

# Amazon Product Reviews Sentiment Analysis Using Supervised Learning Algorithms

by

Mohibullah Hawladar  
17101058

Arjan Ghosh  
20141045

Zaoyad Khan Raad  
17101077

Wali Ahad Chowdhury  
17301057

Md Sazzad Hossain Shehan  
17301150

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science

Department of Computer Science and Engineering  
Brac University  
June 2021

© 2021. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

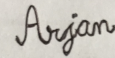
1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:



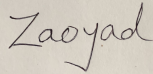
---

Mohibullah Hawladar  
17101058



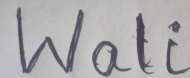
---

Arjan Ghosh  
20141045



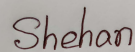
---

Zaoyad Khan Raad  
17101077



---

Wali Ahad Chowdhury  
17301057



---

Md Sazzad Hossain Shehany  
17301150

# Approval

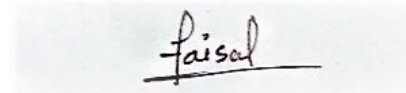
The thesis/project titled Product Reviews: Sentiment Analysis Using Supervised Learning Algorithms” submitted by

1. Mohibullah Hawladar (17101058)
2. Arjan Ghosh ( 20141045)
3. Zaoyad Khan Raad (17101077)
4. Wali Ahad Chowdhury (17301057)
5. Md Sazzad Hossain Shehan (17301150)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 6, 2021.

## Examining Committee:

Supervisor:  
(Member)



---

Faisal Bin Ashraf  
Lecturer

Department of Computer Science and Engineering  
Brac University

Program Coordinator:  
(Member)

---

Md. Golam Rabiul Alam  
PhD Associate Professor

Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)



---

Sadia Hamid Kazi

Chairperson and Associate Professor  
Department of Computer Science and Engineering  
Brac University

## **Ethics Statement**

The research is carried out in conformity with the standards of ethical guidelines of the BRAC University regulations and procedures. We employ information from original sources in our thesis. We are making certain that references and in-text citations are used correctly. We, the five co-authors, accept all responsibility for the thesis code infractions. We read several books and online resources to solve difficulties. We also enlisted the assistance of our university's faculty members. Finally, we proclaim that we are grateful to everyone who assisted us. We did not use any deceptive tactics to complete the thesis. Our research adheres to the ethics guidelines defined by BRAC University.

## Abstract

E-commerce is gaining traction in today's digitalized environment by taking products closer to customers without forcing them to leave their homes. A customer must study hundreds of reviews before making a purchase. The amount of internet evaluations for a single product can easily approach millions and make tracking and understanding of client feedback difficult. In the era of machine learning, however, it would be much easier to gain thousands of input and knowledge from them if a model were employed to polarize and understand from them. Consequently, sentiment analysis is a new study area combining natural language processing and text analytic to extract subjective information from sources and classify the polarity of expressed sentiments. We have employed Vector Machine Support, Naive Bays, Decision Tree, Random Forest, Logistic Regression, and MLP Classifiers for large-scale supervised education on the Amazon dataset and obtained satisfactory results. In the meantime, the MLP classifier produced the best results. Finally, this paper discusses sentiment analysis and product feedback opinion mining.

**Keywords:** Sentiment Analysis; Opinion Mining; Amazon Review Analysis; ; Support Vector Machine(SVM); Decision tree; Logistic Regression; MLP: Random Forest Analysis

## **Dedication (Optional)**

To our parents, we dedicate our thesis. Because of their unending support, love, and encouragement, we are inspired. Above all, we owe our parents' love and support to them. We will continue to draw from both of you for the strength and courage you both gave me as we strive to reach for the heights and chase our ambitions. Thank our sister and brother too.

## **Acknowledgement**

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption. Secondly, to our Advisor Mr. Faisal Bin Ashraf sir for his kind support and advice in our work. We really want to thank him for being an incredible advisor and an incredible listener.

# Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x
Nomenclature	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Statement . . . . .	2
<b>2 Related Work</b>	<b>3</b>
2.1 Literature Review . . . . .	3
<b>3 Methodology</b>	<b>17</b>
3.1 Our Working Approach . . . . .	17
3.2 ALGORITHM FOR OUR PROPOSED APPROACH . . . . .	19
3.3 Dataset Description . . . . .	19
3.4 Data Preprocessing . . . . .	20
3.5 FEATURE EXTRACTION . . . . .	21
3.6 Support Vector Machine (SVM) . . . . .	23
3.7 Naïve Bayes . . . . .	25
3.8 Linear regression . . . . .	26
3.9 Decision Tree . . . . .	27
3.10 Random Forest . . . . .	28
3.11 Neural Network . . . . .	30



<b>4</b>	<b>Evaluation and Performance Analysis</b>	<b>32</b>
4.1	Performance Metrics . . . . .	32
4.2	Result Analysis . . . . .	33
4.3	Comparison of Accuracy Between All Model . . . . .	37
4.4	Comparative Analysis . . . . .	38
<b>5</b>	<b>Conclusion and Future Work</b>	<b>40</b>
	<b>Bibliography</b>	<b>42</b>

# List of Figures

3.1	Amount of Positive and Negative Reviews Before Balancing . . . . .	17
3.2	Amount of Positive and Negative Reviews After Balancing . . . . .	18
3.3	Workflow . . . . .	18
4.1	Experimental result for TF-IDF (In %) . . . . .	34
4.2	Clustered Bar Diagram for Accuracy of TF-IDF . . . . .	34
4.3	Clustered Bar Diagram for Accuracy of BOW . . . . .	35
4.4	Experimental result for BOW (In %) . . . . .	35
4.5	Clustered Bar Diagram for Accuracy of W2V . . . . .	36
4.6	Experimental result for W2V (In %) . . . . .	36
4.7	Comparison Between Precision and Recall . . . . .	38

# List of Tables

2.1	Comparison among accuracy based on literature reviews . . . . .	16
4.1	Experimental result for all models . . . . .	37
4.2	Previous Works on Electronic Data Experiment . . . . .	39

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$\epsilon$  Epsilon

$v$  Upsilon

(FN False Negative

BOW Bag of Wordl

FP False Positiv

MLP Multilayer Perceptron

SVM Support Vector Machine

Tf-IDF Term frequency–Inverse document frequency

TN True Negative

TP True positive

W2v Word2vector

# Chapter 1

## Introduction

### 1.1 Background

Sentiment analysis is a method for determining whether the information is positive, negative, or neutral. Additionally, it is referred to as opinion mining and emotion AI. Sentiment analysis enables the rapid extraction of insights from vast volumes of text data. It is primarily a classification algorithm that seeks to classify an opinionated point of view and its temperament while also highlighting pertinent information [6]. Sentiment analysis analyzes customer sentiment using natural language processing, text analysis, and statistics. Trying to assess their customers' emotions, such as what they say, how they say it, and what they mean, is one of the most important business practices. Customer sentiment can be gleaned from tweets, comments, reviews, and other areas where people mention your brand. Sentiment analysis is the study of these emotions through the use of the software. It is much needed for the developers and business leaders in a modern workplace. Deep learning, like many other areas, has elevated sentiment analysis to the forefront of cutting-edge algorithms. Product reviews are one of the well-documented applications of sentiment analysis. Moreover, today, the internet pervades the globe. The internet is widely used today. We use the internet on a regular basis for a variety of purposes. Furthermore, purchasing products via the internet is not uncommon these days. Because of the internet, people's purchasing habits have shifted drastically. The online marketplace has taken over the world due to widespread marketing advertisements, and as a result, new products are being introduced regularly. So, consumers have no other option but to rely on product feedback to help them decide. Even though browsing online and comparing individual reviews can be confusing for consumers, we need numerical ratings and reviews to help them make an informed buying decision. Customers tend to find the most valuable reviews as soon as possible in order to make their decision. As a result, reviewers' ability to predict the user rating based on the review is critical. By gaining a general sense of the text analysis, you will potentially enhance the customer experience. It benefits both the company and the seller by increasing sales and popularity and encouraging sellers to develop their product by knowing its needs constantly. Both parties profit from this arrangement. Indeed, opinion analysis aids in determining the most critical problems. Due to the automated nature of sentiment analysis, decisions can be made based on a vast volume of data rather than gut feeling, which is not always reliable. This paper aims to recognize positive and negative customer feedback of various products and build

a supervised learning model for large polarizing amounts of data. Our dataset includes customer feedback and ratings, which we received from Amazon's Consumer Reviews. Following the extraction of features from our dataset, we used them to construct several supervised models. We used both traditional algorithms such as naive Bayes and linear supporting vector machine and deep learning metrics such as multilayer perceptron (MLP).

## 1.2 Problem Statement

Conventional sentiment analysis looks at the intensity of feelings, emotions, urgency and motives while models consider positive, negative and neutral emotions. By working with questions and opinions, and feedback, the system can track specific words and concepts to provide detailed info for those interested in such analysis. In some instances, before making a purchase, customers will use sentiment analysis to research a product or service. Marketers will use surveys to find out what the public's ideas and tastes are to determine how well their products are doing and how much or how little consumers want them. Corporations and marketing agencies used the results of this methodology to gather may also be used by businesses and firms in collecting valuable input on newly released products. Sorting out the emotion associated with ratings, social media discussions, and processing these quicker and more accurately expands the choices for making more specific and precise decisions. Without an extensive collection of Sentiment Analysis, the current methods are completely ineffective, inaccurate, and unable to perform sentiment analysis due to inadequate data, labeled information, and the inherent complexity of complex sentences.

# Chapter 2

## Related Work

### 2.1 Literature Review

Sentiment analysis is frequently pointed to opinion mining because it utilizes consumer reviews to ascertain an item's ranking. It belongs to the artificial intelligence sector. Since online knowledge is rapidly expanding, it is regarded as critical in the current situation, as a large number of obstinate client writings are available on the web now. Sentiment analysis can be described as the examination of a customer's thoughts and emotions. As the volume of data rises, the importance of sentiment analysis and opinion mining increases proportionately. Machines must be dependable and practical in order to interpret and comprehend human emotions and feelings. We recently studied some Sentiment Analysis research papers and discovered a wealth of knowledge.

A thesis in which the researcher applied sentiment analysis to hotel reviews. As we can see, internet use is steadily growing. Before making an online booking, people research the location or product in question. Online marketing is highly beneficial in the field of tourism hotel promotion. Tourists and locals often share their hotel experiences through online reviews and hotel review books. As a result, numerous hotels can be found online. Due to the abundance of reviews on the internet, it is exceedingly difficult for the visitor to determine the best one or comprehend all of the reviews. Tourists cannot comprehend any of the reviews they read, regardless of whether they contain positive or negative opinions. It would be easier to detect positive or negative reviews if we use sentiment analysis. 93 percent of hotel managers believe that online tourist reviews are vital to their property's potential success [19]. The authors of this study depend on hotel guest reviews. The author obtained the dataset for this study from an English hotel analysis, which was preprocessed first. They will focus their efforts on both positive and negative feedback. After that, perform the manual marking. They reread the sentence and assign a value of 1 (positive) and 0 (negative) to each term (negative). Manual marking results in 3946 sentences being labeled 1 and 1053 sentences being labeled 0 [19]. They shared data using the k-folds cross-validation technique. They employ a variety of data preprocessing techniques in this research paper, including dataset dimension reduction, case folding, punctuation removal, stop word elimination, lemmatization, and tokenization. Dimensional reduction is the process of determining which dimensions are essential for their study. They delete stop words that are superfluous to the

dataset. They employ bag of words techniques to extract features. Bag of words collects the text-based data and then generates a list of all the vocabulary contained in the dataset. Additionally, they used the feature selection process. The feature selection process begins by selecting features for training data and then restricting the text classification to a few features to simplify the model building process. This is a more effective classifier since the vocabulary is reduced. It can increase accuracy by removing the noise function. They conducted their research using the naive Bayes form. This technique is based on multinomial distributions. They often employ K-fold cross-validation as a classifier. If we look at the result section, we can see that preprocessing can boost system efficiency. It has been shown that using tokenization alone results in an average F1-Score of 90.2 percent, 90.7 percent, and 90.9 percent lower than preprocessing, which results in an average F1-Score of 90.9 percent, 91.4 percent, and 91.2 percent [19]. The method can produce an optimal result for classifying positive and negative sentiment orientations, with the optimal result being an F1-Score average of 91.4 percent when preprocessing classification is used. It outperformed the bag of words technique.

