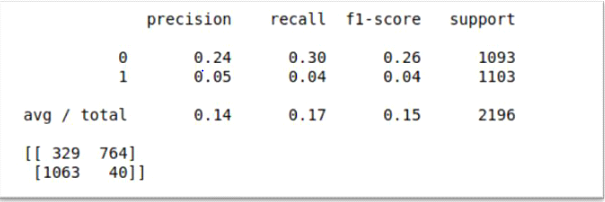
**Results:**

Sentiment Score Calculation

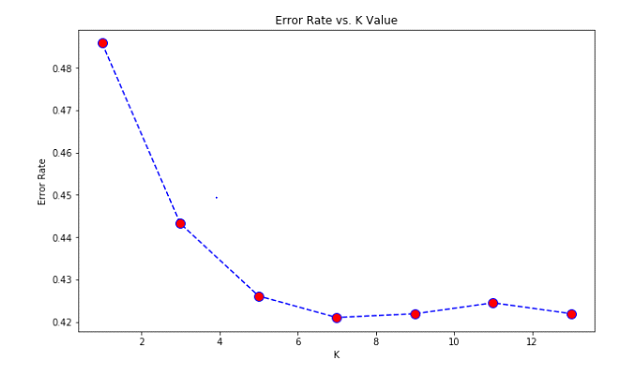
We used SENTIWORDNET, which is an Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining that will be used to construct Sentiment Score. SentiWordNet gives three types of Sentiment scores to each synset of WordNet: positivity, negativity, and neutrality. Each synset of WordNet is assigned one of the three types of Sentiment ratings. As a result, we compute a sentiment score for each review in our project and compile a dictionary of all of the reviews. As part of this research, we first generated an overall sentiment score for each product, and then we plotted a histogram of the results.



Aside from that, for feature-based rating, we are categorizing devices based on how well they perform certain characteristics (such as touchscreens or processors). Sometimes a customer is simply interested in a nice touchscreen laptop and is not concerned with the overall rating; in this case, we may also give this kind of service.



We have used Naïve Bayes Classifier and Decision Tree Classifier. We are getting maximum accuracy by using our best settings (i.e min\_df = 3 and max\_df = 0.4)



**Project Management:**

Project management is the use of specific knowledge, skills, tool and techniques to deliver something of values to people. In this project, we tried to help people understand the best amazon ring product.

As a whole, the outcomes of this endeavor were really positive. An adequate number of user-generated evaluations were successfully classified by the classification algorithm, with an accuracy of 67 percent for the naive bayes classifier and 80.33 percent for the decision tree classifier, according to the results. We did see that there were certain outlier goods in our project, which were distinguished by the fact that they received a small number of evaluations (less than 50). Because of this, our ranking graph has lengthy tails on the positive side, and our ranking is dispersed throughout. This is owing to the fact that people, on the whole, prefer to read favorable evaluations rather than negative ones.

**Implementation status report:**

**Work completed:**

**Description:**

We have completed the python code to derive the reviews from the amazon website of the amazon ring video doorbell product. We have been inspired by sentimental analysis to measure people's opinion by using NLP and Text analysis.

As a whole, the outcomes of this endeavor were really positive. An adequate number of user-generated evaluations were successfully classified by the classification algorithm, with an accuracy of 67 percent for the naive bayes classifier and 80.33 percent for the decision tree classifier, according to the results. We did see that there were certain outlier goods in our project, which were distinguished by the fact that they received a small number of evaluations (less than 50). Because of this, our ranking graph has lengthy tails on the positive side, and our ranking is dispersed throughout. This is owing to the fact that people, on the whole, prefer to read favorable evaluations rather than negative ones.

Because of this assumption, it is reasonable to infer that, given the limited number of reviews available, a product is likely to get more positive than negative reviews and so a high sentiment score. An example of this was found in a product that had just 6 reviews yet managed to rank on the upper end of our overall ranking histogram despite the limited number of reviews. This project might be further developed to give more feature-based ranking based on the study of user reviews, which would be a significant improvement. Additionally, the fundamental analysis of reviews reveals a general trend in the preferences of users, which manufacturing businesses may use to study and develop their products depending on the feedback received from customers.

As an example, consider Product A, which has a pretty decent rating and is listed in the top 10 goods in the overall ranking. However, if we look at Product A's rating based on a certain characteristic, it may be rated significantly lower than it otherwise would have been. Users' feedback patterns provide manufacturers with valuable information into which individual features of their goods are being rated negatively, which ultimately leads to an increase in the overall rating of their respective items.

**Resposibility:**

Priyanka Gopu – Worked on Introduction, Abstract and Worked on the Source code

Vijaya Maneesha Reddy Duggimpudi – Worked on the Background, workflow diagram, dataset and Worked on source code Manvitha Reddy Karra – Worked on Analysis of data and references and Worked on Source code.

**Contributions:**

Priyanka Gopu – 33.5

Vijaya Maneesha Reddy Duggimpudi – 33.5

Manvitha Reddy Karra – 33

**Issues/concerns:**

We have faced the difficulties in deriving the dataset.

Complications in filtering the data.

Faced difficulties while using the different classification algorithms.

**References:**

1. AmazonScraper by Adam Griffiths (<https://github.com/adamlwgriffiths/amazon_scraper>)
2. Naïve Bayes Classifier(http://scikit-learn.org/stable/modules/naive\_bayes.html)
3. DecisionTreeClassifier(http://scikit-learn.org/stable/modules/generated/ sklearn.tree.DecisionTreeClassifier.html)
4. SentiWordNet(http://lrec.elra.info/proceedings/lrec2010/pdf/769\_Paper.pdf)