

# Sentiment Analysis on E-commerce Based Product Reviews Using Machine Learning Algorithms

by

Ritu Arobi  
17101330

Rakibul Hassan Rijon  
17301090

Nimul Haque  
17301136

T.I.M. Sakir Safkat  
17201112

Sk Safoan  
17301160

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 and Engineering

Department of Computer Science and Engineering  
BRAC University  
May 2022

© 2022. BRAC University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is 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:



---

Ritu Arobi  
17101330



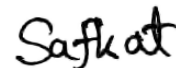
---

Rakibul Hassan Rijon  
17301090



---

Nimul Haque  
17301136



---

T.I.M. Sakir Safkat  
17201112



---

Sk Safoan  
17301160

# Approval

The thesis titled “Sentiment Analysis on E-commerce Based Product Reviews Using Machine Learning Algorithms” submitted by

1. Ritu Arobi (17101330)
2. Rakibul Hassan Rijon (17301090)
3. Nimul Haque (17301136)
4. T.I.M. Sakir Safkat (17201112)
5. Sk Safoan (17301160)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on May 19, 2022.

## Examining Committee:

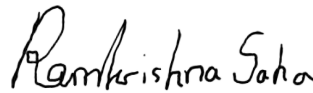
Supervisor:  
(Member)



---

Dewan Ziaul Karim  
Lecturer  
School of Data and Science  
Department of Computer Science and Engineering  
Brac University

Co-Supervisor:  
(Member)



---

Ramkrishna Saha  
Lecturer  
School of Data and Science  
Department of Computer Science and Engineering  
Brac University

Thesis Coordinator:  
(Member)

---

Md. Golam Rabiul Alam  
Associate Professor  
Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)

---

Sadia Hamid Kazi, PhD  
Chairperson and Associate Professor  
School of Data and Science  
Department of Computer Science and Engineering  
Brac University

# Abstract

Opinions from others can be important when a choice is necessary, mainly when valuable resources such as time and wealth are involved. People often depend on their peers' past experiences. The analysis of sentiments or opinions is a computation using text analysis of people's emotions, thoughts, and feelings. In recent years, it became the most significant natural language processing and sentiment analysis research approach. Because it is founded on people's views and as all people have different viewpoints to see things, it is becoming more popular every day. In this digital world, when a person decides to purchase a product or utilize a service, they get access to many customer evaluations. Still, it is a tiresome process to read and analyze them all. Moreover, when an organization wants to make a profit, find new possibilities, anticipate sales trends, and manage its reputation via public opinion or sell its product, it also needs to address many customer remarks accessible to its customers. Therefore, our goal is to show that it is feasible to do so with sentiment analysis approaches. With sentiment analysis, it is easy to analyze and extract a vast number of accessible data comments from both consumers that can aid in fulfilling the objectives of the organization. The dataset utilized here is obtained from the Ali-Express e-commerce website's online product reviews. We use various data processing techniques such as tokenization, removal of punctuation marks and stop words, stemming, TF-IDF, and parts of speech tagging. The results of our study comprise several methods of machine learning techniques. In this work, we examined eight distinct types of Machine Learning Algorithms, including Naive Bayes Classifier, SVM, Random Forest Classifier, Logistic Regression (L.R.), K- nearest neighbors, XGBoost Classifier, Decision Tree Classifier, and Gradient Boosting, and compared their precision and accuracy to find the most accurate one. According to our analysis, Logistic Regression is performing better among all the other seven classifiers having an accuracy of around 98%, and KNN is performing lowest, having an accuracy of 46%. Other classifiers such as Decision Tree is, giving 69%, Gradient Boosting 63%, Naive Bayes and SVM 82%, and Random Forest and XGBoost are giving 80% of accuracy.

**Keywords:** Machine Learning, Tokenization, Naive Bayes Classifier, SVM, Random Forest Classifier, Logistic Regression (L.R.), K-nearest neighbors, XGBoost Classifier, Decision Tree Classifier, Gradient Boosting, TF-IDF, etc.

## **Acknowledgement**

First and foremost, we thank the Almighty God for allowing us to complete our thesis without major interruption. Next, we would like to thank our supervisor, Dewan Ziaul Karim sir, and our co-supervisor, Ramkrishna Saha sir, for their unwavering support and assistance during the process of our project, which we greatly appreciate. Finally, we may not be able to fulfill our goals if we do not have our parents' unwavering support. We are on the approach of graduating as a result of their wonderful assistance and prayers.

# Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Acknowledgment	v
Table of Contents	vi
List of Figures	viii
List of Tables	ix
Nomenclature	x
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	3
1.2 Research Objectives . . . . .	4
<b>2 Literature Review</b>	<b>6</b>
2.1 Related Works . . . . .	6
<b>3 Methodology</b>	<b>9</b>
3.1 Proposed Model . . . . .	9
3.2 Work Plan . . . . .	10
3.2.1 Data Preprocessing . . . . .	10
3.2.2 Tokenization . . . . .	12
3.2.3 Changing Characters Into Lower Cases . . . . .	12
3.2.4 Removing Punctuation . . . . .	12
3.2.5 Remove Stop words . . . . .	12
3.2.6 Normalization . . . . .	12
3.2.7 Term Frequency (TF) Inverse Term Frequency (IDF) . . . . .	13
<b>4 Implementation</b>	<b>15</b>
4.1 Data Analysis . . . . .	15
4.1.1 Input Data . . . . .	15
4.1.2 Imbalance dataset . . . . .	17
4.1.3 Over-Sampling . . . . .	19
4.1.4 Under-sampling . . . . .	20

4.1.5	Combinations of Techniques . . . . .	21
4.2	Model Architecture Using Classifiers . . . . .	21
4.2.1	Naïve Bayes . . . . .	21
4.2.2	Support Vector Machine . . . . .	26
4.2.3	Random Forest . . . . .	28
4.2.4	Logistic Regression . . . . .	29
4.2.5	K-Nearest-Neighbors . . . . .	30
4.2.6	Decision Tree . . . . .	32
4.2.7	Gradient Boosting . . . . .	32
4.2.8	XGBoost . . . . .	34
4.3	Hyper-Parameters Tuning . . . . .	35
4.3.1	Technique01: Manual Search . . . . .	36
4.3.2	Technique02: Grid Search . . . . .	36
4.3.3	Technique03: Randomized Search . . . . .	37
4.3.4	Technique04: Halving Grid Search . . . . .	38
4.3.5	Technique05: Halving Randomized Search . . . . .	38
4.3.6	Technique06: Hyperopt-Sklearn . . . . .	38
4.3.7	Technique07: Bayes Grid Search . . . . .	38
<b>5</b>	<b>Analysis &amp; Results</b>	<b>40</b>
5.1	Analysis . . . . .	40
5.1.1	A. Confusion matrix . . . . .	40
5.1.2	B. Cross Validation: . . . . .	42
5.2	Result Analysis . . . . .	43
<b>6</b>	<b>Conclusion and Future Research</b>	<b>51</b>
6.1	Conclusion . . . . .	51
6.2	Future Work . . . . .	51



# List of Figures

3.1	Flow Chart of the proposed model of Sentiment Analysis on E-commerce Based Product Reviews Using Machine Learning Algorithm. . . . .	10
3.2	Steps of Data Preprocessing . . . . .	11
4.1	Screenshot of Dataset . . . . .	15
4.2	Pie Chart Of Different Sentiments Of Our Dataset . . . . .	16
4.3	Data Sampling . . . . .	18
4.4	Hyperplane of support vector machine . . . . .	27
4.5	Margin of support vector machine . . . . .	28
4.6	Working Model of Random Forest Classifier . . . . .	29
4.7	Working Model of Logistic Regression Classifier . . . . .	30
4.8	Steps for K-NN Analysis . . . . .	31
4.9	Working Model of Decision Tree Classifier . . . . .	33
4.10	Xgboost's principle features . . . . .	35
4.11	Working Model of Hyper-Parameters Tuning . . . . .	36
4.12	Working Model of Grid Search . . . . .	37
4.13	Working Model of Randomized Search . . . . .	37
5.1	Result Analysis . . . . .	44
5.2	Confusion Matrix of Logistic Regression . . . . .	46
5.3	Confusion Matrix of Desision Tree . . . . .	46
5.4	Confusion Matrix of Gradient Boosting . . . . .	47
5.5	Confusion Matrix of KNN . . . . .	47
5.6	Confusion Matrix of Naive Bayes . . . . .	48
5.7	Confusion Report of Random forest . . . . .	48
5.8	Confusion Matrix of SVM . . . . .	49
5.9	Confusion Matrix of XGBoost . . . . .	50

# List of Tables

4.1	Details Of Collected Products with Ratings. . . . .	16
4.2	Example of Naïve Bayes. . . . .	22
4.3	Calculation of Laplace Smoothing. . . . .	25
4.4	Result Analysis of K-Nearest-Neighbors Classifier. . . . .	32
4.5	Result of Hyper parameters tuning accuracy. . . . .	39
5.1	Percentage of Precision and Recall . . . . .	45

# Nomenclature

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

*FN* False Negative

*FP* False Positive

*KNN* K-Nearest Neighbor

*LR* Logistic Regression

*NB* Naive Bayes Classifier

*RFC* Random Forest Classifier

*SVM* Support Vector Machine

*TFIDF* Term frequency–Inverse Document Frequency

*TN* True Negative

*TP* True Positive

*XGB* Extreme Gradient Boosting Classifier

# Chapter 1

## Introduction

An emotion is a different feeling, opinion, thought, or judgment. It, sometimes referred to as opinion mining, examines the feelings of specific individuals. Mainly it is a technique in which the datasets consist of feelings, attitudes, or evaluations that take the way a person thinks. It is a callous effort to attempt and grasp the good and the bad in one statement. Sentiment analysis encourages consumers to determine if the product information is good or not before it is obtained. Marketers and companies use this analysis to understand their products and services to be given according to the consumer's demands. The Internet is a helpful location for knowledge about feelings. As the Internet user grows, every user is interested in presenting his view via different media. The data on the Internet has created an opinion-based opinion. Sentiment analysis helps examine these opinions and gain important information that will contribute to decision-making for other users.

The purpose of sentiment analysis is to discuss various items on the Internet. The challenge is to extract and classify the opinions based on numerous product attributes for a specific product or brand. The extraction and training of the classifier are based on the same. The characteristics should increase the exercise speed. Good precision must be attained on the test data. The characteristics of a product to be discussed must be established. It means that the user should be aware of the fundamental value of the product. The characteristics must be positively and negatively categorized.

The majority of research on sentiment analysis of product evaluations uses a binary classification of "positive" and "negative" reviews. Substant categorization work has been performed on reviews and comments from interactive websites using various machine learning methods. In this technique, the model evaluates their structure and the contextual dependence of each word inside the sentence, breaks them down into phrases, and examines each sentence to determine the reviewers' feelings.

Several sentiment analysis studies were conducted for different e-commerce platforms. For example-

- An e-commerce website analysis called Amazon, an examination of its beauty products by a KTH student, was conducted in 2018. He concentrated on summaries of the assessments, which were more valuable and accurate than the

entire assessment. He had accuracies with the SVM and Naïve Bayes classifier, which might reach more than 90% . For a great deal of data, he discovered that SVM performed better than Naïve Bayes.

- Ji Fang and Bi Chen are using two standard techniques for studying sentiments - lexical lookup and machine learning. Lexicon search begins with a positive and negative word lexicon. Lexicons of the present feeling do not capture the current sensitivity of feelings. The proposed method uses machine learning techniques such as SVM to incorporate sentiment lexicons into existing knowledge, which improves the accuracy of sentiment analysis.

In this paper, we will use various data processing systems such as tokenization, punctuation removal, stemming, Lemmatization, Normalization, and the TF-IDF vectorizer. Our study's findings include varying machine-learning approaches that demonstrate which will perform well by comparing their accuracy.

## 1.1 Problem Statement

Because many products are produced by different manufacturers, giving meaningful reviews is a must. The amount of studies related to a product or brand is scary, and dealing with big data is no exception. The sentiment approach of the evaluation is provided by classifying reviews based on customer sentiment into positive and negative opinions. As a result, it leads to enhanced judgment. Furthermore, according to an Amazon research, over 88% of online customers trust reviews as much as personal recommendations. Any online product with many good reviews makes a strong case for its validity. [1]

Books, or any other online goods, without feedback, make target users hesitant. Quite simply, a range of good reviews validates the authenticity of products to the customers. People appreciate others' consent and experience, and the only way to know others' impressions of a product is to read reviews. Opinions gathered from users about specific items or subjects directly impact future client buying decisions.