With the help of sentiment analysis Natural Language processing (NLP) plays a significant role in extracting information. Sentiment analysis is a burgeoning research area in the field of Natural Language Processing (NLP). It has received considerable attention in recent years. As seen, sentiment classification is used to check or evaluate any user comment to obtain the user's opinion. Sentiment analysis is a type of machine learning. Here, computers identify and interpret human sentiments, feelings, and views. They share their feelings by star ratings; thumbs up or thumbs down. This study's data was gathered from online product reviews on a sample website that we developed. To identify such phrases, a negative phrase recognition algorithm is used. Evaluation tests are used to assess results. Finally, we discuss our potential work in sentiment analysis [17]. They use a review of the product based on the customer's feedback in the article. There were both favorable and unfavorable remarks. Since the comments were attached to a phrase, they used the Parts Of Speech Tagging (POST) technique. The Negation Phrase Identification algorithm was used to identify and process several sentences containing negative expressions. As they build a website, customers can leave reviews in both star rating and text format. The star count is classified into three categories: decent, bad, and excellent. The count can range from one to three. Count 1 corresponds to poor, 2 to nice, and 3 to excellent. They have four different forms of ratings: star, thumbs-up and thumbs-down, textual, and emoji. They use two files containing positive and negative terms extracted from the dictionary for function labeling. The resulting collection of pre-processing operations is compared to the original files. We discovered that they coded favorable terms with a '0' and negative words with a '1'. They used K-means clustering to divide the collected dataset into a predetermined number of clusters. The conclusion they reached after clustering positive and negative terms. Their dataset includes online product reviews, as well as the binary emotion polarity labels associated with them. They earned 90 percent for RECALL, 87 percent for Precision, and 90.47 percent for Accuracy following performance assessment [17]. They could obtain a more accurate result if they used the neutral result in their dataset.



In this paper, the researcher survey on product reviews sentiment analysis. The internet's growth is accelerating. Individuals use the internet before purchasing a product. They attempt to glean information about the product from the review section. Millions of product reviews are now available on the internet. Consumers often seek quality information from online customer reviews before making a purchase decision, and many businesses rely on online reviews for product creation, marketing, and customer relationship management [7]. Both customers and businesses profit from this type of knowledge. When reviews are given in text format, it can be challenging to locate the mining. Consumer reviews of products, in particular, are provided in text format; they first parse the reviews using a Natural Language Processor to define the characteristics of the product and then use a sentiment classifier such as Naive Bays or SVM to classify the comments as positive or negative sentiments [7]. They attempt to apply the product aspect rating system in this paper. It automatically extracts critical information about goods from online customer feedback. Consumer reviews are given in text format for each product. Thus, they begin by parsing the feedback using a natural language processor to ascertain the characteristics of a particular product. They also create an aspect ranking for their paper, beginning with aspect recognition, followed by sentiment classification, and finally, product aspect ranking. They evaluate the review to ascertain market sentiment. They employ the NLP Stanford parser technique for product aspect recognition. They attempt to extract frequent noun phrases. They deal with positive and negative feedback in the sentiment classification section. They classified positive reviews as positive and negative reviews as unfavorable, classified as positive and negative opinions. They train the sentiment classifier using the feedback as a sample. They used supervised learning to mark the reviews in order to develop an extraction model. This is referred to as an extractor model. They measure the weight of a product in the product aspect ranking section based on the consumer feedback provided by the customer. This algorithm makes use of TFIDF techniques to determine the weight of a term paper. The weight of an aspect is determined by combining the aspect value provided by the TFIDF with the occurrence frequency of positively and negatively opinionated terms associated with the aspect expression. Since they did not find a frequency-based approach in their paper, they were unable to define the genuinely critical aspects of the goods, which resulted in a decline in the review's efficiency and a lower score.

Another research was conducted on Sentiment Analysis of User Reviews of Mobile Applications, in which the author analyzed over 140,000 actual user reviews of mobile applications. The world is being increasingly digitalized daily. A smartphone is an integral part of everyday life. The usage of cell phones is increasingly growing. As a result, understanding customers' feelings and wishes is becoming increasingly relevant. It is necessary, therefore, to think about their motivations. The customer left a review on the internet in the form of textual comments. The analysis was obtained from a mobile WeChat customer. They assign a point value to the analysis ranging from one to five. They combine 4 to 5 points to create a favorable review, 1 to 2 points to create a negative review, and 3 points to create an unbiased review. There are 145,263 experimental data sets, including 109,901 positive example data sets, 23,654 negative example data sets, and 11,688 neutral example data sets, which account for 75.67 percent, 16.28 percent, and 8.05 percent, respectively [5]. They

discovered a study that was significantly shorter than the PC reviews. According to our experiment results, the most superficial analysis contains only one Chinese character, while the most complex contains more than 6000 [5]. They began by pre-processing the dataset into a feature vector representation. The text is dense with internet slang, typographical mistakes, and colloquial terms used in mobile application reviews, making it impossible for the traditional Chinese Word Segmentation method to segment the words or phrases correctly. There may be two opposed polar sentences. This form of sentence can be synonymous with the word reflect. To circumvent this problem, the researcher employed the N-gram technique. One article was so long that it was divided into sections such as 1+, 10+, 20+, 100+, and 300+. Here, 1+ was used to describe the study of multiple Chinese words. The dataset was then categorized using classification algorithms. They segment the data into three distinct sub-processes. To classify sentiments, the author used the Support Vector Machine and Naive Bayes methods. They demonstrate their finding by comparing these two classifier methods. For positive classification, the LibLinear approach was found to be more accurate. On the other hand, Naive Bayes was more successful at classifying objects as negative or neutral. Both methods are nearly identical in terms of accuracy, but Naive Bayes performs better for multinomial classification than the AUC index's linear approach. We saw in the classification algorithm link that the Bayesian method outperformed the SVM method. However, in the feature representation section, N-Gram (N=2) is used to achieve the best result following the Chinese word segmentation. They intend to continue investigating a comprehensive solution for mobile application review classifications based on the characteristics listed in the paper.

The researcher demonstrated some results using the sentiment analysis technique based on user reviews of products using the Naive Bayes technique in this article. Individuals can find hundreds to thousands of reviews for a single product. As a result, it is challenging to determine whether a product is beneficial to them or not. Extracting details from feedback based on their sentiment is essential not just for the customer but also for the user. A customer may make a critical decision about a product based on the analysis. That is why the researcher attempts to obtain a result by using an algorithm. The researcher begins this research by preprocessing the data to make it more manageable. It enables them to use them throughout the mining process. They perform case folding, tokenization, stop word elimination, and stemming during the preprocessing stage. They employed case-folding to convert each word in a sentence to a lowercase. It assists them in cutting a sequence of characters from a given collection of documents into pieces of text that conform to the requirements structure they desire. They omit the stop word because it is superfluous to their dataset. They benefit from POS tagging because it enables them to have a tag for each word. In this study, POS tagging was performed using a Stanford CoreNLP library [12]. They used the stemming approach as the final stage of data preprocessing. They reduce each word to its stem or root form. As we can see, the researcher used the Porter Stemming algorithm, which is the most widely used and popular stemming algorithm in the world, to stem the English language terms. Following that, the authors perform the feature selection process. This section is critical to the analysis. They used this approach to retrieve all of the classification words. They conducted their research using two bag-of-words. One is used for a

group of words that contain the aspects, while the second is used for a group of words that contain an emotion polarity tendency. The words with the tags JJ, JJR, JJS, RB, RBR, and RBS were classified as having feelings, while the words with the tags NN, NNS, NP, and NPS were classified as having aspects [12]. They analyzed their dataset using the Naive Bayes algorithm. They classified two factors, dubbed aspects and sentiment. Both variables have an impact on the way words are used in sentences. They reached this conclusion based on the polarity of some aspects' sentiments. It was accomplished by counting the positive, negative, conflict, and neutral polarities. They collected 3618 reviews for their training dataset and grouped them into five categories (food, price, service, ambiance, miscellaneous). There are only 96 reviews in the test collection. If we look at the results section, we can see that they received the highest positive score for various factors and the lowest negative score for food, which was 8. They obtained the same result for all facets of the dispute. They used Chi-Square to minimize the amount of data needed for the Naive Bayes model construction. 0.2, 0.1, 0.075, 0.05, and 0.01 are the values. These values were chosen because they are often used in the Chi-Square feature selection method [12]. The classification result indicated that the accuracy was greatest when Chi-Square was omitted. Once the reviews have been examined, it is obvious that the Naive Bayes classifier performed admirably for aspect-based sentiment analysis, with the best F1-Measure of 78.12 percent. However, the best F1-Measure for aspect classification was 88.13 percent, and for emotion classification, the best F1-Measure was 75 percent. The Chi-Square method demonstrates that it can significantly reduce the computation time required for the Naive Bayes algorithm's classification procedure.

Another research considered the effects of Natural Language Processing (NLP) and linguistic preprocessing on the accuracy of extensive data analysis. To begin, a thorough act analysis of the method was performed on 12 network-connected machines. One of those machines was designated as a NameNode, another as a Secondary NameNode, and the remaining machines were designated as DataNodes. However, while the DataNodes were equipped with eight virtual CPUs and eight gigabytes of RAM, the NameNode and Secondary DataNode were equipped with twenty-four virtual CPUs and sixteen gigabytes of RAM. All machines ran Hadoop version 2.6.2.0 under the CentOS 7 operating system. The sentiment analysis technique was implemented in MapReduce using the Mahout Library adaptation 2.6.2.0 and a Java Compiler of version 1.8 [16]. Then, for testing purposes, a real-world data set collected from Twitter was used to implement an algorithm properly.

Twitter is a powerful platform for users of social networking to post tweets expressing their opinions on a specific topic. The Stanford Twitter Sentiment data collection was used in this analysis. For training purposes, it suppressed 1600 000 tweets and divided them into optimistic and negative emotions. Six fields were included in each row: the polarity of the tweet (0 for negative and 4 for positive), the id, the date, the flag, the user, and the tweet's content. The hazard was limited to the first and last fields in this investigation. The test data set included 498 records classified as "positive," "negative," and "neutral." Additionally, only "positive" and "negative" records were selected from the test dataset, which consisted of 359 records divided into 177 negative tweets and the remaining positive tweets. They used the POS tagging method to give adjectives a higher weight in Twitter than other words. As their Naive Bayes algorithm employs TF-IDF to calculate term weights, they increased

the weight of the adjective by escalating the term frequency. The following tweet illustrates the impact of weighted Pos tags on classification accuracy: "My dentist appointment today was very enjoyable," while this tweet is addressed as "positive," the Nave Bayes classifier classified it as "negative," since the terms "dentist" and "quiet" are synonymous with the negative category in the Naive Bayes system. However, by employing Pos tagging and assigning a higher weight to the adjective "enjoyable," the Nave Bayes system graded this tweet as having a positive polarity. This article demonstrates that, in order to ensure excellent efficiency for Nave Bayes using Pos tagging, the weight of the adjectives should be increased to four. Despite the increased running time, the introduction of NLP and linguistic preprocessing into the classification process resulted in an increase in classification accuracy . Additionally, this article demonstrates that the highest accuracy for sentiment analysis classification was obtained by removing URLs, stop words, other users' mentions, and hash tags (via linguistic and natural language processing preprocessing) and increasing the weight of adjective in the sentiment classification, which was 71.0 percent and 72.7 percent, respectively, for Nave Bayes and Pos tagging. Following that, Table 3 compares the accuracy of the two methods, with and without linguistic preprocessing and lemmatization [16]. The experiment indicates that after removing numbers, hashtags, URLs, and user mentions (via linguistic and natural language processing pre-processing) and lemmatization, the accuracy improved marginally. At the same time, lemmatization imposed a significant overhead on the execution time.

Nowadays, research into sentiment analysis and opinion mining is critical. Most companies generate a variety of data, which they must evaluate to make critical decisions that support the industry. Additionally, social media generates a massive amount of data, which must be analyzed and significant perceptions gleaned from it. The dataset was compiled for this article from product reviews on the Amazon website. Additionally, this dataset contains six product reviews about cameras, portable workstations, mobile phones, tablets, televisions, and video surveillance [20]. Each product evaluation was sentimentally analyzed and then categorized using machine learning techniques such as Naive Bayes and SVM. As previously mentioned, the dataset was gathered from Amazon and is in JSON format. After that, we preprocessed the dataset by performing tokenization, stop word elimination, stemming, and punctuation mark removal. Preprocessing is critical for sentiment analysis and opinion mining. Following that, we used the bag-of-words model to analyze the text results. The bag-of-words demonstration can be used to simplify natural language processing and data recovery. A text is constructed using this method as the bag of its words, disregarding linguistic use and even word order but retaining multiplicity [20]. As a result of using bag-of-words, the occurrence of the same word will be counted here as well. Following the completion of the bag-of-words, we evaluated and measured the sentiment score for each sentence. To measure sentiment ranking, the dataset was associated with opinion lexicons. Then, 2006 positive and 4783 negative terms were identified, and a sentiment score for each sentence was measured. To identify the sentiment, we used the previously calculated sentiment score in conjunction with a variety of algorithms such as Naive Bayes and Support Vector Machine (SVM). We experimented with accuracy, precision, and F scores immediately after completing sentiment classification using both Naive Bayes and

SVM [20]. As shown in Table 2, the accuracy, precision, and F score for Nave Bayes are significantly higher than those for the Support Vector Machine (SVM) for all six product reviews. For example, the Naive Bayes classifier achieved 98.17 percent accuracy for Camera reviews, while the SVM classifier achieved 93.54 percent accuracy. In this proposed strategy, machine learning methods were used to combine a dictionary-based approach with a lexicon-based approach. If we use aspect-level sentiment analysis on the same dataset in the future, the results will change, and we will be able to determine exactly what individuals favored or despised. For instance, the camera’s quality, megapixels, image size, structure, lens, and image quality. An aspect level approach is a fine-grained approach that produces excellent results in Sentiment Analysis [20].

E-commerce websites are steadily gaining popularity in the world. Individuals visit them not only to shop for items but also to understand the mindsets of other customers and users. The reviews that online consumers submit to the internet assist new buyers in purchasing the correct product and also assist businesses in improving their business strategies in order to compete in this dynamic global market. We have developed a prototype web-based framework for recommending and comparing products marketed online in this paper [8]. Natural language processing was used to automatically interpret the reviews and determine their polarity in this instance. We used the Naive Bayes classification method. Additionally, we removed feedback of product features and their polarity. We visually demonstrate to the client which product is superior based on a variety of metrics, including star ratings, review date, supportiveness ranking, and polarity of feedback. However, when it came to selecting an e-commerce platform, we chose Flipcart over Amazon because Flipcart is ranked higher in the Indian context. Additionally, the number of reviews on several items on Flipcart is higher than on Amazon. Since mobile phones are one of the most viewed and sold items on Flipcart and have the most feedback and ratings, we chose mobile phones as the product [8]. Now, we will look at the technique that was endorsed for performing opinion mining on customer reviews of mobile phones on Flipkart in detail. However, we focused solely on measures that included numerical instruments and calculations. To begin, we extracted the star ratings, review date, review material, and helpfulness score for each review from Flipkart. There are many resources available for automatically extracting feedback from websites, including Visual Web Ripper, Scrapy, Handy Extractor, and Helium Scraper.

Nonetheless, we used our own Python code to extract feedback for this article. We extracted an additional 1039 Samsung Galaxy Duos 2 feedback from Flipkart. Table 1 summarizes the order of the reviews. After separating the 1039 ratings, the text part was extracted for natural language processing. To process the feedback, Part of Speech (Pos) tagging was used. TreeTagger and the Natural Language Tool Kit were used to complete the POS tagging (NLTK). Pos tagging is the method of associating a word in a document with a specific part of the expression. Thus, once all of the text in all reviews has been tagged with the Pos, every phrase will be classified as a part of speech (noun, verb, adverb, or adjective) [8]. However, a single review can provide both positive and negative feedback. As a result, each analysis is divided into sentences. Following that, the polarity of the feedback was calculated using a Naive Bayes classifier. However, a single review can provide both positive

and negative feedback. As a result, each analysis is divided into sentences. Following that, the polarity of the feedback was calculated using a Naive Bayes classifier. Then, we attempted to deduce the features based on the feedback. For instance, "Poor screen resolution," "Battery drains too quickly," or "Excellent audio quality" are all examples of frequently written reviews for mobile phones. The screen resolution, battery life, and audio quality are all considered features in this case. This feature-based extraction is enormously beneficial to buyers and sellers who wish to improve their products and marketing procedures [8]. MALLET was used to describe these features. It primarily identifies sentences that contain features unique to mobile phones. These sentences have been extracted and analyzed for positivity and negativity. After completing the search for all of the features shown in figure 3, a graph is created about that feature to count positive and negative reviews. Then, we calculated the score for a product using star ratings, the polarity of the reviews, the age of the review, and the helpfulness score of the review. Finally, we presented a comparison of two products based on their product scores and the polarity of feature feedback to assist customers in selecting the superior product [8]. For future work, we propose providing a summary of feedback for more than two products and automatically ranking products based on the features that customers care about.

Another research demonstrated a sentiment polarity discovery method that utilizes machine learning algorithms to identify the sentiment polarity of Bengali tweets. Sentiment analysis is a field of study concerned with the extraction, identification, and classification of the sentiment content of composed text units. The sentiment is implied by thoughts, including attitudes, emotions, and beliefs. Sentiment polarity identification is a common task in sentiment analysis [14]. There have been numerous studies on sentiment polarity detection in the past, which included sentiment analysis of English text data. However, nearly 230 million Bengali speakers are scattered across the world and share a vast amount of knowledge on the internet every day. As a result, our investigation here is limited to the identification of sentiment polarity in Bengali tweets. To begin, we preprocessed all of the data. We removed immaterial characters from the data throughout preprocessing. Furthermore, this is critical for tweet info, which is notoriously noisy. Then, we removed from the corpus any words that appeared only once as irrelevant features. After completing preprocessing, we performed feature extraction due to the noisy nature of the data. To extract the primary function, we used an N-gram tokenizer, which converts the input tweet to word n-grams such as unigram, bigram, etc. The feature value indicates the frequency in which an n-gram feature occurs in a tweet. Additionally to n-gram features, we incorporated SentiWordnet features. SentiWordnet is a lexical resource for inferencing [14]. The SentiWordNet function value is determined based on the number of polarity words of a particular type found in a tweet. For instance, if there are three different types of sentiment, such as positive, negative, and neutral, we consider three different SentiWordnet features. If there are  $m$  n-gram features, our feature set prevents the addition of more than  $m+3$  features, where 3 represents SentiWordnet features. Each tweet has now been converted to a vector presentation of length  $m+3$ , with the vector labeled with the training tweet's class. Then, we built a model by training a classifier on the labeled vector obtained from the training set and saving it for future use. This model learns how to classify tweet data into three polarity sentiments using positive, negative, or neutral training data. However, during the testing process, an unlabeled tweet was also converted to a function

vector using the procedure described above. However, there are numerous classifiers available in Weka, which was created at the University of Waikato in New Zealand. However, among the algorithms available in Weka, we used only Naive Bayes and SVM classifiers in this paper. The naive Bayes multinomial calculates class probabilities for a given tweet. Essentially, the Naive Bayes Multinomial classifies a test document into the class with the greatest posterior probability. On the other hand, SVM infers a hyperplane that uniquely divides a set of positive examples from a set of negative examples. It employs a non-linear mapping to convert training data to higherdimensional measurements and searches for the hyperplane with the greatest margin between the groups. The central idea behind this algorithm is that a hyperplane with a large margin should be a 'cautious' classification boundary that avoids making prediction errors by being too close to one of the classes' boundaries [14]. Currently, we are supervising a variety of studies on this dataset. Additionally, we performed experiments to demonstrate the adequacy of various n-gram features, including unigram, bigram, trigram, and SentiWordnet features, as well as their combinations. The system's efficiency was determined using the conventional accuracy metric, as detailed in Table 1. As shown in Table 1, the Naive Bayes classifier produces the best result with unigram, bigram, and SentiWordnet features, with a value of 44.20, while SVM produces the best result with unigram and SentiWordnet features, with a value of 45. However, at the conclusion of our paper, we compared our proposed model to three of the top tweet sentiment polarity detection systems that competed in the SAIL 2015 contest, as illustrated in Table 2 [14]. As shown in Table 2, our proposed best model, which incorporates SVM with unigrams and SentiWordnet, achieved a substantially higher level of precision than the other three best models entered in the SAIL2015 contest.

Another experiment was conducted in which they introduced a model for automatically detecting and analyzing fake reviews by quantifying the polarity of user opinions. Sentiment analysis was used to evaluate Amazon reviews, while the Fake Review Detection Framework (FRDF) was used in conjunction with Natural Language Processing techniques to identify and detach fake reviews [23]. First, we extracted the data corpus from Julian McAuley's "Amazon Product Data by Julian McAuley." From May 1996 to July 2014, this dataset includes product reviews and metadata from Amazon. Each review includes a star rating of 1 to 5 and comments from the consumer. All 4181 reviews are for specific items. Previously, reviews were removed because they lacked both vocabulary and an emotion rating. There were 95,737 outstanding reviews in total. The majority of reviews received a five-star rating, while the majority received a four-star sentiment rating. This indicates that the items are usually excellent or that the experience with the product was exceptional [23]. While the initial assessment received five stars, opinion analysis assigned it just three points. Several minor flaws in the product were noted. For example, one user said that "the USB HotSync does not function properly, forcing you to attempt a sync again." USB HotSync is not functioning properly, necessitating another attempt at sync. It is possible to fully erase your data by performing a hard reset (restart your Visor from scratch). Additionally, it is necessary to remember that there are additional problems with the Lexical Affinity Sentiment Analysis methodologies. The Visor initiates the application and prompts you to act. Click on the backup or restore button, and the program will take care of the rest [23].

When the session is complete, you can uninstall the module, and everything will return to normal. To determine the study’s performance, we used the star score as the actual sentiment value. Novielli et al. obtained  $F1 = 0.9$  in the positive class of the stackoverflow corpus using SentiStrength, a lexicon-based classifier close to our process. This means that 89.87 percent of the 52,018 positive reviews are accurate, while 10.13 percent are incorrect. Falsely positive reviews for highly-rated products may have little impact on the decision-making process but will artificially raise or lower the product’s rating. Duplications from distinct user IDs (users) on the same product can signify an attempt to use the most difficult-to-detect form of bogus analysis. After eliminating the erroneous scores, the difference in ranking positions is examined (for example, very bad; bad; regular; good; very good). This analysis incorporates the star rating and sentiment ranking (see Table 10). (Also see Tables 11–14.) As a result of these results, it is clear that false reviews impact the rating of such products and have the potential to influence customer opinion. By eliminating them, we will create more trustworthy content [23]. The higher the threshold, the less reliable the false feedback detection becomes. The most famous examples are duplicated reviews written in various brands using various users. They were indicating a need to provide a fabricated analysis. If a review bears a striking similarity to other reviews and includes more than three obvious words that are similar, it is considered bogus. 85.53 percent of 781 reviews listed as false were confirmed, while 14.47 percent were incorrect. As can be shown, SVM-based approaches work exceptionally well. The best results are obtained when clustering methods and graph-based methods are used (Mukherjee et al., 2012). (Ye Akoglu, 2015). We developed a modular architecture for extracting textual information from user feedback. We used this framework to study an Amazon corpus of software product reviews. The dashboard shows the most affordable brands as well as sentiment values relevant to brand recognition and product costs [23]. For instance, if the percentage of negative feedback is higher for high-priced goods than for low-priced goods, sentiment value could help in the discovery of superior products. This article contains several limitations that could limit the generalizability of the findings. The research methodology was applied exclusively to the e-commerce website Amazon. The sentiment analysis was conducted using the AFINN lexicon. The cosine similarity measure was used to identify fake reviews. While we established a cutoff point for determining if a review is a duplicate, a more significant number of them may have reached different conclusions [23].