Many social media platforms make their application programming interfaces (APIs) available to researchers, allowing academics and technologists to collect and analyze data. Many famous application owners now offer three APIs: the REST API, the Search API, and the Streaming API so that the developers may use the REST API to get status data and user information and the Search API to find specific Twitter content. The Streaming API is used to capture Twitter content in real-time. Developers may also combine similar APIs to create their own apps. As a result, sentiment analysis looks to have a solid foundation thanks to massive amounts of online data. However, the inherent limitations of this type of Online Data may make sentiment analysis problematic. The first disadvantage is that people's opinions are unreliable since they may freely offer their ideas. For example, internet spammers publish spam on forums instead of discussing topical views. Some spams are inconsequential, while others contain irrelevant or fraudulent viewpoints. The second flaw is that such information is not always available. Precision is more like a label for an opinion, indicating whether it is good, harmful, or neutral. [29]

Our research aims to classify good, negative, or neutral customer comments on various items and create a supervised model to polarize a vast number of evaluations. If you are provided with a written text, you must extract expressions of opinion conveying a target attribute and classify them as one of the numerous feelings. The document level, sentence level, and phrase-level are the three levels of emotion polarity categorization depending on the breadth of the text. If a document sends a negative or positive message, it is checked by the document level. In contrast, the sentence level is concerned with the sentiment categorization of each sentence, and the phrase level is affected by what individuals like or dislike about their views. It is necessary to examine the contextual polarity of sentences. Many past studies have been conducted on this topic, including words and phrases categorized as having a previous positive or negative polarity. [44] Preliminary classification is helpful in many situations, but when contextual polarity enters the picture, the meaning obtained through positive or negative polarity may be vastly different. Understanding such feelings implies several activities.

The initial duty would be to acquire information from the e-commerce site. It contains user reviews of different items such as mobile phones, televisions, cameras, etc. Web scraping may be used to capture data on the fly. The polarity of the opinions must next be examined. For example, "good" and "bad" have negative and positive connotations. Following that, determining the strength of the viewpoint yields a more accurate result. Assume that the intensity of the words used to communicate the user's perspective is considered. "Good" and "great," for example, suggest varying levels of positive feeling. The strength may also be determined by adding "extremely" before any word. Finally, a review is categorized for sentiment classes. For example, the polarity of reviews determines which are excellent and negative reviews. [34] The opinion words are adjectives such as hatred and love, as well as other P.O.S. categories such as nouns, verbs, and adverbs. When a text has various features and emotions, a basic strategy such as extracting the opinion words closest to the target, then the attribute does not function effectively. It entails training the classifier on the same. The features should be designed so that the training pace is increased. The accuracy of test data must be high. The characteristics of a product under consideration must be determined. Users should know it as the foundation for the product's success.

Many research articles on product reviews, sentiment analysis, and opinion mining have been published. Researchers gathered sentiment from studies and evaluated the data to develop a commercial model for those works. They said that their displayed tools were trustworthy enough to supply them with great accuracy. Because of business analytics, their decision was made more appropriate. They also experimented with identifying false reviews, detecting emotions in studies, and guessing gender based on names. E-commerce websites may have a wide range of product reviews. These assessments can assist you in understanding consumer behavior and making sound decisions. It is feasible to conduct both structured and unstructured studies. Filtering away irrelevant data can provide valuable business insights. Big Data has helped businesses to thrive and grow by relying on facts rather than intuition. It helps with more targeted social influencer marketing, customer base segmentation, recognizing sales and marketing possibilities, detecting fraud, quantifying risks, better working on plans, customer segmentation analysis, etc.

To discover the best classifier for our study work, we want to train and categorize the review data by Naïve Bayes classifier, Random Forest, SVM, and Decision Tree which are some of the examples of machine learning categorization models as well as we applied cross-validation.

## 1.2 Research Objectives

Sentiment Analysis is a technique for identifying positive and negative sentiments in text. Organizations typically use it to discover reactions from neutral data, evaluate brand reputation, and for better sells. As it enables organizations to understand their customers' perceptions of their brand, this sentiment analysis is very important. Sentiment Analysis is becoming a virtual gadget to filter and grasp that opinion as

clients share their perspectives and thoughts more plainly. Automatically evaluating client input, such as reviews in overview reactions and web-based media debates, enables companies to understand what makes clients happy or unhappy and adjust products and administrations to answer their clients' needs. By automatically identifying the mood of social media chats, reviews, and other types of feedback, businesses can develop and grow faster and gain a lot of profits. Our objective is to create a better model for various evaluations that can deal with people's emotions. Our goals are as follows:

- To ensure growing one's business by handling all massive data efficiently and cost-effectively
- To develop a better model for sentiment extraction using machine learning.
- To discover a user's opinion about a specific item by evaluating a large quantity of information from several sources.
- To offer recommendations on improving the model.



# Chapter 2

## Literature Review

### 2.1 Related Works

Sentiment analysis refers to text mining that finds and extracts sentiment. It is fast-developing research and Natural Language Processing topic (N.L.P.). The significance of sentiment analysis is growing by the day. Product reviews by customers may be used to learn about people's attitudes regarding a given issue. This article examines the methods and strategies used to generate feature-by-feature product summaries and evaluates them to create a genuine review.

In [32], the author shows how subjective contents are retrieved and sentiment sentences with excellent or negative words. These statements are tokenized into English words that are separated. They used a feature vector for the dataset. The primary purpose of sentiment analysis on product reviews is to check a unique set of criteria and tactics for extracting product characteristics and analyzing them to form a genuine review. They employed P.O.S. tagging to get a better outcome. [29], [32]

In [29], the objective is to determine the intensity of opinions by paying attention to modifying the words that express the opinions and extracting the characteristics of the opinions generated by the users.

In [38], the authors tried implementing some systems for analysis. They used Sentence Level, Aspect Level, and Document Level to detect and extract opinions from the text. The pre-processing part does tokenization, punctuation removal, and stemming. For sentiment scores, they use Score Generation. In [38] for accuracy, the author used Naive Bayes and SVM.

In [37], the authors remove the HTML tags in data pre-processing. The authors used Multinomial naive Bayesian (M.N.B.), linear support vector machine L.S.V.M., and Long short-term Memory (L.S.T.M.). Here they used the Keras library, which consists of 4 layers. (Embedding, Spatial dropout, L.S.T.M., Dense)

We shall discover something new in the paper [31] the author's intuitionistic fuzzy set theory for rating all of the goods. They also recognized positive and negative attitudes using the How Net sentiment lexicon. PROMETHEE II, which was created by Brans in 1982 and has been frequently used to handle practical decision-making

issues, has also been used.

In this research article [33], The author analyzes consumer product reviews for sentiment using different ML techniques and unstructured data from Amazon Mobile Phone Reviews. It was pre-processed to measure the view of the reviews using supervised learning after being filtered to remove errors that occurred in the data. They found datasets for this review by applying different ML models. They went through a process to find the best classifier for this task. Their ultimate objective was to collect many online reviews from Amazon.com. The data collection includes over four million reviews for around four thousand five hundred mobile phones. In the second phase, they pre-processed the data from a CSV file, such as stopping removing words, emitting punctuation marks, numerals, etc. They then became involved in feature extraction. After collecting positive and negative attitudes, they employed different ML classifiers. They divided the positive and negative thoughts using "the polarity approach." Because good evaluations outnumbered negative ones, they utilized "Under Sampling." The dataset is trained and tested using a random sample of three thousand data. They have now employed the K-10 model for cross-validation to examine the outcome section. In the future, the work may be expanded to conduct multiclass categorization of reviews, which will provide the customer with a displayed type of review and, therefore, a better assessment of the item. It may also predict an item's ranking from a survey. It will provide clients with an actual rating since sometimes the rating received by the thing and the opinion of the reviews do not correspond.

In this paper, [43] concentrates on Bengali sentiment analysis using K.N.N. Their study works with the user feedback of Bengali text. They have collaborated with a Bangladeshi e-commerce site known as 'Daraz' with their vast datasets. For this task, they have chosen five different machine learning algorithms: K.N.N., XGI-Boost, Support Vector Machine, Logistic Regression, and Random Forest. The initial hypothesis of their program was to collect data and label the data. Feature extraction and sampling were done after processing the data, and after that, they moved on to sentiment detecting. They have collected around 7095 feedbacks about their products from a different range of customers. They have mainly focused on eight kinds of information: product type, category, reviewer, product name, comment, comment data, and ratings. They have also scaled the data set from 1 to 5. For doing feature extraction, they have preferred to choose TF-TDF Vectorizer. It is witnessed that a Random Forest Classifier provides a precision of approximately 90.84% in comparison with other algorithms. However, K.N.N. gives an exactness of 96.25%. They have only worked with fewer datasets so far, so this model can be developed for a much bigger dataset.

The fundamental goal of this paper, [41], was to utilize the SVM classifier, which analyzes multiple datasets to characterize sentiments and language in product evaluations. Furthermore, multiple data sets were utilized for training, testing, and implementing Support Vector Machine learning algorithms. They carried out text preprocessing, data clustering, and SVM classification. Several datasets were used in this work for training, testing, and simulating the support vector machine learning method to calculate the polarity of ambiguous emotions or reviews. During the

Transformation process, each word is computed using "TF-IDF." They utilized an SVM classifier for positive and negative evaluations after eliminating outliers using data clustering, following the application of the evaluation technique, accuracy, and recall, the Support Vector. Using the evaluation technique, precision, and recall, the Support Vector Machine (SVM) classification algorithm outperforms the others with 89.98% accuracy. Further phrase constructions would improve this accuracy even more. Finally, it is presumed that the SVM worked.

This paper, [1] illustrates the sentiment analysis of commodity feedback while implementing text categorization and pool-based active learning in the machine learning approach. The author of this paper has profoundly used the supervised learning technique on an extensive scale of Amazon set to separate it and get the expected outcome. They have associated with a large data set and performed data pre-processing and feature extraction. They have chosen to use Naive Bayes, SVM, S.G.D., Linear Regression, Random Forest, and Decision Tree for this program. SVM is the most accurate amongst them. In practice, the follow-up functions may be done by fully automating the data labeling process with relatively little assistance from Oracle. The model is compatible with applications that can communicate with consumers.

The writer of [28] focuses on the analysis of sentiments on reviews of movies by applying a variety of ML algorithms. They have divided the phases into three portions. This three-part collection comprises the steps of pre-processing, feature extraction, and classification. For more sophisticated classification algorithms, they used the Naive Bayes classifier. They then used the K.N.N. classifier for a variety of datasets. Finally, for this dataset, they employed Random Forest Classifier. They have gathered data from over 2000 user-generated movie reviews on IMDb. They used the 'Weka tool' to perform text preparation. The "Weka tool" which are using data mining activities such as clustering algorithms, prediction, and character extraction are supported through "Weka tool." Even though several algorithms were utilized, Naive Bayes had the highest accuracy of 81.4 percent. Because the author worked with a smaller dataset and a smaller number of methods, other algorithms may provide better accuracy for this dataset.

# Chapter 3

## Methodology

### 3.1 Proposed Model

Our proposed approach investigates the relationships between online product reviews and revenue performance using sentiment analysis and machine learning technologies. The algorithms are performed on product reviews to anticipate product collection based on the reviews and analyze how much influence the reviews have on the collection. Based on today's user reviews, the product collection for the following day is determined. It is also estimated whether the collection would be high or low.

The very first module involves data collecting and pre-processing. The experimental data is initially obtained from the e-commerce website Aliexpress.com. Our research proposes a supervised learning approach for polarizing massive unlabeled review data. We must classify unlabeled data before using it in a supervised learning model. The data is then pre-processed in the second stage. Text pre-processing techniques are classified into P.O.S. tagging and stop word removal. Textual data in P.O.S. comprises tokens, which are blocks of characters. The input reviews are tokenized, and pre-processing begins.

The following phase is feature selection, which involves extracting valuable characteristics from the acquired data. Only three of the five features have been considered in the dataset, namely the product name, rating, and reviews. The review's sentiment orientation is determined. That conduct supervised learning, and the dataset is provided sentiment tags to match each review. The classified data is trained and evaluated using eight different models in the following stage. Our model employed many classifiers to get a decent degree of accuracy. After obtaining a good result, we processed the tagged datasets. In Figure 3.1, we tried to show a flowchart of our proposed model. At the 'Initialization step,' we collected data, did the necessary processing, and selected the attributes. After that comes the 'Learning step,' we trained 80% of our data according to the algorithm of our classifiers. Lastly' we applied classifiers on 20% of our Test Data Set data at the' Evaluation step.'

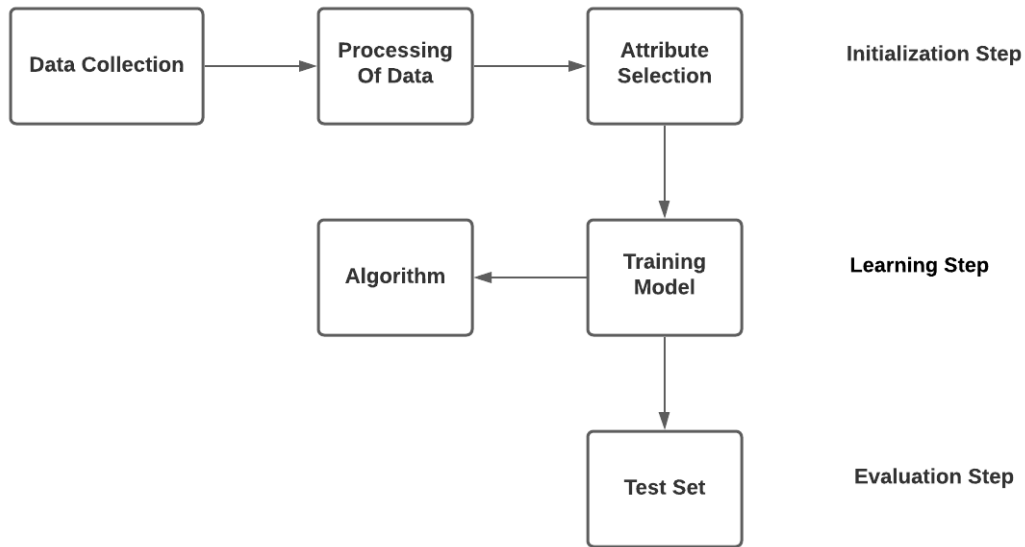


Figure 3.1: Flow Chart of the proposed model of Sentiment Analysis on E-commerce Based Product Reviews Using Machine Learning Algorithm.