Another research evaluated Amazon mobile review data and used sentiment analysis to ascertain the reviewer’s emotion. It enables us to comprehend others’ perspectives on any topic. By using sentiment analysis, we can determine the widespread sentiment or emotion associated with any product. This can help develop the products, as the consumer can quickly determine which products are the right fit. The researcher analyzed an Amazon smartphone analysis dataset for the paper. Nearly 400 thousand reviews were submitted. The dataset contains fields such as the title of the product, the brand, the price, the rating (1-5), the review document, and the number of people who believe the review is helpful [22]. Data visualization is important because it is the representation of any information or data in a graphical format. The training set consisted of 27,799 random reviews, while the research set consisted of 3089 random reviews. They visualize their dataset using Jupyter Note-



book in Python. They illustrate the dataset using various figures. They discovered that the top three brands, Samsung, Blu, and Apple, dominate the total number of reviews. When the terms (CBOW) were removed, the number of reviews plummeted exponentially. They enlisted the assistance of shallow neural networks capable of learning the weights associated with word vector representations [22]. This model converts the analysis text to sentences and generates a list of vocabulary words. They then convert each review to a numerical representation by calculating the average feature vectors for each term. They attempt to suit their feature vectors using supervised learning algorithms after determining the numerical representation of the text data. They began with the multinomial Naive Bayes algorithm. When the Bayes theorem is applied with a clear presumption that a function is always independent, it determines the likelihood of each group. In their case, we discovered that they extracted features using a combination of event counting and function extraction. They classified using multinomial Naive Bayes. They also used Logistic Regression. It is a widely used algorithm. They extracted features using the TF-IDF vectorizer and classified them using logistic regression. Their final classifier was Random Forest. As we know, it is a form of ensemble learning that is used for classification. It generates numerous decision trees during training and outputs the classes associated with each tree. They did this by extracting features using the word2vec model and classifying them using the random forest model. After experimenting with various algorithms, they discovered that logistic regression combined with the TF-IDF vectorizer generated the best results for their dataset. It generated the most true positives and negatives in the confusion matrix. They could improve their findings by using a larger sample size in their training collection. Additionally, they may classify feedback into a greater number of classes rather than a binary classification. In the future, their work will include data analysis for other review sites and a variety of other social media channels.

Another experiment used sentiment analysis to create a mechanism for identifying whether a viewpoint is positive, negative, or neutral. At the moment, sentiment analysis is the most frequently investigated topic. The sentiment analysis result is founded on a thorough study of case studies in politics, crime, economics, universal endeavors, cinema, fashion, and justice. Social media is the primary tool for amassing people's speculations and sentiments about a non-identical trending subject. This investigation paper will focus on supervised learning, a subset of machine learning [21]. The Data Set used in this analysis was airline reviews, which were gathered from the Twitter platform. The dataset, on the other hand, was dense with unstructured airline feedback. Thus, to convert unstructured airline feedback to standardized reviews, we performed preprocessing steps such as stop word removal, @ removal, hashtag removal, POS tagging, and opinion score calculation. Following that, we quantified the organized airline feedback using numerical values. Then, two supervised machine learning algorithms, namely Naive Bayes (NB) and Support Vector Machine (SVM), were used to classify the sentiment analysis correctly, and the results of these two classifiers were compared in the subsequent section of this paper. As previously mentioned, the data source for our research paper is Twitter [21]. We chose Twitter over other social media platforms because it contains only text, while Facebook and Instagram contain images, videos, and numerous web sharing links. We gathered about 10,000 Tweets from the tweeter,

which included both positive and negative feedback of the airline. Due to the fact that Twitter includes multiple hashtags, slang terms, and URL links inside the document, we needed to preprocess the data until assigning numerical values to structured reviews. We completed the @ elimination, hashtag removal, stop word removal, and stemming in the preprocessing section. Text analysis preprocessing is also crucial for qualified models. After preprocessing was complete, we assigned numerical values to all the unique text words. This paper aims to analyze data using supervised and unsupervised learning techniques, which are subsets of the machine learning method. Thus, in this analysis, we used both Naive Bayes and SVM [21]. Naive Bayes is a term that refers to a set of classification algorithms that are based on the Bayes Theorem. When used for text data analysis, such as Natural Language Processing, the Naive Bayes classifier produces impressive results. The naive Bayes algorithm provides us with a probability analysis of the data set we've given. On the other hand, SVM is a supervised learning model that includes associated learning equations for classification and regression analysis. SVM is often referred to as the universal learner. It specified both the input and output formats. The yield may be positive or negative, and the input vector space is used [21]. The text document is not suitable for educational purposes. To perform SVM, these texts are converted to an ordered format. Now, we experimented with both algorithm forecasts and Twitter reviews in order to determine the best performer. Table 1 contains the classification report for both algorithms. If we look at the results of both classifiers in Table 1, we see that SVM is more accurate than Naive Bayes, as SVM achieves nearly 83 percent accuracy in this experiment, while Nave Bayes achieves 77 percent accuracy. Additionally, the precision and recall values for SVM are 90.33 and 81.79, respectively, whereas they are 89.00 and 83.75 for the Naive Bayes classifier. Now, if we examine the model performance using the dataset analysis and predict the yield using both algorithms, we find that the SVM classifier predicts the yield accurately more frequently than not, while Naive Bayes cannot always predict the yield accurately [21].

In another paper, some researchers talked about opinion mining. Opinion mining can be thought of as a categorization process with three levels of categorization: document-level, sentence-level, and aspect-level classification. Document level analysis categorizes the entire document as positive or negative by treating full document as a fundamental information unit. Sentence level analysis seeks to categorize every sentence, which is comparable to document level classification but on a smaller scale of data because a sentence would be a tiny document. Aspect level analysis, on the other hand, seeks to classify items based on certain aspects. [11] In that study they offer an intuitive methodology for the classification of internet news via the study of the sentence level utilizing a dynamic library that facilitates news polarity by defining positive, negative or neutral. They highlight various features that are the major categorization factors of their sentencing recognition and analysis method. Such as Subjectivity and Neutrality, Positive Subjectivity and Negative Subjectivity. In another section, they detail their strategy to capture words and phrases from the newspaper and the libraries design they use to classify news items: detection of sentences, defining library and classifying the news article, extracting words from the sentences, assignment of weights. On windows environment, they used java programming language and Netbeans IDE to program and construct user interface for

the result of their technique to detect the polarity of the news article. They can observe that just 5 out of 56 papers indicate a differential polarization, leading to an error margin of 8.93 percent. Further investigation showed that whenever there are fewer sentences, our methodology occasionally confronts difficulties in detecting the true polarity. Almost all of the news lines have neutral polarity. Only a few phrases decide the polarity of the entire text in case of a short piece. Since the number of short articles in the online news is relatively small and our methodology correctly identifies polarity for the average size of articles, the methodology is good enough to detect the polarity of online news articles. [11]

Title	Year	Dataset	Feature	Accuracy
1.Sentiment Analysis on Hotel Review using Multinomial Naïve Bayes Classifier[19]	2018	Hotel Reviews	Bag of Word	91.4%
2.Sentiment Analysis on Online Product Review[17]	2017	Online Product Reviews		90.47%
3.Survey on Product Review Sentiment Analysis with Aspect Ranking[7]	2015	Online Consumer Reviews	Aspect Ranking Algorithm	
4.Sentiment Analysis on Review of Mobile Users[5]	2014	Mobile Users Review of we chat	Feature Extraction(Selection)	96.7%
5.Aspect-based Sentiment Analysis to Review Products Using NB[12]	2017	Product Reviews from SemEvol2014	Bag of Word	88.13%
6.The Effects of NLP on Big Data Analysis: Sentiment Analysis Case Study[16]	2018	Tweets from Twitter	TF-IDF	72.7%
7.Sentiment Analysis on Product Reviews Using Machine Learning Techniques[20]	2018	Products Review from Amazon	Bag of Word	98.17%
8.Recommending Products to Customers Using Opinion Mining of Online Product Reviews and Features[8]	2015	Flip cart Mobile Reviews	Feature Extraction using MALLET	
9.Sentiment Polarity Detection in Bengali tweets Using Multinomial NB and SV machines [14]	2017	Bengali Tweets	N-Gram Tokenizer	
10. A Case Study on Sentiment Analysis and Fake Reviews Detection for Marketing Decision Making[23]	2019	Product Reviews from Amazon		89.87%
11. Sentiment Analysis of Amazon Mobile Reviews	2019	Amazon Mobile Review	CBoW, Word2Vec, TF-IDF	93%
12.Comparison of NB and SVM algorithms, based on Sentiment Analysis Using Review Dataset[21]	2019	Airline Reviews from Twitter		83%
13.Polarity Detection Of Online News Articles Based On Sentence Structure And Dynamic Dictionary[11]	2017	News Article Data		91.07%

Table 2.1: Comparison among accuracy based on literature reviews

# Chapter 3

## Methodology

### 3.1 Our Working Approach

First of all, we collected the data set of product reviews from Amazon. The data set was a labeled data set. The data set was labeled by star rating. Then we converted the star-rating column into binary classification. Binary classification classifies the components of a set into two groups on the premise of a classification rule. First, we added a new column named “sentiment,” which was used to determine which review is positive and negative. We considered four and five stars as positive reviews, so we converted those reviews into one and two stars as negative reviews, so we converted those reviews into 0. Generally, three-star ratings stand with neutral opinion means neither positive nor negative. As we are using binary classification, we had to consider only positive and negative reviews. So we discarded the three-star ratings from our data set and took the remaining reviews. After modifying, we found around 70000 positive reviews and 24000 negative reviews.

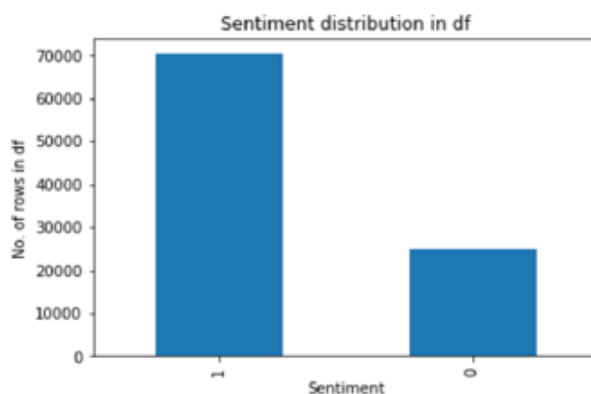


Figure 3.1: Amount of Positive and Negative Reviews Before Balancing

As it is an authentic dataset, the number of rows are not equally distributed for this two sentiments. We are not dealing with problem of imbalanced classes in this research so we balanced the classes by taking 24000 reviews for each classes.

We used several models for the extraction of features. The models are Bag of words (BOW), Term Frequency and Inverse document frequency (TF-IDF) and Word2vec. We used these models individually to build and vectorize features. We then perform

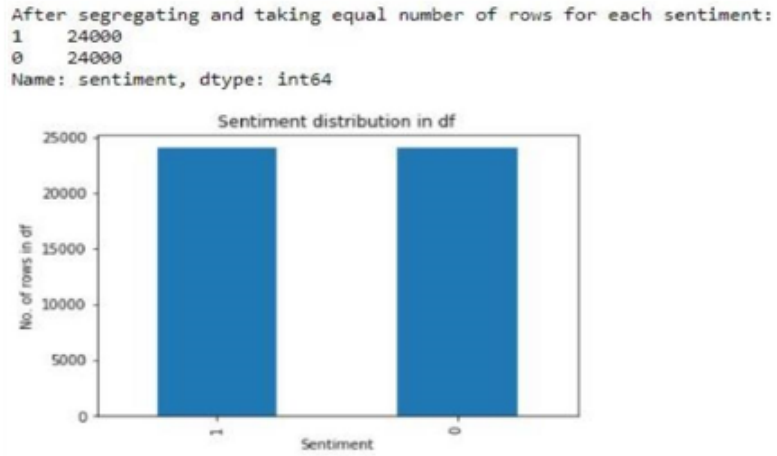


Figure 3.2: Amount of Positive and Negative Reviews After Balancing

them separately by supervised learning to achieve our results. Finally, the outcomes of these models will be compared to assess which model works more accurately. We extracted review text and the overall review extremity from the data used to analyze the review extremity. The following is a schematic of our strategy:

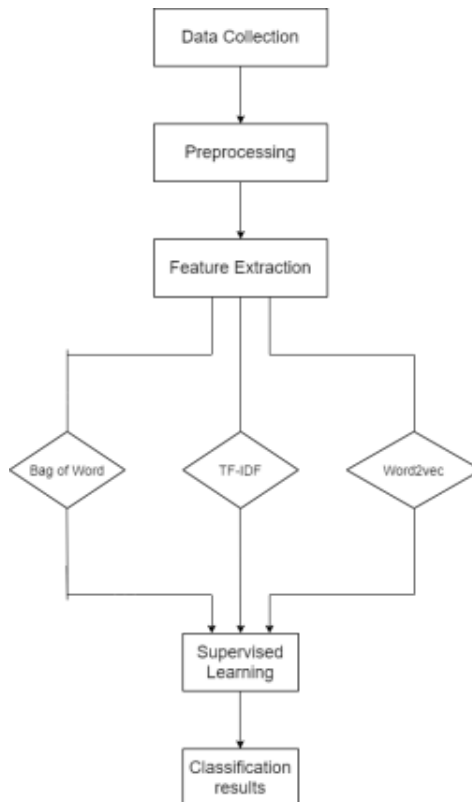


Figure 3.3: Workingflow

## 3.2 ALGORITHM FOR OUR PROPOSED APPROACH

### Input:

labeled data.

### Output:

1. Load the labeled dataset which contains 1 for positive and 0 for negative.
2. Pre-processed the data set.
3. For every  $D=D_1, D_2, \dots, D_n$  in labelled data set.
4. Extract the feature ( $D_n$ ).
5. Split into testing and training set.
6. Classifier. Train( ).
7. Displaying result(accuracy, recall, f1-measure, precision).
8. End.

## 3.3 Dataset Description

As the dependency for buying any kind of product through online platform is increasing day by day, it is also necessary to make those platforms effective and efficient both for the clients and sellers so that, both of the groups get benefits from the platform. For this purpose, we choose Amazon product dataset. Amazon prime is the largest E-commerce site where people can buy and sell their products. The dataset we used was all about mobile electronics product. The dataset was sentence level dataset. There are three level of dataset and they are as follows: Document level, Sentence level, Aspects and Entity level [6].

### Document level

To determine an opinion or views in terms of positive or negative we use document level data. It evaluates the entire record as a bunch of data unit. To generate the sentiment of complete document it is needed.

### Sentence level

Sentence level sentiment analysis is used to determine whether each sentence reflects a positive, negative, or neutral viewpoint about a service or good. This sort is used

to process single-sentence feedback and suggestions from stakeholders.

### **Aspect level**

Aspect-level is a special edition of entity level sentiment analysis. To figure out the senses or polarities towards different entities in the document separately aspect level sentiment analysis is used.

The dataset had 48000 rows of review text with 15 columns. The overview of the data are as followings:

- “Review ID”: Where the id of the review was given.
- “Product ID”: Every product has a different ID.
- “Product title”: Mobile electronics.
- “Star rating”: Opinion of the customers through different stars.
- “Review headline”: Subject of the review.
- “Review body”: Review in details.
- “Review date”: Date of the reviews.

## **3.4 Data Preprocessing**

### **Tokenization**

Tokenization is a process of breaking down long strings of input text into smaller parts. It turns a string or document into tokens (smaller chunks). Tokenization is most commonly used in splitting values like document, sentence, and paragraph into smaller units which can be words or sub words. These smaller units are known as tokens. It is one step in preparing a text for natural language processing. The main benefits of using tokenization are; it makes it easier to map part of speech, matching common words, removing unwanted tokens.

### **Removing Stopwords**

Stopwords are those words that are considered as widespread words to describe less meaning, such as article, pronoun, preposition, etc. In NLP and text mining, stop words are removed generally. Depending on the country, language, and other factors, different stop terms are used in different formats [6]. In a document, there can be many meaningless words. If we do not remove these words, it will be challenging for the classifier to determine the accurate result. It is undeniable that in a document, there will be a vast number of articles and pronouns. If we do not remove this, then the classifier will prioritize these words as they are used many times in the document and will not give an accurate result. So it is necessary to remove these stop words.

### **Stemming**

Stemming is the strategy of diminishing accentuation in words to their root shapes, such as mapping an assemble of words to the same stem. Indeed, the stem itself is not a considerable word inside the language. The method of lessening intonation towards their root shapes is called Stemming; this happens so that portraying a gather of relatable words beneath the same stem, indeed in case the root has no suitable meaning. The reason for stemming is to decrease our lexicon and dimensionality



for Natural language processing assignments and progress speed and effectiveness in data recovery and data handling errands. There are both English and non-English stemmer available for stemming. For English language stemmers, the mainly used stemmers are Porter and Lancaster. Among them, Porter stemmer is the oldest one.

### **Pos Tagging**

The process of labeling the parts of speech for each word in a document is called pos tagging. This task is not straightforward. Pos tag is one of the many disambiguation tasks. Because in the same document, the word can be used on different parts of speech. It may be a handle of changing over a sentence to shapes – list of words, list of tuples (where each tuple is having a frame (word, tag)). In the case of maybe a part-of-speech tag, the tag implies whether the word may be a thing, descriptive word, verb, and so on. Pos tagger is the method which done this work.

## **3.5 FEATURE EXTRACTION**

In real life, all the data we collect are in large entireties. To urge this data, we require a handle. Physically, it isn't conceivable to get ready for them. Here is when the concept of Feature extraction comes in. Assume you wish to work with a couple of the big machine learning experiments or the coolest and well-known spaces, such as deep learning, where you'll utilize pictures to create a wander-on challenge revelation. They are making projects on computer vision where you will work with thousands of curiously extend inside the picture data set. To work with them, you have had to go for a feature extraction strategy, which can make your life simple. Extensive Feature extraction may be a parcel of the dimensionality diminishment handle, in which a starting set of the rough data is isolated and diminished to more sensible bunches. So once you would like to get ready, it will be less requesting. The first fundamental characteristic of these large data sets is that they have a vast number of entities. These entities require a portion of computing resources to plan them. So feature extraction makes a distinction between encouraging the driving highlight from those gigantic data sets by select and combines components into highlights, in this way, reasonably reducing the entirety of data. These highlights are straightforward to get ready but can still delineate the veritable data set with exactness and innovativeness. Many models can be used to create features from a dataset in feature extraction. For example, Bag of words (BOW), Term Frequency and Inverse document frequency (TF-IDF), Word2vec, etc. In our experiment, we are using three separate models for feature extraction.

### **BAG OF WORDS**

For humans, it is easy to understand a word or sentence in a fraction of time seems very easy but for a machine, this task is not as easy to get into main tithe reason behind it is a machine cannot play with raw text data. It is vital for machine to break down the text into a numerical format so that the machine can understand it very quickly. Here comes the concept of Bag of words(BOW), which helps to convert text into a vector format. BOW is one of the convenient models for feature extraction. To implement this model, we create a bag of words or a dictionary to

represent a text or a document. Among all the text, we consider the unique and useful word to build the dictionary. For creating the dictionary, we first POS tag the reviews. Then from the POS tag, we consider only adjective ,adverb and verb to create our desired dictionary. The reason behind considering the only adjective, adverb, and the verb is to make the dictionary more efficient [6]. After that, for each document, we have to count the score of the words. Next, to use the vector for the machine learning model as an input or output, we convert a separate document of free text into a vector. To create the vector, we need to count individual words in every document. The leading scoring methodology is to stamp the closeness of words as a Boolean value, 0 for absent, 1 for present [15]. Then, we will count the frequency of each word to determine how many times the word has occurred in the document or text. With the help of the frequency, we create a matrix where each row represents all the documents; for example, if we define documents in a variable call 'd', the rows will look like d1,d2,d3.....dn where n is a positive integer. On the other hand, each column represents all unique words of all documents; if we define all unique words in a variable call 'w', the columns will look w1,w2,w3.....wm ; where m is a positive integer.

### TF-IDF

The term TF-IDF stands for term frequency (TF) and inverse document frequency (IDF) (IDF). It primarily defines the significance of a word in a document. TF is used to determine the frequency or rapidity of a term in a document, while IDF is used to determine how unusual a term is. The TFIDF value rises in proportion to the number of times a word appears in the database but is offset by the number of current records that contain the word [13]. Rather than referring to a phrase in a record by its rough repeat (number of occurrences) or relative repeat, TF-IDF weights each term by isolating the term repeat by the number of documents within the corpus containing the word [13]. Despite data preprocessing, there might be repeated terms that are irrelevant to sentiment analysis. For example, in a product buy and sell review package, the terms "buy" and "sell" are overused, but these words are useless in sentiment analysis. Each word has a unique TF-IDF score associated with it.

$$\mathbf{tf}(t, d) = \frac{f_d(t)}{\max_{w \in d} f_d(w)}$$

$$\mathbf{idf}(t, D) = \ln \left( \frac{|D|}{|\{d \in D : t \in d\}|} \right)$$

$$\mathbf{tfidf}(t, d, D) = \mathbf{tf}(t, d) \cdot \mathbf{idf}(t, D)$$

$$\mathbf{tfidf}'(t, d, D) = \frac{\mathbf{idf}(t, D)}{|D|} + \mathbf{tfidf}(t, d, D)$$

$f_d(t)$  := frequency of term  $t$  in document  $d$

$D$  := corpus of documents

The weight of a term is calculated by evaluating the score of its TF and IDF product. In addition, this score plays a significant role in determining how much rare or frequent a word is. The more the TF\*IDF score is, the more rare the word is, along with the less TF\*IDF score is, the more frequent the word is in the document [15].

## Word2Vector

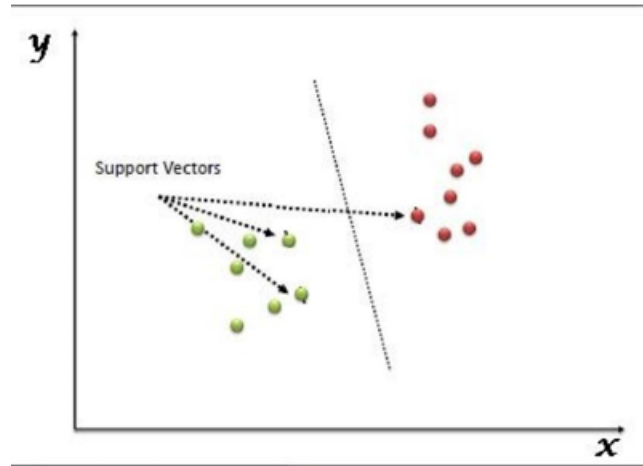
Word2vec is a method for embedding words. The word2vec model takes as input a text corpus and outputs word vectors. It derives the meaning of words by constructing a vocabulary from the training text details. This model efficiently implements the continuous bag of words (CBOW) and skip-grams architectures for computing vector representations of words. CBOW predicts the meaning of a sentence, while Skip-gram predicts the meaning of a word based on its context. The created vectors of words in the corpus always appear spatially in vector space [13]. In our study, we used pre-trained Google News Word2vec embedding with 300 dimensions. If we insert any target word into the pre-trained dataset, it will return a vector representing the words with a high probability of appearing near the given the word. We will pass and train each pair in our entire dataset's w2v (target word, context word). Following preparation, the outcome is a vector output containing the terms highly likely to occur in proximity to the target word if any target word is inserted into the range.