## 3.2 Work Plan

### 3.2.1 Data Preprocessing

Sentiment analysis on textual data is commonly used to help firms analyze brand and product sentiment in consumer feedback and better understand client wants. So to find the sentiment polarity precision (Positive Negative) of our novel dataset, we have to preprocess the dataset. The data preprocessing steps with description are given below:

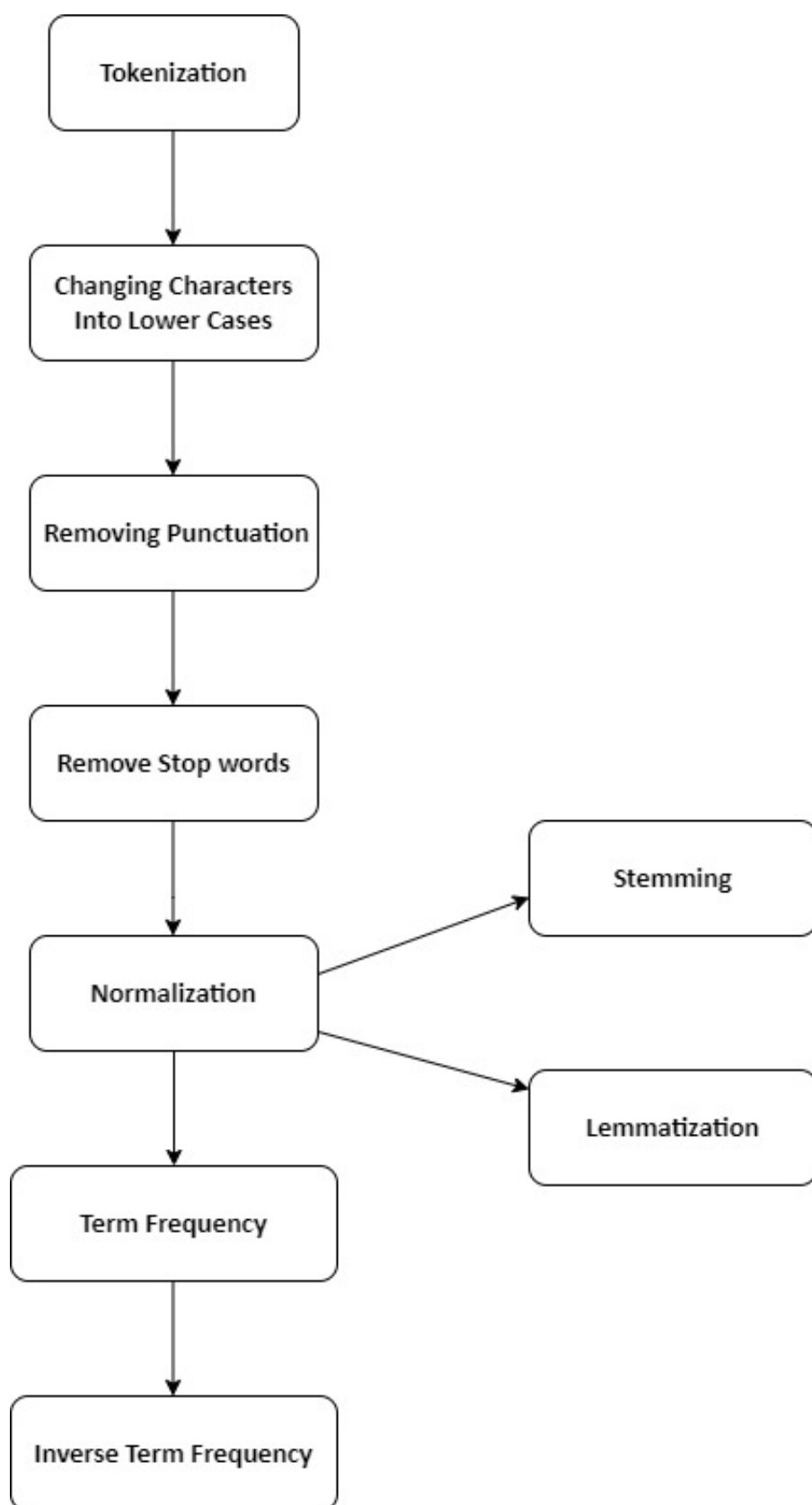


Figure 3.2: Steps of Data Preprocessing

### **3.2.2 Tokenization**

In NLP, the first step is to find tokens or fundamental units that do not need to be fragmented in further processing. The entity word is the most fundamental type of token in NLP. However, we are concerned about employing a computer to detect tokens that do not have separate delimiters, such as English idioms and fixed phrases. [2] Tokenization is breaking down a statement, comment, essay, or even a textual report into smaller pieces, such as specific expressions. Tokens are the names given to these smaller units. Words, numerals, or punctuation marks might be used as tokens.

The nltk library was used to fix the majority of these issues. The module converts words into tokens that are then utilized as input for the normalization and sterilizing procedures. It can also turn a text string into numerical data that machine learning algorithms can use.

### **3.2.3 Changing Characters Into Lower Cases**

We changed all of the characters in the review text to lowercase. It helped us in avoiding repetitive content. "GOOD," "good," and "Good" all of these terms signify the same thing. Still, differing character cases appear as distinct entities, causing unnecessary calculations.

### **3.2.4 Removing Punctuation**

Following that, all punctuation (e.g.,?! ) is removed. During frequency generation in this method, punctuations were avoided.

### **3.2.5 Remove Stop words**

Prepositions (e.g., at,of,for,to,after), conjunctions (e.g., and,or,but), and articles (a,an,the) are all considered stop - word in this paper. These stop words may cause an unexpected rise in frequency, resulting in computational complexity and, as a result, reduced performance.

### **3.2.6 Normalization**

Normalization is a process with a more advanced cleaning procedure that maintains smoothness. During this process, The inflectional form is eliminated to acquire the primary form of a word. Normalization also reduces the number of identical characters in a word, eliminates variances, and removes any extraneous information. [49] Two popular main methods of Normalization are stemming and lemmatization which combine all of the words under one cover. Though they are both normalization methods, they are very different. We have used Normalization in our dataset.

## i. Stemming

Several word variations provide no new information and instead produce redundancy, resulting in ambiguity while training machine learning models for predictions. Let us take the statements "He likes to walk" and "He likes to walk." Both represent the same thing. In our case, the stemming function will remove the suffix from "running" and change it to "run." The "Porter Stemmer" module is being used in the case to run the project. [46]

## ii. Lemmatization

Unlike stemming, lemmatization uses vocabulary and morphological study of words to produce Normalization. Lemmatization aims to delete only inflectional ends and return the base form of the lemma word. Though it is slower than stemming, it is a much more effective and intelligent text analysis technique. Its goal is to keep the words' structural-relationship-maintained. The WordNetLemmatizer() function was the first and most commonly used, so we also used it. [46]

### 3.2.7 Term Frequency (TF) Inverse Term Frequency (IDF)

For our instance, TF-IDF is a measure of how important a word is in a comment relative to all the words in our training data. For example, if a word such as "the" appeared in most of the comments, the TF-IDF would be small as this word does not help us differentiate across comments. Note that "**TF**" means "**Term Frequency**" whereas "**IDF**" means "**Inverse Document Frequency**". Specifically, "**TF**" denoted by  $tf(w, c)$  is the number of times the term  $w$  appears in the given comment  $c$ . Whereas "**IDF**" is a measure of how much information a given word provides in differentiating comments. Specially, "**IDF**" is formulated as -

$$idf(w, D) = \frac{\log(\text{Number of comments in train data } D)}{(\text{Number of comments containing the word } w)} \quad (3.1)$$

To combine "**TF**" and "**IDF**" together, we simply take the product, hence -

$$TFIDF = tf(w, c) \times idf(w, c) = \frac{(\text{Number of times } w \text{ appears in comment } c \times \log(\text{Number of comments in train data } D))}{(\text{Number of comments containing the word } w)} \quad (3.2)$$

Now the **TF-IDF** can be used to weight the vectors that result from the "Bag of Words" approach. For Example, suppose a comment contains "this" 2 times, hence  $tf = 2$ . If we then had 1000 comments in our training data, and the word "this" appears in 100 comments,  $idf = \log(1000/100) = 2$ . Therefore, in this example, the **TF-IDF** weight would be  $2*2 = 4$  for the word "this" appear twice in a particular comment. To incorporate **TF-IDF** into the naive-bayes setting, we can compute -



$$Pr(word|spam) = \frac{(\sum_{c \text{ is spam}} TFIDF(word, c, D))}{(\sum_{word \text{ in spam } c} \times (\sum_{c \text{ is spam}} TFIDF(word, c, D)) + \text{Number Of Unique Words In Data})}$$

(3.3)

# Chapter 4

## Implementation

### 4.1 Data Analysis

Our dataset is a self-created novel dataset in which we gathered product data from Ali-Express.com and converted it to CSV format. How we gathered our input data and how we pre-processed our collected data are briefly explained below —

#### 4.1.1 Input Data

Ali-Express is a reputable website that has earned a reputation among customers for high-quality products, fair prices, and timely delivery. Over 34,892 Ali Express product reviews and product names, product ids, and reviewer names have been compiled into a dataset. We had to manually collect the data because we tried multiple web scraping tools but could not find any that scraped the Ali-Express products the way we wanted. We used Chromes's built-in web scraping tool, Webautomation.io, Parsehub, ScrapeHero, Octoparse, and Webharvy as web scraping tools. However, we tried several coding techniques such as python, BeautifulSoup, and others to share Ali Express items but could not find a suitable solution. We are highly likely to receive different languages of reviews of different products because we are personally collecting 34,892 data.

Product Name	Product ID	Reviewer	Rating	Comment
Multifunctional Waterproof Intelligent Smart	1.01E+15	S***r	5	Everything is cool
Multifunctional Waterproof Intelligent Smart	1.01E+15	G***v	5	All norms
Multifunctional Waterproof Intelligent Smart	1.01E+15	Z***r	5	Thank you seller! I received the parcel, two products together, as in the spring. I'm very glad, I recommend!
Multifunctional Waterproof Intelligent Smart	1.01E+15	F***e	5	Arrived before time
Multifunctional Waterproof Intelligent Smart	1.01E+15	S***a	5	Very Excellent.
Multifunctional Waterproof Intelligent Smart	1.01E+15	Y***i	5	Pretty ring overall. Come home quickly enough. I do not know if it's the black color I took but the degrees are almc
Multifunctional Waterproof Intelligent Smart	1.01E+15	4***r	5	The order is satisfied, though long went, then somewhere lost since 24.11 took Toko 20.12 in the mail, although he
Multifunctional Waterproof Intelligent Smart	1.01E+15	P***o	5	Excellent meets the expectations
Multifunctional Waterproof Intelligent Smart	1.01E+15	K***i	5	It works like a thermometer sticking on the glass of the aquarium. Cool thing.
Multifunctional Waterproof Intelligent Smart	1.01E+15	M***a	5	Contains the property of feeling the body heat as shown in the picture I recommend dealing with this seller
Multifunctional Waterproof Intelligent Smart	1.01E+15	S***y	5	Oki looks cool. But no what count to properly pokazywa?o temp.
Multifunctional Waterproof Intelligent Smart	1.01E+15	S***v	5	It's been over a month. I ordered it-it's not clear! Ring, blue, glitters. Temperature, like, seems! Accuracy-error in 1

Figure 4.1: Screenshot of Dataset

Figure 4.1 demonstrates the compilation of our data set. We gathered information based on the Product Name, Product ID, Reviewer, Rating, and Comment. We have shown the number of the products and the number of all-star (from 1-to 5)rating reviews.

Number of products	5 Star	4 Star	3 Star	2 Star	1 Star
99	6476	5177	5236	5667	8468

Table 4.1: Details Of Collected Products with Ratings.

In Table 4.1, we presented the details of a total review of 99 products, and among those, 5218 have 5 Star ratings, 3801 have 4 Star ratings, 3666 have 3 Star ratings, 2546 have 2 Star ratings, and 4869 have 1 Star ratings. As there are five individual ratings, we will consider ratings 5 and 4 as positive, 3 as positive, and 2 and 1 as Negative.

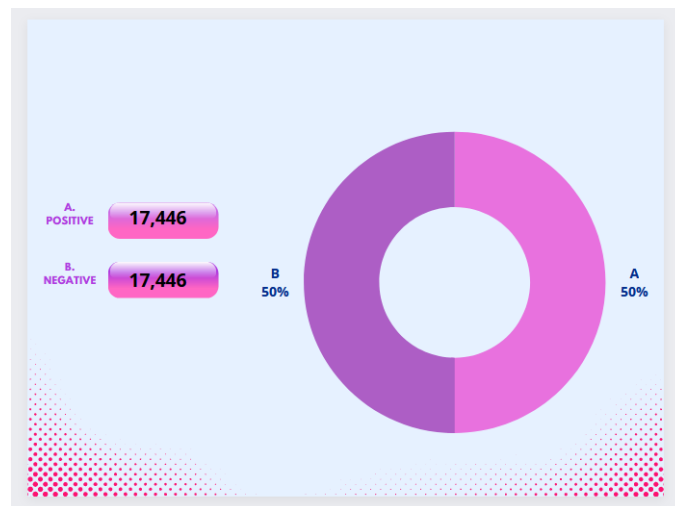


Figure 4.2: Pie Chart Of Different Sentiments Of Our Dataset

Here, Figure 4.2 indicates the percentage of three different sentiments. We have chosen ratings 1 and 2 as Negative, having a total of 38% , rating 3 as Neutral having a total of 19%, and rating 4 and 5 as Positive, consisting of 43%.