## 3.6 Support Vector Machine (SVM)

A support vector machine (SVM) is used for classification algorithms between two-group classification problems. A support vector machine is actually a supervised machine learning algorithm. There are many kinds of SVM. Like Admin type SVM where the cluster setup processing is automatically creates the admin support vector machine for the cluster. Another is Node type SVM which is created when the node basically join into the cluster. We can see that the node type SVM represents the individual notes of a cluster. We can also see the SVM type like System SVM and Data SVM. In our model we also used the Data SVM algorithm to show our result. We used supervised learning, which entails training a machine learning approach using labeled data. This means that the data I have is already classified correctly. As we can see, one of the most common applications of supervised learning is to assist us in predicting values for new types of data. We discovered that the data we used is not clearly separable linearly in our model for running the SVM algorithms. We are unable to draw a straight line that classifies our results. However, it is trivial to convert to linearly separable data in any higher dimension. We used the linear kernel method to solve this problem. As this approach is the quickest. Due to the fact that our data set contains a large number of features, a linear kernel would perform better on it. The kernel's actual job is to take data as an input and attempt to convert it into the desired format. A Linear Kernel is better than any other Kernel at training the data collection. As we can see, when training an SVM with a Linear Kernel mechanism, only the C Regularization parameter needs to be optimized. However, when practicing with other kernel types, it is necessary to optimize the parameter, which means that performing a grid search would typically take longer than usual. A kernel function can be written mathematically as:

$$\mathbf{K}(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

As can be shown, this is an equation that can be used to calculate the inner products of any new Input vector of (x) with all support vectors in the training results. Following the use of the linear kernel, we attempt to plot each data item as a point in an n-dimensional space, where "n" denotes the number of features. Each type's function has a value that corresponds to a specific coordinate. Following that, we can perform classification by determining the hyperplane that best distinguishes the two classes



We formalize the principle of goodness by introducing a parameter called the margin. The band width is sufficient for determining the boundary or classifier without requiring any training samples. Our objective is to determine the decision boundary with the greatest possible margin. As for the area surrounding the training region, which is essentially the decision boundary, the margin cannot move. As the radius increases, the feasible area shrinks. In 1970, mathematicians Vapnik and Chervonenkis introduced the idea of the VC dimension, in which they attempted to estimate potential research error ( $R(a)$ ) as a function of training error and some VC dimension function.

$$R(\alpha) = R_{train}(\alpha) + \sqrt{\frac{f(h)}{N}}$$

$$f(h) = h + h \log(2N) - h \log(h) - c$$

$$\text{Margin} = \rho$$

$$\text{Relative Margin} = \frac{\rho}{D}$$

$$h \leq \min\left\{d, \left\lceil \frac{D^2}{\rho^2} \right\rceil\right\} + 1$$

As can be shown, the VC dimension, "h," was written as the minimum of the inverse of the relative margin square and for the dimensions of the data. Thus, if we can maximize the relative margin, we can also minimize the inverse square margin. To

do so, we attempt to compute the distance between the line and the support vectors, referred to as the margin. The margin is greatest for the hyperplane dubbed the ideal hyperplane. The SVM model makes an attempt to define the decision boundary. It attempts to make the border as broad as possible between the two classes. Using this type of Transformation, we can restore the linear separator in the higher dimension to the original dimension. This SVM algorithm performs exceptionally well when there is a simple separation of groups. If the number of dimensions exceeds the total number of samples, this algorithm is extremely successful and memory efficient. However, SVM algorithms do have some disadvantages. To begin, the SVM's magic is the classifier's primary disadvantage. We discovered how difficult it is to interpret the dynamic data transformations and resulting boundary plane. This is why it is sometimes referred to as a black box.

### 3.7 Naïve Bayes

Our model distinguishes between positive and negative reviews. We know that written reviews are excellent datasets for performing sentiment analysis because they often provide a score that can be used to train an algorithm that will automate the process. For this purpose, the Nave Bayes algorithm is a very common form of classification algorithm. Any sentiment analysis problem can be solved using a multinomial naive Bayes classification algorithm [3]. However, the Nave Bayes technique's fundamental concept is to determine the probabilities of the classes assigned to the texts; by doing so, we can obtain the joint probabilities of terms and classes. Due to the limited size of our training data collection, we test Naive Bayes in our model. It works well for features that are relatively small. If we use a large number of features, the model may not be accurate since the probability is distributed and may not obey a Gaussian or other distribution algorithm. To begin, we must determine the likelihood of data by class. This means that we must first segment our training data set by class. Additionally, we can build a dictionary object with the class value as the key and a list of all the records as the value. We partitioned the dataset into two parts: a feature matrix and a response or goal vector. The function matrix (X) contains all of the dataset's vectors (rows), each of which contains the value of the dependent features. The number of features is denoted by d, for instance, X= (x1,x2,x2, xd). We can see, the Response or target vector (y) contains the value for each row of the feature matrix's class or category variable. The naive Bayes algorithm makes the assumption that each function or variable in the same class will contribute independently and equally to the outcome. We compute the prior likelihood associated with the specified class labels. We attempt to measure the likelihood for each class using each attribute. Additionally, the Bayes theorem enables the calculation of the posterior likelihood P(c—x) from P,P(x), and P(x—c). The posterior probabilities were calculated using the following formula:

$$P(c|x) = P(x|c)P(c) / P(x)$$

As can be shown, P(c—x) is the posterior probability associated with the class (c, target) defined in the predictor (attributes). P denotes the class's prior likelihood.

$P(x=c)$  is the likelihood, which is the predictor's probability given the class.  $P(x)$  is the predictor's prior likelihood. After separating the data, we summarize it. We determine the mean. The term "mean" refers to the average value, which can be measured as follows:

$$\text{Mean} = \frac{\text{sum}(x)}{n} * \text{count}(x)$$

Where the list of values or column is  $x$ . We measure the number of positive and negative feedback based on the data collection. A similar approach is used by an algorithm to forecast the likelihood of different classes based on various attributes. The naive Bayes algorithm is most often used for text classification and problems with multiple groups. Since our data set contains several groups, it is ideal for this. We measure the training set probabilities using the total number of documents ( $N_{\text{doc}}$ ).

$$P(c) = \frac{N_c}{N_{\text{doc}}}$$

If we look at the formula we will be found that, in the formula  $N_{\text{doc}}$  is the total number of Document (in total how many documents we used) and  $N_c$  is the number of document. From this we get the probabilities of the documents  $P(c)$ . For categorization the text we used the formula:

$$P(w_i|c) = \frac{\text{count}(w_i, c) + 1}{\sum_w \text{count}(w, c) + 1}$$

Finally, in order to determine the likelihood of a given sample for all possible values of the class variable  $c$ , we simply need to determine the output with the highest probability

We refine our classifier to obtain the best results from Naive Bayes algorithms. Additionally, we employ some kind of classifier combination. We disregard the unknown word contained in our data set when applying the Naive Bayes algorithm. If we discover a stop term, we completely disregard it because it has no effect on the overall results. The naive Bayes algorithm is a simple and efficient way to predict the class of a test data set. Additionally, it performs well when used in multiclass prediction. The disadvantage of the Naive Bayes Algorithm is that it is based on the assumption of independent predictors [3]. As shown by real-world examples, obtaining a fully independent set of predictors is nearly impossible.

### 3.8 Linear regression

Linear regression is based on modeling relationships between scalar and explanatory variables. It is a linear approach for modeling relations. In statistics and machine learning, linear regression is expressed as an algorithm. It is primarily used to predict the value of a variable or a point on a graph based on the values of other related variables. There are many types of linear regressions. The most frequently

used techniques are simple linear regression with a single explanatory variable and multiple linear regression with multiple explanatory variables. There are several types of dependent and independent variables in linear regression, including outcome variables, criterion variables, endogenous variables, exogenous variables, and predictor variables. There are three main usage of linear regression. Firstly it is used to identify the strength of predictors. Secondly forecasting effects or finding the impacts of changes linear regression can be used. Thirdly linear regression predicts trend forecasting that means it can get estimated goal.

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

Linear regression formula is also known as slope formula. There is a formula  $Y = a + b \cdot X$  which defines the linear regression equation with a dependent and independent variable. In the formula  $Y$  means the score of dependent variable (uses y-axis),  $b$  is the regression coefficient or the slope line,  $a$  is the  $Y$  intercept and score of the independent variable is  $X$  (uses x-axis). It is a fundamental linear regression equation used to determine the relationship between two independent variables. Linear regression one of the effective classifiers in sentiment analysis. It is largely used to predict any sentiment. It can convert values into positive and negative forms.

### 3.9 Decision Tree

The decision tree is one of the algorithms used in the Supervised Learning Algorithm. It is often used to solve classification and regression problems. It is a graphical representation of the possible outcomes of a given situation. In fact, a decision tree is a tree-structured classifier in which internal nodes define the dataset's features, branches define the decision rules, and each leaf node displays the results. By learning some decision rules from training data, a decision tree is used to construct a training model that assists in determining the value of a target variable. DT is a classification used in data mining that is based on the branch process as a result of decision making for each potential outcome. An established tree contains three types of nodes: root node, internal node, and leaf node. Although associated decision trees are being incrementally built, DT simultaneously breaks down into the smallest subsets. To begin, the root node is referred to as the topmost decision node when it comes to decision-making because it has no incoming or outgoing edges. Internal nodes, on the other hand, have both incoming and outgoing connections, or at least one has existed in this node. Finally, the leaf node lacks outgoing edge representations for decision-making. In a more complicated case, DT can be embedded in decision rules, and the more accurately the model fits when a sine curve is combined with a set of IF-THEN rules and used for decision making. To perform decision-making, the sophistication of tree patterns has an impact on the

result's accuracy. In decision-making, the DT is better suited for classifying than for computing both categorical and numerical data [18].

The classification technique selects DT based on the folder structure of a text. All cleaned documents are counted and assigned ranks during the TFDIF process. Then, words can be classified using the decision tree classifier. Prior to classifying a document and determining the folder to which it belongs, the association between each word in the document must be analyzed. Thus, the decision tree approach can be used to complete the analysis. Each document will be inserted into the Decision tree's nodes and checked to determine if it satisfies the specification or attribute. Finally, in terms of assessment and visualization, the auto classification algorithm will be used to submit the results of documents to their appropriate folders [18].

### 3.10 Random Forest

Random forests is a learning method which is used for classification, regression or any other task that is operated by constructing a line of a decision tree in the training time and gives the output in the class of an average prediction of an individual decision tree. It is an algorithm of supervised learning. As we were working with a large dataset so it gave a good result for our dataset. We know that random forest work well when the dataset is large and the interpretability is not major concern. It works well for the classification and regression problems. When the trees are being grown the random forest algorithm introduces additional randomness into the model. To use the random forest classifier on our dataset, we must first encode it. It will convert each unique value in a column to a number. We randomly selected N records from the dataset. We build a decision tree based on the records. For each value in our data set, we used the encoding. We create a features dataset that is suitable for our classifier. For our dataset, we used many estimators (decision trees). We used the following formula:

We took the sum of the each feature's important value of every tree. The value was divided by the total number of the tree.

We perform a vote for the each of the predicted result which we were found. For the final prediction we used the result which was gives the most vote result. Random forest classifier gives satisfactory result because of the number of decision trees participating. We can see that it can handle the over fitting problem easily. As it is consists of multiple single trees the result were more accurate and gave a good result for our dataset.



$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

- $ni_{sub(j)}$  = the importance of node j
- $w_{sub(j)}$  = weighted number of samples reaching node j
- $C_{sub(j)}$  = the impurity value of node j
- $left(j)$  = child node from left split on node j
- $right(j)$  = child node from right split on node j

We also calculated the importance of each decision tree. For calculating the feature we used

$$fi_i = \frac{\sum_{j: \text{node } j \text{ splits on feature } i} ni_j}{\sum_{k \in \text{all nodes}} ni_k}$$

- $fi_{sub(i)}$  = the importance of feature i
- $ni_{sub(j)}$  = the importance of node j

$$RFfi_i = \frac{\sum_{j \in \text{all trees}} \text{norm}fi_{ij}}{T}$$

- $RFfi_{sub(i)}$  = the importance of feature i calculated from all trees in the Random Forest model
- $\text{norm}fi_{sub(ij)}$  = the normalized feature importance for i in tree j
- T = total number of trees

## 3.11 Neural Network

A neural network is a massively parallel processor that is predisposed to storing and distributing experiential data. It is similar to the brain in two ways: 1. Knowledge is learned via a learning process through the network. 2. Interneuron attachment strengths, referred to as synaptic weights, are used to preserve details. The technique is decidedly non-mathematical. Let us be less heuristic. Numerous characteristics of neural network models are shared. We provide you with a set of inputs for each model  $x = (x_1, \dots, x_n)$  and some processes that result in the required output  $y = (y_1, \dots, y_m)$ . Our models are predicated on the assumption that the process is implemented using a particular mathematical function, e.g.  $y = G(x)$  for a particular function  $G$ . Working with  $G$  is extremely challenging [1]. Additionally, we cannot predict with certainty that the unknown  $G$  can be determined precisely. We are employing a number of parametrized functions, that is, some inputs  $x$  are associated with "right" outputs  $y = G(x)$ , which we believe will enable us to choose the parameters. The structure is very generic in this case. Indeed, it is still too general in neural networks. Neural network models can be thought of as a subclass of  $F(x, w)$  choices, with distinct rules and regulations and complex procedures for parameter selection [1].

### 3.11.1 Multilayer Perceptron (MLP)

The multilayer feed-forward perceptron (MLP) model is one of the most conceptually appealing neural network models. This is a model with a finite number of layers in its most simple form. Each layer of the MLP Model is composed of a finite number of units. In a neural network, approximation is used (often called neurons). Each unit of the preceding layer (and thus the previous one) is connected to the following layer. These relations are referred to as bonds or synapses in general. The term feedforward refers to the transfer of information from one layer to the next (hence the term). The output layer is composed of the first layer, dubbed the input layer. The intermediate layers are subsequently referred to as the secret layers. Interestingly, the output is accomplished in the final layer, not the output layer. This model is governed by the following statutes and regulations: 1. The  $j$ th unit of the input layer outputs the (input) value  $X_{0j}$ . 2. The  $k$ th unit of the  $i$ th layer receives the output  $x_{ij}$  from the  $j$ th unit of the  $(i-1)$ st layer. After multiplying the values of  $x$  by certain constants (referred to as weights), the products are applied. 3. The above number is multiplied by a shift  $\theta_{ik}$  (referred to as a threshold or bias) and then by a fixed mapping (referred to as an activation function). The resulting value corresponds to the output  $X_{i+1,k}$  of the  $i$ th layer's  $k$ th unit, that is,

$$x_{i+1,k} = \sigma \left( \sum_j w_{ikj} x_{ij} - \theta_{ik} \right).$$

Typically, the activation function, the number of layers, and the number of units in each layer are set a priori for some reason. The following move is to choose the weights  $w$  and thresholds in some way. These latter values are typically chosen to ensure that the model behaves correctly when presented with a collection of inputs

and associated outputs. (The training set is referred to as this.) Learning or training refers to the process of setting weights and thresholds. The fundamental learning algorithm used in the multilayer feedforward perceptron model is backpropagation. Backpropagation is a variation on the gradient descent algorithm [1]. It is important in both this model and the theory of neural networks. We will not discuss this algorithm in detail or the many numerical difficulties that it entails. Multilayer feedforward perceptron models can be classified according to their number of hidden layers, or layers other than the input and output layers. As is readily apparent, neural network theory has its own jargon.

# Chapter 4

## Evaluation and Performance Analysis

### 4.1 Performance Metrics

To determine the classification's accuracy, evaluation metrics are important. . The precision of a classifier on a given test dataset is the proportion of datasets categorized correctly by the classifier. As the accuracy metric is inconvenient for determining the classifier's efficiency in the field of text mining, we looked for alternative metrics. There are three commonly used measures: F-measure, precision, and recall. However, there are a few things we need to familiarize ourselves with. They provide the following:

True positive (TP): Indicates that the fact is positive and the test predicts positive.

False Positive (FP): This condition occurs when the truth is negative but the test expects a positive result.

False Negative (FN): This term refers to situations in which the truth is positive but the test predicts the opposite.

True Negative (TN): This indicates that the truth is negative and the test expects the truth to be negative.

#### **Precision**

To measure how much exact a classifier works is determined by precision; whether the retrieved documents are correct or incorrect. Less false positives are represented by higher precision value whereas more false positives are represented by lower precision value Precision (P) is the ratio of correctly classified instances to total instances [15]. The representation of precision is:

$$P = \frac{TP}{TP+FP}$$

### Recall

How much sensitive a classifier is, the number of positiveness of returned data is defined by recall. Less false positives are represented by higher recall value whereas more false positives are represented by lower recall value. This is defined as:

$$R = \frac{TP}{TP+FN}$$

### F-measure

By combining precision and recall, a single metric called the F-measure is created. This metric is the harmonic mean of precision and recall. The F-measure is described as follows:

$$F = \frac{2P.R}{P+R}$$

### Accuracy

To predict how frequently the classifier generates the accurate prediction is defined by accuracy. Accuracy is defined as the ratio of correct predictions to total predictions.

$$Accuracy = \frac{Correct\ Prediction}{Total\ data\ points}$$

## 4.2 Result Analysis

To estimate performance levels, performance evaluation metrics are important in classifying performance evaluation. As a result, the accuracy metric is the most widely used. The percentage of test datasets classified correctly by measures indicates the classifier's accuracy on that dataset. The classifiers were applied to a number of feature selection processes, with the TF-IDF, Bag of words and Word2vec features achieving the best performance across all datasets.

In our experiments there were many algorithms used for machine learning, such as Logistic regression, Support Vector (SVM), Naïve Bayes, Decision Tree and Random Forest for TF-IDF. On two categories of product review, we used cross validation approaches and the best classifiers to see what the outcomes were based on the evaluation measures.

Classifier	Accuracy	Precision for 0	Precision for 1	Recall for 0	Recall for 1	F1 score For 0	F1 score For 1	Precision (AVG)	Recall (AVG)	F1 Score (AVG)
Support Vector machine	90	89	91	91	88	90	90	90	90	90
Naïve Bayes	87	84	90	91	82	87	86	87	87	87
Decision Tree	84	80	81	82	80	81	81	81	81	81
Random Forest	86	82	90	91	81	87	85	86	86	86
Logistic regression	89	89	90	90	88	89	89	89	89	89
MLP	91	89	90	90	89	91	88	90	90	90

Figure 4.1: Experimental result for TF-IDF (In %)

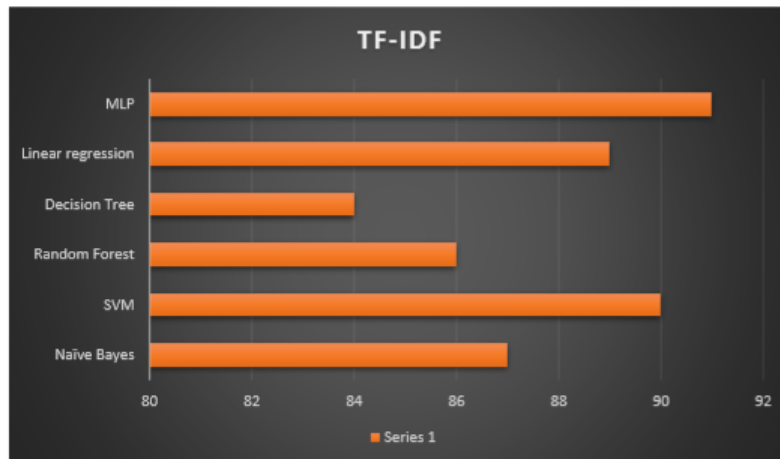


Figure 4.2: Clustered Bar Diagram for Accuracy of TF-IDF

According to the Graph 4.1, it is clearly reflected that, multi-layer perceptron (MLP) comes with higher accuracy which is about 91 percent. We performed different values for the hidden layer to achieve better result. Therefore, using the default value for hidden layer, which is 100, we maximized the accuracy. Then, Second highest accuracy was achieved through the Support Vector Machine classifier where the percentage was 90. Next, Logistic Regression classifier came up with 89 percent accuracy which was third highest followed by the Naive Bayes classifier with 87 percent accuracy. The remaining two classifiers named Random Forest and Decision Tree produced 86 and 84 percent accuracy respectively.

According to the Graph 4.2, the multi-layer perceptron (MLP) has a higher accuracy of about 92 percent for BOW. To get a better result, we experimented with different

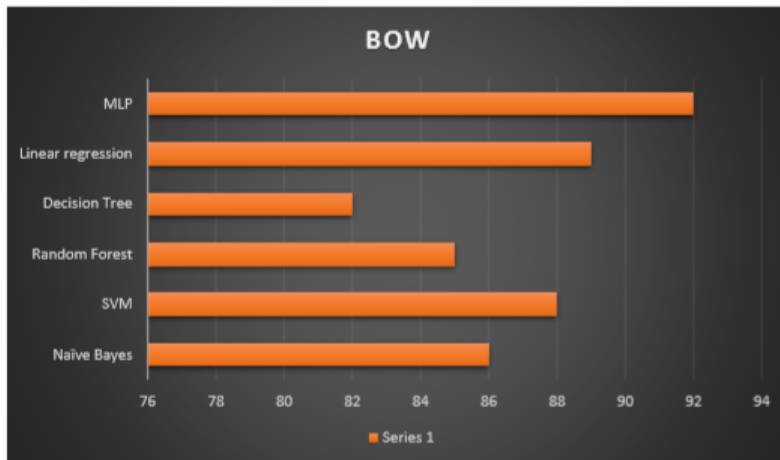


Figure 4.3: Clustered Bar Diagram for Accuracy of BOW

Classifier	Accuracy	Precision for 0	Precision for 1	Recall for 0	Recall for 1	F1 score For 0	F1 score For 1	Precision (AVG)	Recall (AVG)	F1 Score (AVG)
Support Vector machine	88	87	88	88	87	88	88	88	88	88
Naïve Bayes	86	86	87	87	86	87	86	86	86	86
Decision Tree	82	81	80	81	80	82	81	82	82	82
Random Forest	85	86	84	86	83	84	86	85	85	85
Logistic regression	89	90	88	87	91	89	89	89	89	89
MLP	92	91	90	89	92	91	90	92	92	92

Figure 4.4: Experimental result for BOW (In %)

values for the hidden layer. As a result, we increased the accuracy by using the default value of 100 for the hidden layer. The second highest accuracy was attained by the Logistic Regression classifier, which had an accuracy of 89 percent. The Support Vector Machine classifier came in third with an accuracy of 88 percent, followed by the Naive Bayes classifier with an accuracy of 86 percent. Random Forest and Decision Tree, the remaining two classifiers, provided 85 and 82 percent accuracy, respectively.

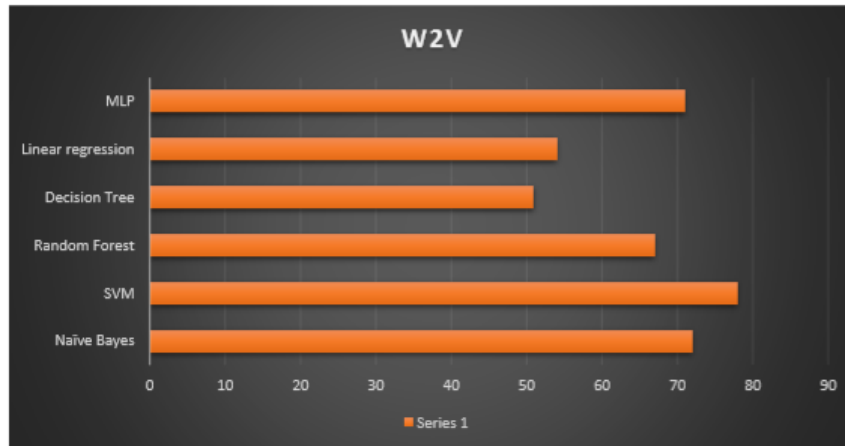


Figure 4.5: Clustered Bar Diagram for Accuracy of W2V

Classifier	Accuracy	Precision for 0	Precision for 1	Recall for 0	Recall for 1	F1 score For 0	F1 score For 1	Precision (AVG)	Recall (AVG)	F1 Score (AVG)
Support Vector machine	68	67	69	61	65	60	65	68	68	68
Naive Bayes	62	63	61	61	62	60	63	62	62	62
Decision Tree	51	49	53	46	56	49	53	51	51	51
Random Forest	67	72	80	71	81	77	75	76	76	76
Logistic regression	54	55	56	55	57	52	58	56	56	56
MLP	71	69	73	66	76	69	73	71	71	71

Figure 4.6: Experimental result for W2V (In %)

The multi-layer perceiver (MLP) is approximately 71 per cent accurate according to graph: 4.3 we tested various values for the secret layer to achieve a better outcome. In consequence, the default value of 100 for the secret layer increases the accuracy. The second highest precision was achieved by the 68 percent accuracy Support Vector Machine Classifier. The classification for random forests ranked third with a



67 percent accuracy, followed by the 62 percent accuracy classification for Naive Bayes. The remaining two classifiers supported 85 percent and 82 percent accuracy respectively named Logistic Regression and Decision Tree.