### 4.1.2 Imbalance dataset

An imbalanced set of data is one with unequal categorization percentages. The majority of classes make up a high share of the data set. Minority classes make up a lesser share of the population. The class distribution is biased or skewed instead of being equal or near to equivalent. For example, we collected review sentiments and have 80 positive reviews and 20 negative reviews, with only these samples representing our dataset which we used for training data. It's an instance of an unbalanced categorization problem. A variety of reasons might contribute to the class distribution in an imbalanced categorization predictive analysis task.

The two fundamental causes for this are : data sampling and domain properties. One such foe is observational, or measurement error, frequently cited as the most prevalent error in field-based data collecting. When there is a flaw in your measuring procedure, you have measurement mistakes. .

#### A. Biased Sampling

#### B. Measurement Errors

For instance, examples may have been drawn from a certain geographical region or historical time passes, and the distribution of classes may shift dramatically or even reverse.

Errors may have occurred during the collection of the observations. One form of mistake can be applied by incorrect multiple samples. Alternatively, the procedures or mechanisms that gathered instances might have been broken or degraded, resulting in the imbalance. When an imbalance is caused by a sample bias or a measurement mistake, it is usually simple to repair the imbalance by optimizing sampling techniques and correcting the measurement error. This is due to the training dataset not accurately representing the issue. The imbalance might be a feature of the issue domain.

For example, the natural occurrence or existence of one course may outweigh the presence or occurrence of other classes. This might be because of producing observations in one type takes more time, money, calculations, or other resources. As a result, collecting more domain samples to enhance class distribution is usually impractical or impossible. To understand how to differentiate between classes, a model is necessary. There could be several strategies for creating a balanced dataset from an imbalanced dataset. However, the most popular and beneficial are Data Sampling, Clustering the abundant class, Class weights in the models, Change Evaluation Metric, and Miscellaneous. We employed a mixture of Data Sampling strategies called a combination of techniques that was more effective than any other data sampling method.

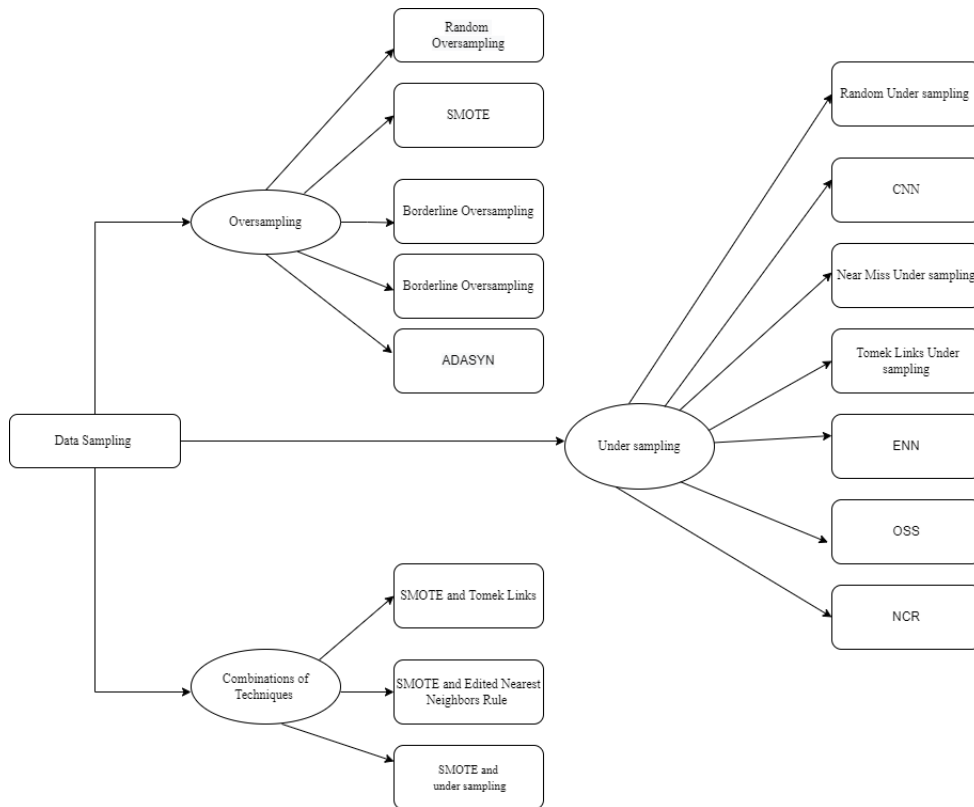


Figure 4.3: Data Sampling

### 4.1.3 Over-Sampling

When one type of data represents an underrepresented minority group in the data sample, over-sampling tactics may be used to repeat these outcomes for an equal percentage of positive findings in training. Oversampling should be used when data scientists do not have enough information. One type is plentiful, or the majority, while the other is rare, or the minority. Scientist raises the number of rare events by oversampling. The scientist uses some kind of technology to generate fictional incidents. [27]

#### Random Oversampling

Random Oversampling entails including extra copies of select minority classes in the training data. In minority class, instead of reproducing each copy, we can choose some random and then can replace. Oversampling can be performed several times. It is known as the oldest method proposed, which is trustworthy as well. [22].

#### SMOTE

In numerous ways, a dataset used in a standard classification issue may be over-sampled (given a labeled training set of photos, using a classification algorithm to categorize a series of images ). The most common technique is SMOTE (Synthetic Minority Over-sampling Technique). To demonstrate how this strategy works, consider several training instances with  $s$  sample and  $f$  features in the data's feature space. It's important to note that these features are constant when it comes to simplicity. Consider a dataset of birds that needs to be classified. Beak length, wingspan, and weight might be used to represent the feature space for the minority class that we want to oversample (all continuous). Choose a random sample from the collection and consider its  $k$  closest neighbors to oversample (in feature space). Combine the vector between one of those  $k$  neighbors with the most recent data point to create a synthetic data point. Multiply this vector by a random variable  $x$ , with a value ranging from 0 to 1. To make the new, fictional data point, combine this with the current data point. [3]

#### ADASYN

The ADASYN algorithm, or adaptive synthetic sampling, expands the concept of SMOTE by emphasizing the importance of the categorization border in problematic minority classes. [9] For specific minority class instances, ADASYN uses a weighted distribution based on their level of complexity in training, with more artificial data generated for minority class examples that are more difficult to learn.

#### Borderline-SMOTE

Borderline-SMOTE entails identifying incorrectly classified examples of the minority class, such as via a  $k$ -nearest neighbor classification model, and only creating

synthetic data that are "difficult" to classify. [7]

### **Borderline Oversampling**

Borderline Oversampling is a SMOTE addition that adapts an SVM to the dataset and uses the binary classifier specified by the support vectors as the foundation for constructing synthetic samples, assuming that the decision border is the area where more minority cases are needed. [15]

#### **4.1.4 Under-sampling**

Under-sampling is a strategy for balancing uneven datasets in which all data in the minority class is retained but the size of the majority class is reduced. It's one of a number of methods data scientists can use to extract more accurate data from a set of disparate sources. If there is enough information for a detailed analysis, under-sampling is feasible. The data scientist uses all of the rare events while reducing the amount of abundant cases to create two classes of similar size. To achieve the same diversity of events as the minority class, scientists generally erase occurrences at random from the majority class. [12]

### **CNN**

CNN was created to minimize the amount of data used by the k-nearest-neighbors approach. It works by identifying the instances in the dataset and adding them to the store only if its current contents cannot correctly classify them. After all instances in the minority class have been determined to add to the store, it might decrease numerous cases in the majority class. [42]

### **Near Miss**

"Near Miss" methods use KNN to choose examples from the majority class. NearMiss-1 locates instances from the majority class that is the smallest distance from the other nearest minority group examples. NearMiss-2 chooses largest group instances with the shortest average duration from among the three important selected features. For each nearest minority class example, NearMiss-3 selects numerous majority class cases. [48]

### **Tomek Links**

Tomek Links, the most well-known removal under-sampling method, was first devised as part of an update to the Condensed Nearest Neighbors algorithm. A Tomek [30] Link is a pair of samples in the training dataset that are both nearest neighbors (have the shortest route in feature space) but belong to discrete categories. Tomek Links are frequently misclassified examples located along the class border, and the

instances in the majority class are discarded.

## ENN

Another approach for picking examples for deletion is the Edited Nearest Neighbors rule, or ENN [14] for short. This rule entails finding and eliminating misclassified cases in a dataset using  $k=3$  nearest neighbors. The ENN technique can be run on the same dataset numerous times to improve the selection of instances in the majority class. This addition was first referred to as "unlimited editing," but it is now more widely known as Repeatedly Edited Nearest Neighbors. In addition to the "choose to keep" vs. "select to remove" families of under-sampling strategies, additional ways combine both approaches.

## OSS

The Condensed Nearest Neighbor (CNN) Rule and Tomek Links are combined in the One-Sided Selection (OSS) approach. [8] The Tomek Links approach is used to remove noisy examples from the class border, whereas CNN is being used to eliminate redundant instances from the majority class's inner density.

## NCR

The Neighborhood Cleansing Rule, or NCR, is another combination under-sampling method that utilizes the Condensed Nearest Neighbor (CNN) Rule to eliminate redundant instances and the Edited Nearest Neighbors (ENN) Rule to minimize noisy or confusing examples.

### 4.1.5 Combinations of Techniques

SMOTE is a very well and commonly utilized oversampling technique. Consequently, it is typically combined with one of several distinct under-sampling approaches. In the study that presented the approach, the most straightforward pairing was combining SMOTE [40] with random under-sampling, which was indicated to perform better than using SMOTE alone. After SMOTE, the technique is applied to the dataset, allowing the editing phase for both the minority and majority classes. The goal is to remove noisy points along the class border from both classes, which appears to improve the performance of classifiers fitted to the altered dataset.

## 4.2 Model Architecture Using Classifiers

### 4.2.1 Naïve Bayes

For sentiment analysis, a Naïve Bayes classifier is a set of deterministic techniques that assess how well a sentence will be right or wrong. The possibility of A if B is



accurate, equivalent to the likelihood of B if A is valid and times by the likelihood of A becoming true partitioned by the possibility of B being true: This is how this theorem works.

$$P(A/B) = P(B/A) * (P(A)P(B)) \quad (4.1)$$

The Naive Bayes algorithm compares words. Using word polarity-trained machine learning models, we can assess whether a word, phrase, or paragraph is positive or negative. Naive Bayes gets more predictive when techniques like lemmatization, stop-word elimination, and TF-IDF are utilized. Here's an instance of how it was performed:

Number	Text	Sentiments
01	Great. I liked it. Fast delivery	Positive
02	Very nice. Looks very good.	Positive
03	great product	Positive
04	Looks like a good quality charger. The seller was nice and shipped quickly	Positive
05	I am satisfied with my order Thank you I recommend	Positive
06	Great company.	Positive
07	Good product and fast charger	Positive
08	Bad product	Negative
09	not like it	Negative
10	I didn't received the product, refund me!	Negative

Table 4.2: Example of Naïve Bayes.

Table 2 represents some examples of Naïve Bayes classifiers with positive and negative sentiments.

### Testing Example

#### Review:

Great. I liked the charger.

It is solved by using two steps-

A. Feature Engineering

B. Calculating Probabilities

Total words in positive class = 42

Total words in negative class = 13

Total unique words = 12 words

Text = “**Great. I liked the charger.**”

#### Feature Engineering :

$$\begin{aligned} \text{Probability}(\text{Text}/\text{Positive}) &= P(\text{Great}/\text{Positive}) \times P(\text{I}/\text{Positive}) \times \\ &P(\text{Like}/\text{Positive}) \times P(\text{The}/\text{Positive}) \times P(\text{Charger}/\text{Positive}). \end{aligned} \quad (4.2)$$

$$\begin{aligned} \text{Probability}(\text{Text}/\text{Negative}) &= P(\text{Great}/\text{Negative}) \times P(\text{I}/\text{Negative}) \times \\ &P(\text{Like}/\text{Negative}) \times P(\text{The}/\text{Negative}) \times P(\text{Charger}/\text{Negative}) \end{aligned} \quad (4.3)$$

$$P(\text{Great}/\text{Positive}) = \frac{3}{42}$$

$$P(\text{Great}/\text{Negative}) = \frac{0}{13}$$

$$P(\text{I}/\text{Positive}) = \frac{3}{42}$$

$$P(\text{I}/\text{Negative}) = \frac{1}{11}$$

$$P(\text{Liked}/\text{Positive}) = \frac{1}{42}$$

$$P(Liked/Negative) = \frac{0}{11}$$

$$P(The/Positive) = \frac{1}{42}$$

$$P(The/Negative) = \frac{1}{11}$$

$$P(Charger/Positive) = \frac{2}{42}$$

$$P(Charger/Negative) = \frac{0}{11}$$

### Laplace Smoothing :

We add 1 to the numerator in Laplace Smoothing and k to the denominator to ensure that our posterior probabilities are never zero. If our training set misses a specific component, the posterior probability is  $1 / N + k$  rather than zero. Putting this number into the product does not affect our ability to anticipate like putting a zero would.

$$\frac{(\text{Number of reviews with } x' \text{ and target } out_{come})}{N + \alpha \times k} = \frac{(\text{positive} + \alpha)}{N + \alpha \times k} \quad (4.4)$$

In a simple language, we can say, actually it works like –

$$P(\text{word}) = \frac{\text{Word count} + 1}{\text{Total number of words} + \text{Unique words}} \quad (4.5)$$

Word	P(Word/Positive)	P(Word/Negative)
Great	$\frac{3 + 1}{42 + 12} = 0.0740$	$\frac{0 + 1}{13 + 12} = 0.04$
I	$\frac{3 + 1}{42 + 12} = 0.0740$	$\frac{1 + 1}{13 + 12} = 0.08$
Liked	$\frac{1 + 1}{42 + 12} = 0.0370$	$\frac{0 + 1}{13 + 12} = 0.04$
The	$\frac{1 + 1}{42 + 12} = 0.0370$	$\frac{1 + 1}{13 + 12} = 0.08$
Charger	$\frac{2 + 1}{42 + 12} = 0.0555$	$\frac{0 + 1}{13 + 12} = 0.04$

Table 4.3: Calculation of Laplace Smoothing.