The multi-layer perceptron (MLP) offered greater accuracy in the dataset, as shown by all of the experiments. Since the working dataset is far larger, the support vector machine performs best for large datasets without over fitting them. The highest accuracy was 92 percent based on these findings, which belongs to the TF-IDF model. On the other hand, The lowest accuracy was 54 percent based on these findings, which belongs to the W2V model.

### 4.3 Comparison of Accuracy Between All Model

We have used supervised machine learning approach for our research. With the help of Support Vector Machine, Naive Bayes, Linear Regression, Decision Tree, Random Forest algorithm we got the results. We also implement multilayer perceptron neural network which is also a supervised learning algorithm.

To estimate performance levels, performance evaluation metrics are important in classifying performance evaluation. As a result, the accuracy metric is the most widely used. The percentage of test datasets classified correctly by measures indicates the classifier’s accuracy on that dataset. The classifiers were applied to a number of feature selection processes, with the TF-IDF, Bag of words and Word2vec features achieving the best performance across all datasets. In TF-IDF we have found an average recall nearly 88 while running the six classifiers.

Classifier	Accuracy For TF-IDF	Accuracy For BOW	Accuracy For W2V
Support Vector Machine	90%	88%	68%
Naïve Bayes	87%	86%	62%
Decision Tree	84%	82%	51%
Random Forest	86%	85%	67%
Logistic Regression	89%	89%	54%
MLP	91%	92%	71%

Table 4.1: Experimental result for all models

On the other hand the precision average was nearly 86 for the classifiers. Again the six classifier gave an average of 89 recall and 87 precision for Bag of Words. However, an average recall of 61 and average precision of 63 were found for the classifiers in Word2Vec.

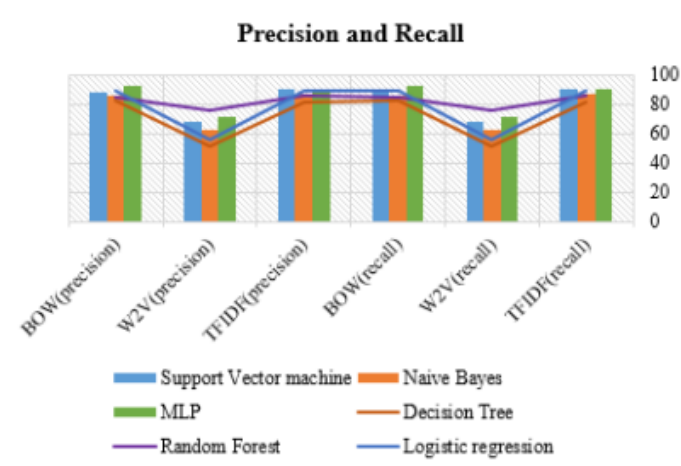


Figure 4.7: Comparison Between Precision and Recall

If we exclude the MLP classifier from the table, it is obvious that the SVM classifier has the highest accuracy of the five classifiers. For the TF-IDF model, the support vector machine (SVM) achieves a 90 percent accuracy. However, the multi-layer perceptron produces the best performance of all six classifiers, at 92 percent (BOW model) in comparison to the other three models. It is the most precise value we discovered. On the other hand, we achieved least accuracy for the W2V model in terms of all applied classifiers as our dataset is not big enough. Moreover, the e-commerce companies release a very limited amount of info about their consumers due to their privacy policy. Hence, with such limited information, Word2Vec will be unable to capture word relationship in the embedding space.

## 4.4 Comparative Analysis

In this section, we attempted to compare our research to other related works. The comparative research was conducted on the basis of precision. The table below provides a comparison. Various research mentioned in the table performed various preprocessing steps and extraction processes. We strive to improve all processes of extraction and pre-processing and to determine the highest possible accuracy in our analysis. Using various preprocessing processes, redundant words have been sorted. Finally, it was possible to achieve greater precision by taking the best features from the datasets and learning by proper classifiers. The table shows that the approaches in our proposed model approaches display more efficiency and better results than any of the existing work.

Table 4.2: Previous Works on Electronic Data Experiment

Paper Title	Publication Year	Dataset	Accuracy
Proposed Model	2021	amazon reviews us Mobile Electronics%	92.00%
Amazon Reviews, business analytics with sentiment analysis [9]	2016	Review of cellphone and accessories	72.95%
Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning [4]	2021	reviews of Kindle	87.33%
Mining comparative opinions from customer reviews for competitive intelligence [2]	2011	Customer product reviews	61.00%
Amazing: A sentiment mining and Retrieval System	2009	E commerce reviews	87.60%

# Chapter 5

## Conclusion and Future Work

We proposed a supervised research model in our work in order to polarize a large volume of unlabeled product review data. Our model is a supervised learning system that combines two different types of feature extractors. We discussed the model's fundamental theory, the methods used in our analysis, and the performance criterion for the experiment we conducted on a large dataset. Additionally, we examined a variety of research papers on sentiment analysis that used a text-based dataset. Additionally, we were able to achieve an accuracy of over 90 percent, a precision of over 90 percent, and a recall of over 90 percent using the F1 scale. We achieved promising results by comparing disparate quantities of data using cross validation, training-testing ratios, and various feature extraction methods. In the majority of cases, the multilayer perceptron neural network model (MLP) produced the best classification results. Due to the fact that e-commerce sites have limitations on data sharing, compiling a large number of standard datasets is challenging. Additionally, scraping data can be difficult because we do not have enough to consider it real-world public input on different products.

Any additional experiments that could be included to improve the model's utility in real-world scenarios. Sentiment analysis techniques have been studied for over a decade and are now being used by companies as a critical component of strategic marketing planning and maneuvering. This change is also due to the advances in data storage, access, and analytics enabled by big data frameworks. On the other hand, big data systems view sentiment analysis as another application that could benefit from advanced data management [10]. While there is a wealth of literature on the difficulties associated with sentiment analysis in big data systems, such as the duration, velocity, and variety problems, the issues of context, veracity, and volatility have received less attention, despite the fact that data taming is important for big data analytics. As a result, the model can be used in conjunction with applications that engage consumers looking for product reviews. Since we used a large dataset, we could use the model on local market websites to enhance accuracy and usability. Finally, we'll continue to work on this until we're confident that this model can be applied to all forms of text-based feedback and comments.

# Bibliography

- [1] P. Allan, “Approximation theory of the mlp model in neural networks [j],” *Acta Numerica*, vol. 8, pp. 143–195, 1999.
- [2] K. Xu, S. S. Liao, J. Li, and Y. Song, “Mining comparative opinions from customer reviews for competitive intelligence,” *Decision support systems*, vol. 50, no. 4, pp. 743–754, 2011.
- [3] V. Muralidharan and V. Sugumaran, “A comparative study of naive bayes classifier and bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis,” *Applied Soft Computing*, vol. 12, no. 8, pp. 2023–2029, 2012.
- [4] C. Rain, “Sentiment analysis in amazon reviews using probabilistic machine learning,” *Swarthmore College*, 2013.
- [5] L. Zhang, K. Hua, H. Wang, G. Qian, and L. Zhang, “Sentiment analysis on reviews of mobile users,” *Procedia Computer Science*, vol. 34, pp. 458–465, 2014.
- [6] X. Fang and J. Zhan, “Sentiment analysis using product review data,” *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, 2015.
- [7] H. Patil and P. Mane, “Survey on product review sentiment analysis with aspect ranking,” *International Journal of Science and Research (IJSR)*, vol. 4, no. 12, 2015.
- [8] P. V. Rajeev and V. S. Rekha, “Recommending products to customers using opinion mining of online product reviews and features,” in *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, IEEE, 2015, pp. 1–5.
- [9] M. S. Elli and Y.-F. Wang, “Amazon reviews, business analytics with sentiment analysis,” *Elwalda, Abdulaziz, et al. “Perceived Derived Attributes of Online Customer Reviews*, 2016.
- [10] N. M. Sharef, H. M. Zin, and S. Nadali, “Overview and future opportunities of sentiment analysis approaches for big data.,” *J. Comput. Sci.*, vol. 12, no. 3, pp. 153–168, 2016.
- [11] M. U. Islam, F. B. Ashraf, A. I. Abir, and M. Mottalib, “Polarity detection of online news articles based on sentence structure and dynamic dictionary,” in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, IEEE, 2017, pp. 1–5.
- [12] M. S. Mubarak, Adiwijaya, and M. D. Aldhi, “Aspect-based sentiment analysis to review products using naive bayes,” in *AIP Conference Proceedings*, AIP Publishing LLC, vol. 1867, 2017, p. 020 060.

- [13] S. M. Rezaeinia, A. Ghodsi, and R. Rahmani, “Improving the accuracy of pre-trained word embeddings for sentiment analysis,” *arXiv preprint arXiv:1711.08609*, 2017.
- [14] K. Sarkar and M. Bhowmick, “Sentiment polarity detection in bengali tweets using multinomial naive bayes and support vector machines,” in *2017 IEEE Calcutta Conference (CALCON)*, IEEE, 2017, pp. 31–36.
- [15] T. U. Haque, N. N. Saber, and F. M. Shah, “Sentiment analysis on large scale amazon product reviews,” in *2018 IEEE international conference on innovative research and development (ICIRD)*, IEEE, 2018, pp. 1–6.
- [16] M. Khader, A. Awajan, and G. Al-Naymat, “The effects of natural language processing on big data analysis: Sentiment analysis case study,” in *2018 International Arab Conference on Information Technology (ACIT)*, IEEE, 2018, pp. 1–7.
- [17] M. V. Mäntylä, D. Graziotin, and M. Kuutila, “The evolution of sentiment analysis—a review of research topics, venues, and top cited papers,” *Computer Science Review*, vol. 27, pp. 16–32, 2018.
- [18] W. Noormanshah, P. Nohuddin, and Z. Zainol, “Document categorization using decision tree: Preliminary study,” *International Journal of Engineering & Technology*, vol. 7, no. 4.34, pp. 437–440, 2018.
- [19] A. A. Farisi, Y. Sibaroni, and S. Al Faraby, “Sentiment analysis on hotel reviews using multinomial naive bayes classifier,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1192, 2019, p. 012024.
- [20] R. S. Jagdale, V. S. Shirsat, and S. N. Deshmukh, “Sentiment analysis on product reviews using machine learning techniques,” in *Cognitive Informatics and Soft Computing*, Springer, 2019, pp. 639–647.
- [21] A. M. Rahat, A. Kahir, and A. K. M. Masum, “Comparison of naive bayes and svm algorithm based on sentiment analysis using review dataset,” in *2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*, IEEE, 2019, pp. 266–270.
- [22] G. Kapil, Z. Ishrat, R. Kumar, A. Agrawal, and R. A. Khan, “Managing multimedia big data: Security and privacy perspective,” in *ICT Systems and Sustainability*, Springer, 2020, pp. 1–12.
- [23] E. Kauffmann, J. Peral, D. Gil, A. Ferrández, R. Sellers, and H. Mora, “A framework for big data analytics in commercial social networks: A case study on sentiment analysis and fake review detection for marketing decision-making,” *Industrial Marketing Management*, vol. 90, pp. 523–537, 2020.