### Calculating Probabilities :

$$\begin{aligned}
 & \text{Probability}(\text{Text}/\text{Positive}) = \\
 & P(\text{Great}/\text{Positive}) * P(\text{I}/\text{Positive}) * P(\text{Like}/\text{Positive}) * P(\text{The}/\text{Positive}) * \\
 & P(\text{Charger}/\text{Positive}) = 0.0740 * 0.0740 * 0.0370 * 0.0370 * 0.0555 = \\
 & 0.000000416063742 \quad (4.6)
 \end{aligned}$$

$$\begin{aligned}
 & \text{Probability}(\text{Text}/\text{Negative}) = \\
 & P(\text{Great}/\text{Negative}) * P(\text{I}/\text{Negative}) * P(\text{Like}/\text{Negative}) * P(\text{The}/\text{Negative}) * \\
 & P(\text{Charger}/\text{Negative}) = 0.04 * 0.08 * 0.04 * 0.08 * 0.04 \\
 & = 0.0000004096 \quad (4.7)
 \end{aligned}$$

$$\text{Probability}(\text{Text}/\text{Positive}) > \text{Probability}(\text{Text}/\text{Negative}) \quad (4.8)$$

”Great. I liked the charger.” This review belongs to the positive class.

## 4.2.2 Support Vector Machine

The support vector machine method is a statistical classification strategy that optimizes the hyperplane separation margin between cases. It's a non-probabilistic binary linear classifier that can effectively differentiate between classes by a large margin. It's a powerful classifier that can handle any size feature vector. A support vector machine (SVM) is a regression and classification algorithm that uses supervised machine learning. It is, however, most commonly utilized when there are problems with classification. [16]

In our study, we employed a support vector machine to classify a set of opinions as positive or negative. SVM is a complexity theory application. The distinctive quality of SVM is that it can learn even when given a large amount of data. Since it can handle large data, SVM is useful for text classification. Another feature of SVM is that it is robust when a small number of examples are split over a high-dimensional space. It is also linearly separable in the majority of data sets. In opinion mining, SVM has given reliable outcomes, and it has outperformed other machine learning algorithms. [16]

Support vector machines can classify by creating hyperplanes that divide data into multiple categories. So that the computer may learn how to categorize successfully, the SVM classifier needs first to be trained on a set of user evaluations. Following the training process, the system learns a classifier. The classifier produces a set of vectors containing the aspect and its sentiment classification. [21]

Given a set of data with labels,

$$D = (x_1, y_1), (x_2, y_2), \dots, (x_m, y_m), \text{ where } x_i \in X \text{ and } y_i \in \{-1, +1\}, \quad (4.9)$$

by using a hyperplane representation of SVM:

$$f(x) = \left( \sum_{i=1}^m \alpha_i K(x_i, x) \right) + b = 0 \quad (4.10)$$

$K(u, v)$  is a kernel function that meets Mercer's condition. Because the previously indicated hyperplane may be considered as a decision boundary, the sign of  $f(x)$  yields the expected label of input  $x$ . For a linear SVM,  $K(u, v)$  is the inner product of  $u$  and  $v$ . The equation  $f(x)$  may be rewritten as:

$$f(x) = wx + b, \text{ where, } w = \sum_{i=1}^m \alpha_i x_i \quad (4.11)$$

where the Lagrange multipliers are determined in such a way that  $f(x) = 0$  denotes the best possible hyperplane with the highest margin.

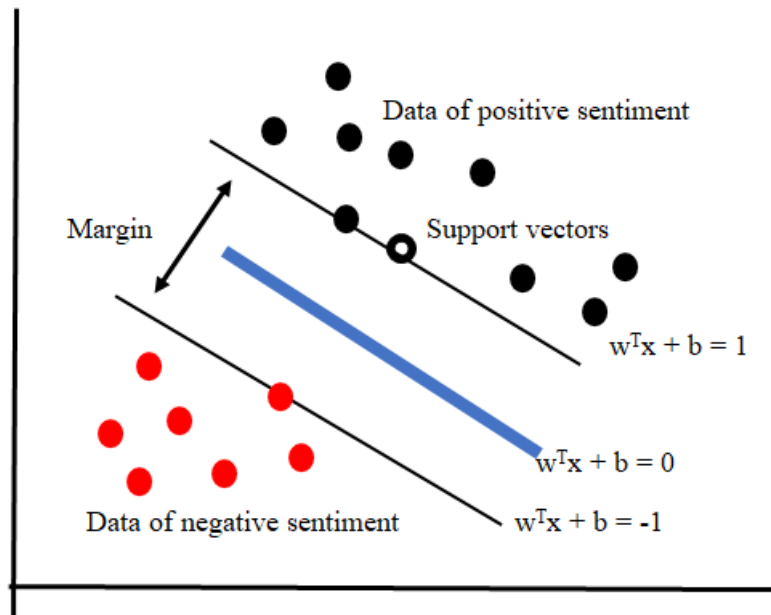


Figure 4.4: Hyperplane of support vector machine

Then the margin is given by:

$$\begin{aligned} \frac{w}{\|w\|} \cdot (x_+ - x_-) &= \frac{w^T(x_+ - x_-)}{\|w\|} \\ &= \frac{w^T \left( \left( \frac{+1-b}{w^T} \right) - \left( \frac{-1-b}{w^T} \right) \right)}{\|w\|} = \frac{2}{\|w\|} \end{aligned}$$

Figure 4.5: Margin of support vector machine

There are so many such hyperplanes that can divide the dataset into two regions. However, SVM makes sure that it chooses the hyperplane that is the furthest away from the nearest data points in the two regions. There are only a few hyperplanes that meet this requirement. SVM provides accurate classification results by ensuring this condition.

### 4.2.3 Random Forest

It is known as a learning approach which is supervised. This classifier is applied to classification and regression both. This may be the most adaptable and consumer-friendly approach. A rainforest is made up of trees. The plants in a rainforest are meant to strengthen it. The random forest algorithm constructs decision trees from the randomized dataset, extracts estimates of each structure, and then decides on the best choice. It also serves as a reliable gauge of the efficiency of the feature. It's a decision tree ensemble method based on randomized segmented data (the divide-and-conquer methodology). The forest is a collection of decision tree classifiers. Using an attribute selection indicator such as information gain, gain ratio, or Gini index, individual decision trees are constructed for each feature. The outcome of regression is the average of all the tree outputs. It is simpler and more resilient than previous non-linear classification techniques. In Figure 4, we tried to visualize the working model of Random Forest by a Flow Chart for better understanding. [19]

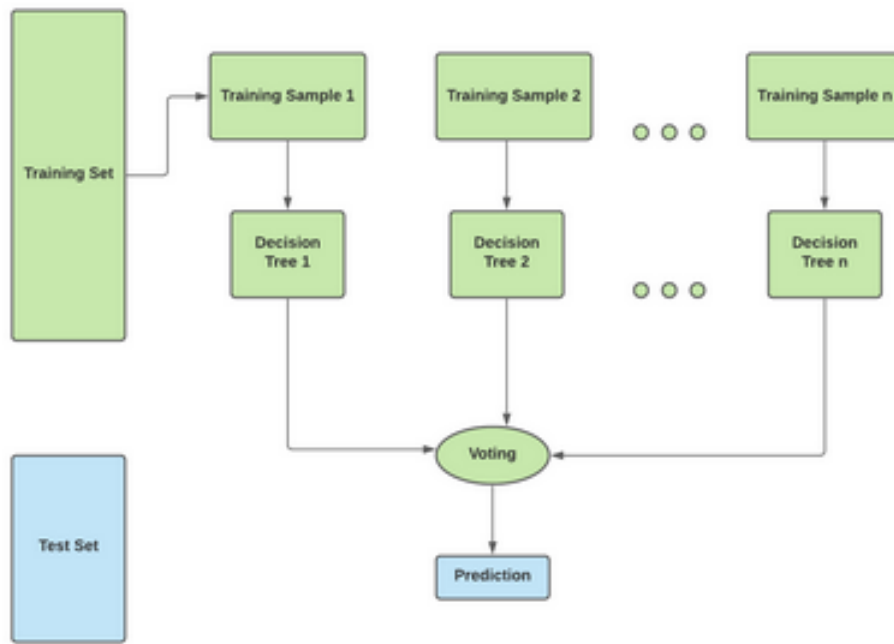


Figure 4.6: Working Model of Random Forest Classifier

#### 4.2.4 Logistic Regression

The logistic regression classifier is a binary classification discriminative model. It assesses the likelihood of something occurring. The target variable (or output)  $y$  can only accept discrete values for a given set of features (or inputs). The program builds a regression model to predict the chance that given data input will fall into the "1" category. The sigmoid function is used to describe the data in logistic regression, much as linear regression implies that the data follow a linear process –

$$g(z) = 1/(1 + e^{-z}) \quad (4.12)$$

In logistic regression, the dependent variable follows the Bernoulli Distribution.

- Maximum likelihood is used for estimation.
- Model fitness is calculated using Concordance, KS-Statistics, rather than R Square.

Logistic regression aims to determine event probabilities by establishing a relationship between variables and the greater chance of specific outcomes. [6]



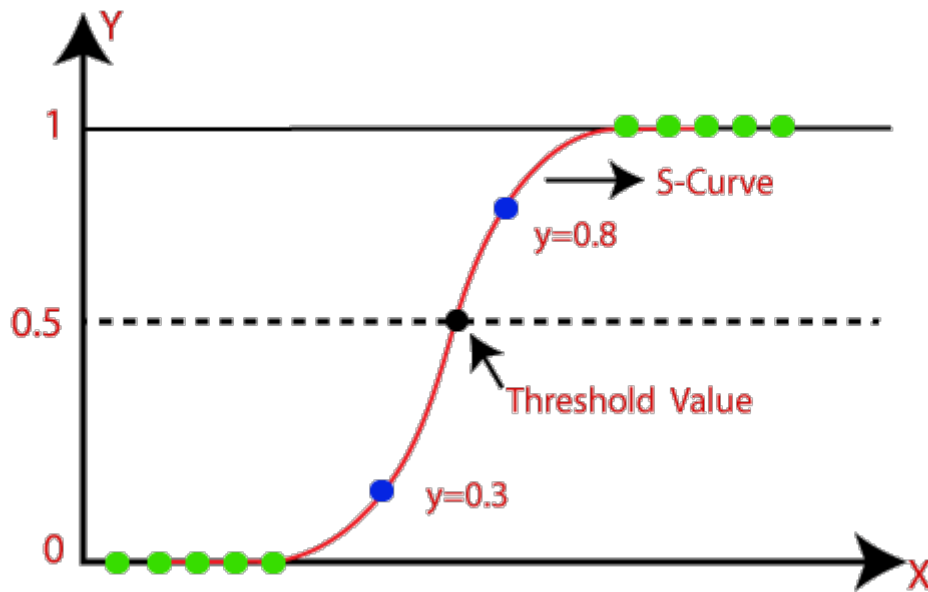


Figure 4.7: Working Model of Logistic Regression Classifier

#### 4.2.5 K-Nearest-Neighbors

K-Nearest-Neighbors (KNN) is a basic yet effective non-parametric supervised classification technique. The KNN algorithm is a sloppy learner's process in which prediction generation computation is performed till classification. KNN is a supervised learning method that classifies outcomes by relying on the majority of the votes from its K closest neighbor product segment. The KNN calculates the distance between the test data and all training samples. Any conventional method can be used to compute the distance. Because of its effective performance, efficient results, and simplicity, the KNN classifier is the most used pattern recognition classifier. In the KNN classification K-Nearest Neighbor, the unknown pattern is assigned the most common class among the classes of its nearest neighbors. One way for machine learning was aware of a method for categorizing objects based on datasets used as learning data. Its implementation was intended to utilize the closest distance or similarity to the item.

The same characteristics were computed for the test data during the classification step. The distance between this new vector and the learning data vector was determined. The nearest K number will then be used. The newly categorized points are expected to be included in most of these classifications. KNN is chosen as the classifier to be used in this approach. The KNN method is a simple algorithm with various applications, including image processing, face and character identification in the video, personal preference prediction of movies, music, product, medical, and genetic data pattern recognition. Because sentiment analysis involves a binary classification and vast datasets must be handled, KNN was chosen. The KNN classifier is fed a rating of the opinion phrase connected with a function in the evaluation. The classifier is trained in this scenario using a manually created training set. An X: Y relationship is supplied in the training set. In this case, x indicates an opinion term's score, and y reflects whether the word is favorable or unfavorable. Because it makes highly accurate predictions, the KNN algorithm can compete with the most

accurate models. The distance measure determines the efficiency of the calculations. As a result, the KNN algorithm is appropriate for applications with sufficient domain knowledge. [5]

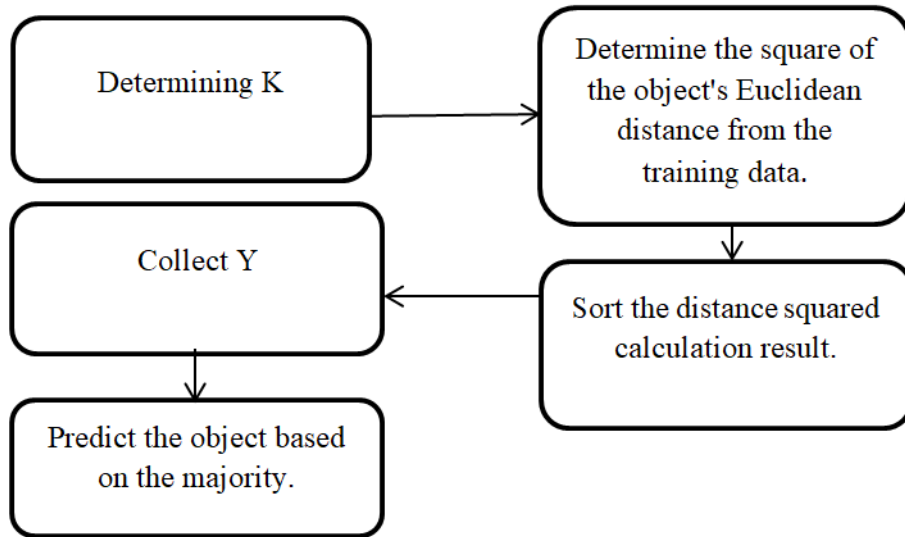


Figure 4.8: Steps for K-NN Analysis

The results of the accuracy, recall, f-score of each label, and the accuracy value of the k-Nearest Neighbor classification are given below:

Sentiment	Precision	Recall	F-Score
Negative	0.76	0.83	0.79
Positive	0.69	0.59	0.63
Accuracy	0.74		
Macro Average	0.72	0.71	0.71
Weighted Average	0.73	0.74	0.73

Table 4.4: Result Analysis of K-Nearest-Neighbors Classifier.

#### 4.2.6 Decision Tree

A simple form for classifying examples is a decision tree. It's a supervised strategy in which data is regularly separated depending on a specific parameter. Decision tree analysis may be used to tackle classification and regression issues. The decision tree algorithm splits a dataset into smaller subgroups while creating a decision tree in the background. A decision tree has nodes, branch nodes, and leaf nodes to build an entire structure. [24]

There is a root node in the figure where the decision tree starts. After that, it is divided into two or more sets. A leaf node is the final output node, and once you get the final node, you cannot split the tree any further.

#### 4.2.7 Gradient Boosting

The idea of the Gradient Boosting Classifier originated from AdaBoost Algorithm. This classifier was created through the iterative generation of weak learners while "boosting." Decision trees were the weakest learners in AdaBoost, and since they were so short, they were spread across a single split. A majority vote makes predictions of the predictions of weak learners based on the accurate weight of their predictions. [47] To produce a powerful prediction model, gradient boosting-based classifiers integrate different weak learning models, such as decision trees. These models excel at dealing with unstructured, complicated data. Gradient Boosting Machines (GBM) and other machine learning algorithms have shown state-of-the-art performance on various standard classification benchmarks. Gradient boosting machines' primary goal is to reduce the loss function, which is similar to gradient descent procedures in neural. When working iteratively, the weights of preceding learners are cemented in place as additional weak learners are introduced to the

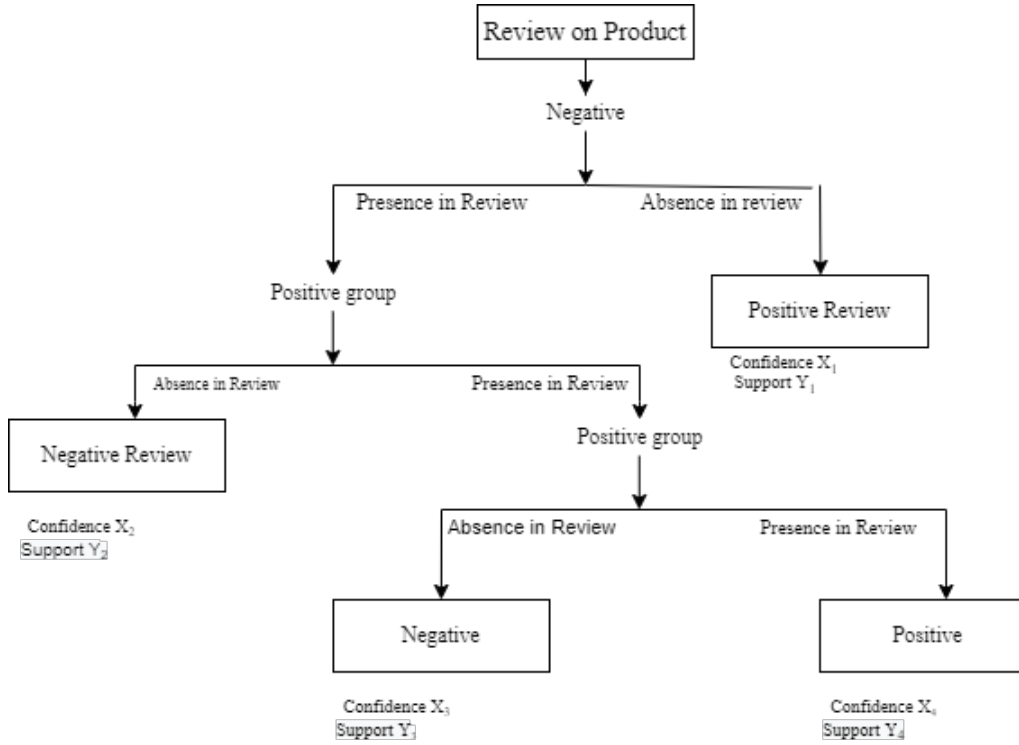


Figure 4.9: Working Model of Decision Tree Classifier

model, keeping the samples for the new layers constant. GBM may also be used to address problems involving multi-class classification and regression. Researchers recently discovered that the gradient boosting machine learning model beat sentiment analysis's support vector machine and logistic regression. Gradient Boosting machine learning is a method. [35]

Before applying the gradient descent process to an error function, the estimating position of a system  $F$  for an anticipated loss function  $g$  should be stated. When dealing with a data set in the form of -

$$X_i, Y_i (i = 1, \dots, n) \quad (4.13)$$

$Y_i$  is a continuous or discrete value response.  $X_i$  is a  $k$ -dimensional variable (the problem is referred to as regression in the case of constant values and classification in the case of discrete values). For simplicity, let's say that  $Y$  is multi-factorial. The goal is to minimize the anticipated loss using the function  $F: \mathbb{R}^k \rightarrow \mathbb{R}$  and expected loss  $g$ . [4]

$$E[g(Y, F(X))], g(., .) : \mathbb{R}^k \times \mathbb{R} \rightarrow \mathbb{R} \quad (4.14)$$

$$\hat{F}(x) = \arg \min_{F(x)} g(y, F(x)) \quad (4.15)$$

F and g, as previously stated, are the estimation and loss functions, respectively. The correlation of X and Y is expressed by the function F. The loss function g in the second argument should be smooth and convex to ensure that the procedure works appropriately. The loss function is determined by the domain in which we must deal. The system's characteristics (for example, whether it includes outliers or is high dimensional) substantially influence how the machine learning job should be approached.

#### 4.2.8 XGBoost

Extreme Gradient Boosting (XGBoost) is a more efficient and scalable gradient-based decision tree ensemble. This algorithm's primary purpose is to increase processing speed and efficiency. The result is produced at a slower rate and is rather lengthy since the Gradient Descent Boosting approach gradually analyzes the data set. As a result, XGBoost is employed to improve the model's performance significantly. [51] It has a linear model solver as well as tree learning algorithms. [51] Furthermore, the XGBoost method has various advantages and is especially well suited to the statistical analysis of big data sets. [45] XGBoost minimizes a loss function to provide an additive expansion of the objective function, similar to gradient boosting. Because XGBoost is solely interested in decision trees as base classifiers, the trees' complexity is regulated using a variant of the loss function. [36]

Xgboost's main characteristics are:

- It creates decision trees in parallel.
- Distributed computing technologies are being used to analyze large and complex models.
- Using Out-of-Core Computing to evaluate massive datasets.
- Cache optimization is used to make the most use of available resources.

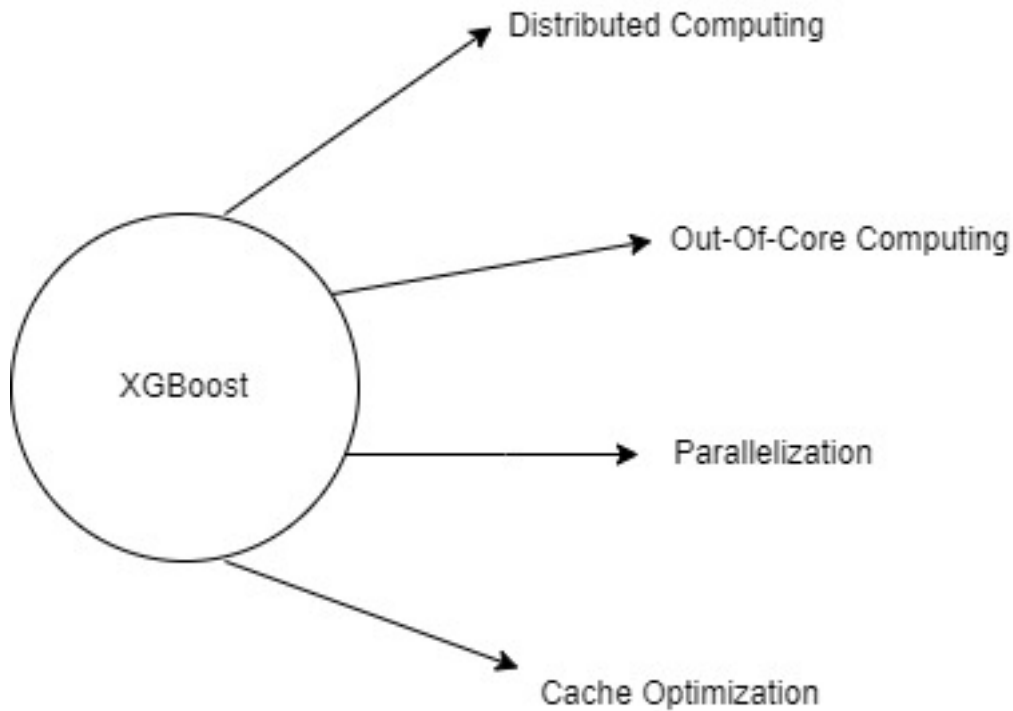


Figure 4.10: Xgboost’s principle features

### 4.3 Hyper-Parameters Tuning

Tuning hyperparameters is an essential aspect of managing the behavior of a machine learning model. Our predicted model parameters yield unsatisfactory results if our hyperparameters are not appropriately tuned since they do not minimize the loss function. This suggests that our model makes more mistakes. In practice, crucial measures such as accuracy and confusion matrix will be lower. We must identify between parameters and hyperparameters in machine learning. A learning algorithm learns or estimates model parameters for a given data collection, then updates these values as it learns. After learning is complete, these parameters are incorporated into the model. Each weight and bias in a neural network, for example, is a parameter. On the other hand, hyperparameters are algorithm-specific, and their values cannot be calculated from data. The model parameters are calculated using hyperparameters. Various hyperparameter values result in different model parameter values for a given data collection. Finding a set of ideal hyperparameter values for a learning algorithm and applying this improved algorithm to every data set is what hyperparameter tuning entails. This hyperparameter optimizes the model’s performance by minimizing a predetermined loss function, resulting in better outcomes with fewer mistakes. It should be noted that the learning algorithm optimizes the loss based on the input data and attempts to find an ideal solution within the constraints of the current setting. However, hyperparameters precisely define this arrangement.

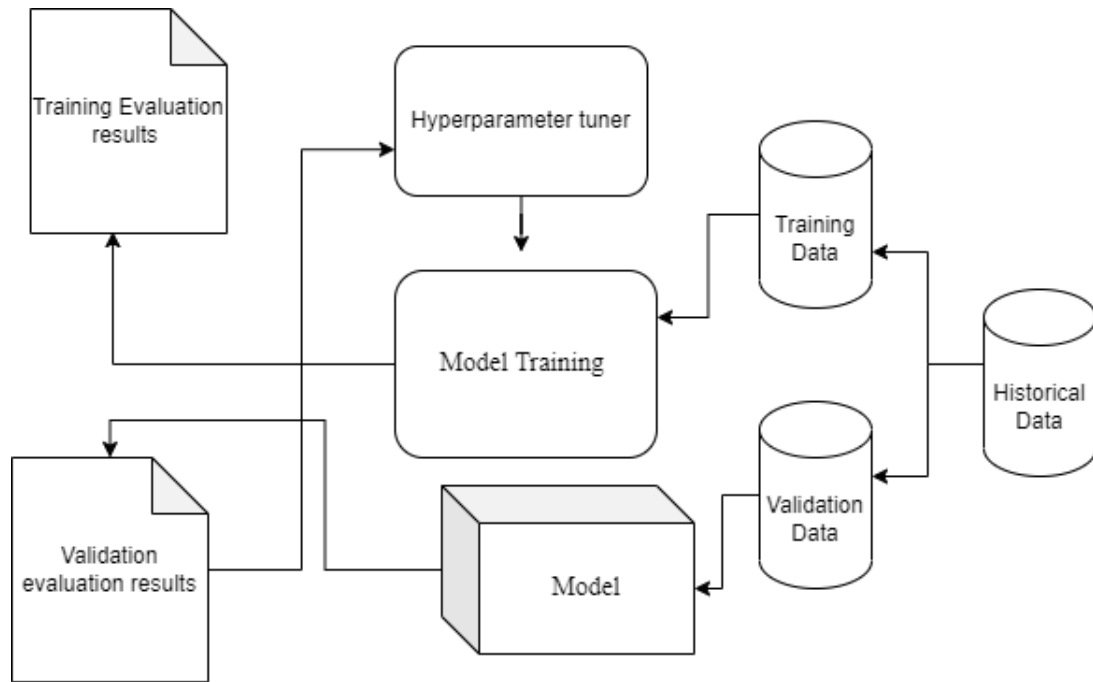


Figure 4.11: Working Model of Hyper-Parameters Tuning

### 4.3.1 Technique01: Manual Search

Manual tuning of hyperparameters does not require a specific library; instead, the programmer must attempt several combinations of hyperparameters for the model and pick the model that performs the best. Some of the hyperparameters that may be tuned for an XGB Classifier model include `maxDepth()`, `gamma()`, `regLambda()`, and `scalePosWeight()`. One may test all hyperparameter value combinations and train models for each, then select the model with the best performance. Another manual search strategy for hyperparameter optimization is to iterate over possible values of hyper-parameters and evaluate each combination. [18]

### 4.3.2 Technique02: Grid Search

Grid Search is a computerized counterpart of human search hyperparameter optimization. `GridSearchCV` is included in the Scikit-Learn package. `GridSearch` is not analytically friendly since it takes a long time to tune, but it eliminates the need to execute several lines of code. The `GridSeachCV` [39] function may be provided the training model and a set of hyperparameters in dictionary format, and it will return the performance model and its score metric.

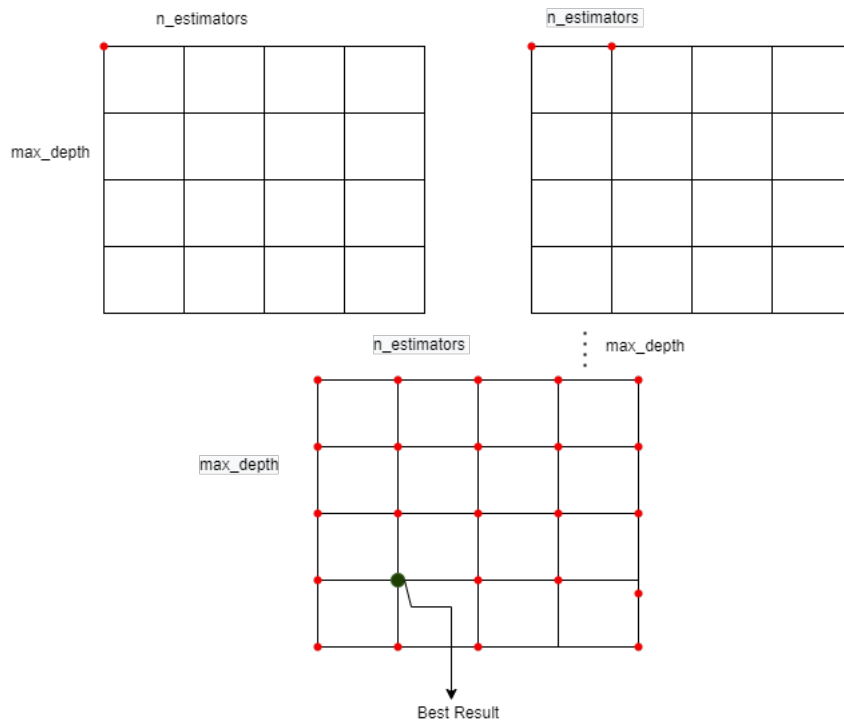


Figure 4.12: Working Model of Grid Search

### 4.3.3 Technique03: Randomized Search

Grid Search tries all possible hyperparameter combinations, enhancing the computation's time cost. A random hyperparameter combination is used to train a model in a randomized search. The overall number of options on which the various models are trained is reduced when randomized search [25] is used instead of grid search. The RandomSearchCV implementation is also included in the Scikit-Learn package. [25]

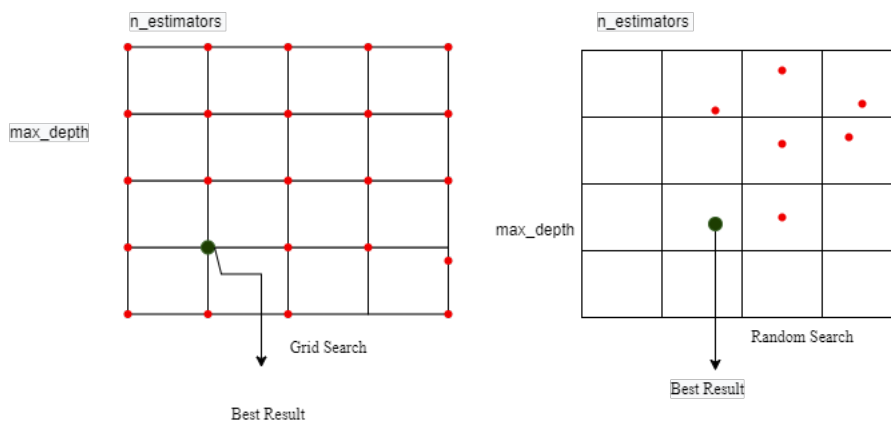


Figure 4.13: Working Model of Randomized Search



### 4.3.4 Technique04: Halving Grid Search

Grid Search Hyperparameter Optimization is an improved version of Grid Search hyperparameter optimization. Halving Grid Search runs across a specified list of hyperparameters using a consecutive halving strategy. The search approach begins by assessing all candidates on a small sample of data and then continuously picks the top candidates using bigger and larger data samples. [50]

The Grid Search technique is less computationally efficient than halving Grid Search. The Scikit-Learn package provides a `HalvingGridSearch` implementation.

### 4.3.5 Technique05: Halving Randomized Search

Halving Randomized Search applies the same sequential halving strategy as Halving Grid Search, but it is more effective. Unlike Halving Grid Search, it does not train on all combinations of hyperparameters but instead selects a selection of hyperparameter permutations randomly. The Scikit-Learn package also provides `HalvingRandomizedSearchCV`. [13]

### 4.3.6 Technique06: Hyperopt-Sklearn

- Hyperopt
- Scikit Optimize
- Optuna

Hyperopt is a Python accessible toolkit for Optimization techniques intended for massive optimizations of models with thousands of parameters. It enables hyperparameter optimization to be scaled over several CPU cores. Hyperopt-Sklearn is a Hyperopt library module that enables the automated search of machine learning algorithms and model hyperparameters for classification and regression problems. [23]

### 4.3.7 Technique07: Bayes Grid Search

Bayes Grid Search models the search space using the Bayesian optimization technique to arrive at optimum parameter values quickly. It improves time consumption by using the structure of the search area. The Bayes Search technique samples new candidates based on prior evaluation results to find the most likely to produce improved outcomes. The `BayesSearchCV` [20] implementation is included in the Scikit-Optimize module.

We need to tune the model hyperparameters to guarantee that our machine learning model and data provide good results. We have used all the seven techniques, but three hyperparameter tuning strategies like grid search, random search, and Bayesian optimization are below. If we need to evaluate our model quickly using

training data, we may use the grid search approach. Otherwise, we should use random search or Bayesian optimization to conserve time and computational resources.

Support Vector Machine	Manual Searching	88%
	Grid Searching	82%
	Randomized Searching	83%
	Bayesian Optimization	83%
Multinomial Naïve Bayes	Manual Searching	83%
Logistic Regression	Manual Searching	80%
	Grid Searching	96%

Table 4.5: Result of Hyper parameters tuning accuracy.

From the table, we can quickly get a clear idea of getting almost 96% accuracy on Logistic Regression.

# Chapter 5

## Analysis & Results

### 5.1 Analysis

Model assessment is critical for creating a predictive machine learning model. Building a predictive model without validating it is not a fit model, but a model that provides maximal accuracy is unquestionably good. You must observe the metrics and make adjustments as needed until you get the required accuracy rate. Confusion Matrix, Chi-Square, Cross-Validation Concordant-Discordant Ratio, Confidence Interval, Gini Co-efficient, Gain and Lift Chart, Kolmogorov-Smirnov Chart, Predictive Power, AUC-ROC Curve are some usual evaluation methods we use. For evaluating our model, we have come across with confusion matrix and K-Fold Cross-Validation.

#### 5.1.1 A. Confusion matrix

A Confusion matrix is a  $N \times N$  is a matrix used to evaluate the model. The matrix compares the actual goal values to the machine learning model's predictions. This provides us with a thorough view of how well our classification model performs and the sorts of errors it produces. [17]

##### **True Positive (TP):**

The actual result was positive, just as the model anticipated it would be.

##### **True Negative (TN):**

The theoretical and measured values are the same. The actual figure was negative, just as the model predicted it would be.

##### **False Positive (FP):**

Type 1 error occurs when the expected value is inaccurate. The model predicted a favorable outcome, while the actual work was negative. It is also known as the

Type 1 Error.

### **False Negative (FN):**

A Type 2 error occurs when the expected value is inaccurate. The model predicted a negative outcome, while the actual result was positive.

### **Precision vs. Recall**

Precision indicates how many of the accurately predicted situations were really positive. [11]

$$Precision = \frac{TP}{(TP + FP)} \quad (5.1)$$

The number of actual positive cases we could precisely identify using our model is called recall.

$$Recall = \frac{TP}{(TP + FN)} \quad (5.2)$$

### **F1-score**

F1-score is a harmonic mean of Precision and Recall, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to recall. [26]

$$F1-score = \frac{2}{\left(\frac{1}{Recall} + \frac{1}{Precision}\right)} \quad (5.3)$$

### **Accuracy**

The accuracy of a classification issue indicates the percentage of valid predictions. It is calculated by dividing the total number of forecasts by the total number of correct estimates.

$$Accuracy = \frac{(TP + TN)}{(TP + FN + FP + TN)} \quad (5.4)$$

Here,

$TP$  stands for True Positives.

$FP$  stands for False Positives.

$TN$  stands for True Negatives.

$FN$  stands for False Negatives.

### 5.1.2 B. Cross Validation:

Cross-validation is a statistical approach for estimating the ability of machine learning models. It is often used in applied machine learning to compare and select a model for a specific predictive modeling issue. It is simple to grasp, implement and produce skill estimates with lower bias than other approaches. [10]

The process contains a single parameter called  $k$  that specifies the number of groups into which a given data sample should be divided. As a result,  $k$ -fold cross-validation is a common name for the procedure. When a precise value for  $k$  is supplied, it can be substituted for  $k$  in the model's reference, such as  $k=10$  for 10-fold cross-validation.

In applied machine learning, cross-validation is commonly used to test a machine learning model's competency on unknown data. A survey method is often used to examine how the model performs. When information not used during training is used to create predictions on training.

It is a popular strategy because it is easy to grasp and produces a less biased or optimistic assessment of model competence than other methods, such as a simple train/test split.

The following is the general procedure:

- Randomly shuffles the dataset.
- Divide the data into  $k$  groups.
- For each distinct group:
  - Consider the group to be a holdout or test data set.
  - Consider the remaining groups to be a training data set.
  - Fit a model to the training data and then test it on the test data.
  - Keep the assessment score but throw away the model.
- Using the sample of model assessment scores, summarize the model's competence.

For each data sample, the  $k$  value must be carefully chosen. A poorly chosen value for  $k$  may result in an inaccurate representation of the model's competence, such as a score with a significant variance (which may fluctuate significantly depending on the data used to build the model) or a strong bias (such as an overestimate of the skill of the model).

Three popular methods for determining a value for  $k$  are as follows:

- **Representative:** The number of  $k$  is selected so that each train/test group of data samples is statistically representative of the larger dataset.

- **k=10:** The number fork is set at 10, a figure that has been established via testing to typically result in a model skill estimate with low bias and a moderate variation.
- **k=n:** The value for k is fixed to n, where n is the size of the dataset, to provide each test sample a chance to be utilized in the holdout dataset. This technique is known as leave-one-out cross-validation.

## 5.2 Result Analysis

The percentage of accuracy of Random Forest is 80%, SVM is 82%, XGBoost is 80%, Logistic Regression is 98%, Decision Tree is 69%, Gradient Boosting is 63%, K.N.N. is 46%, and Multinomial Naïve Bayes is 82%.

A bar chart of the accuracy of all classifiers is below -

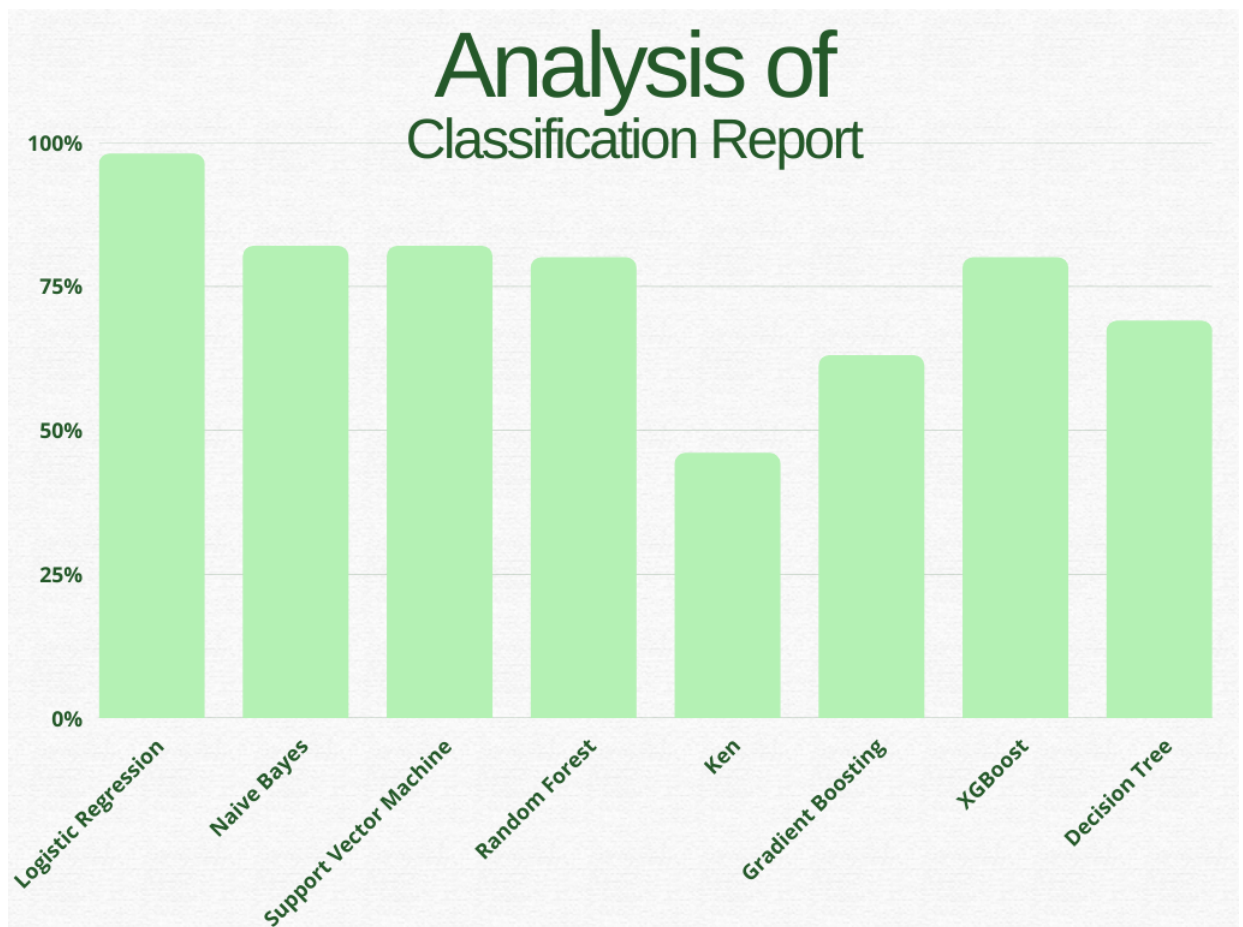


Figure 5.1: Result Analysis

Now, The percentage of precision and recall are below :

<b>Model</b>	<b>Accuracy (%)</b>	<b>Positive Recall (%)</b>	<b>Negative Recall (%)</b>	<b>Positive Pre- cision (%)</b>	<b>Negative Pre- cision (%)</b>
LR	98	96	99	99	98
KNN	46	97	16	40	89
GB	63	0	100	0	63
SVC	82	79	83	73	87
NB	82	79	83	73	87
RF	80	59	92	81	79
DT	69	24	96	77	68
XGB	80	73	85	80	80

Table 5.1: Percentage of Precision and Recall

Confusion Matrix of all the classifiers are below -



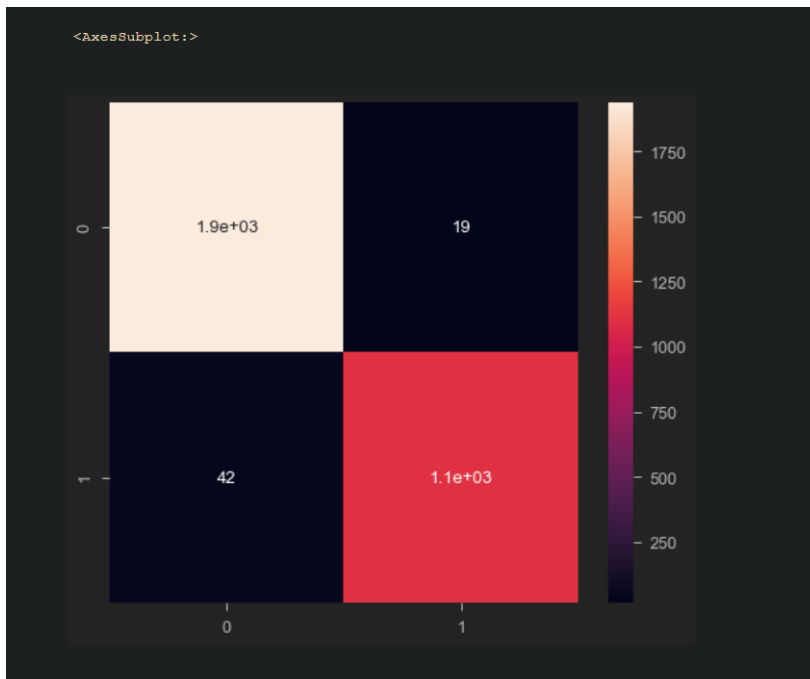


Figure 5.2: Confusion Matrix of Logistic Regression

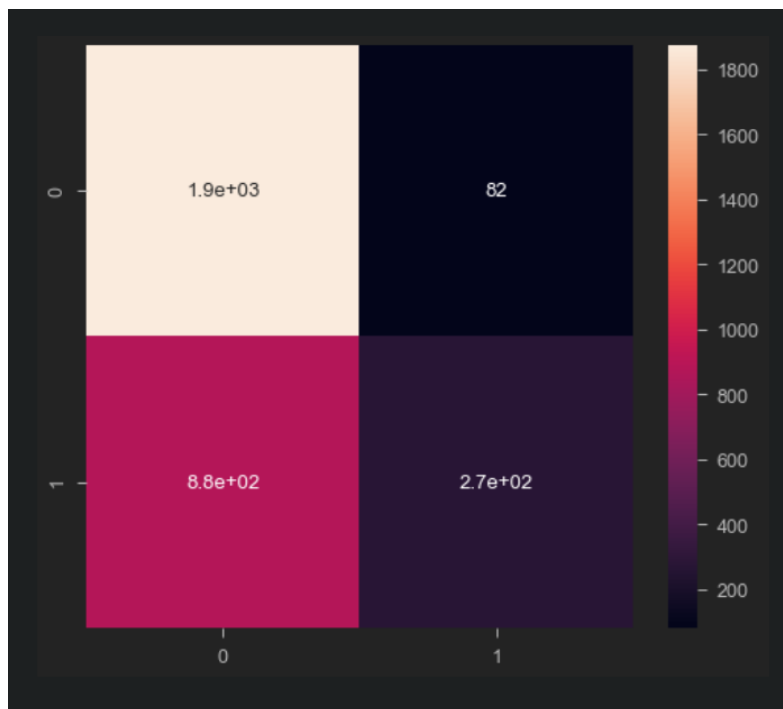


Figure 5.3: Confusion Matrix of Decision Tree

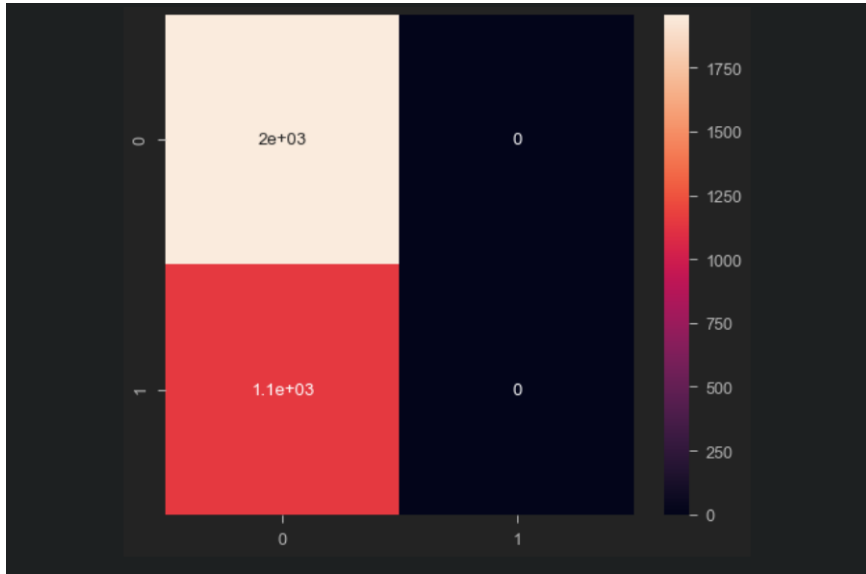


Figure 5.4: Confusion Matrix of Gradient Boosting

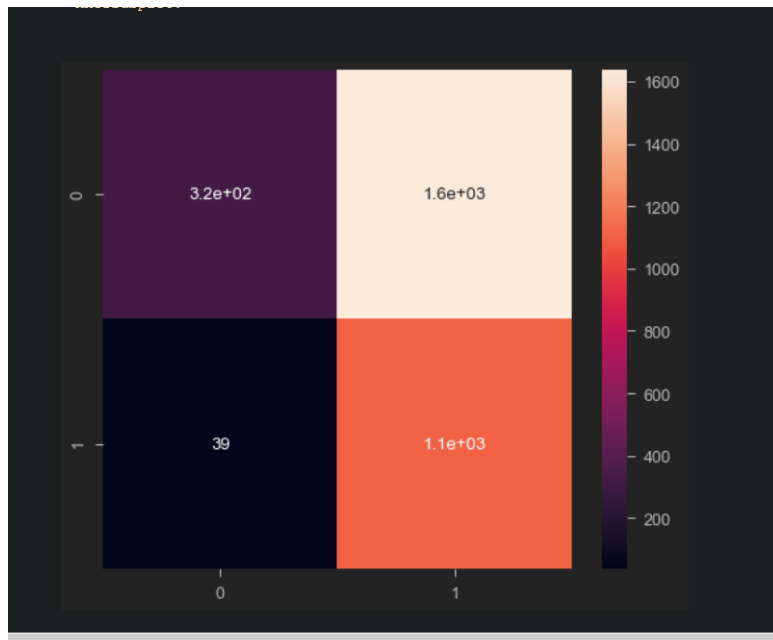


Figure 5.5: Confusion Matrix of KNN

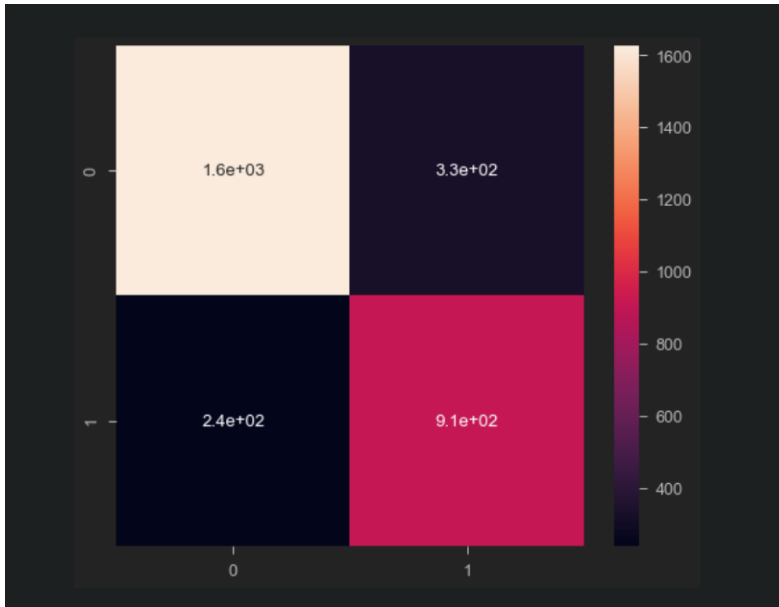


Figure 5.6: Confusion Matrix of Naive Bayes

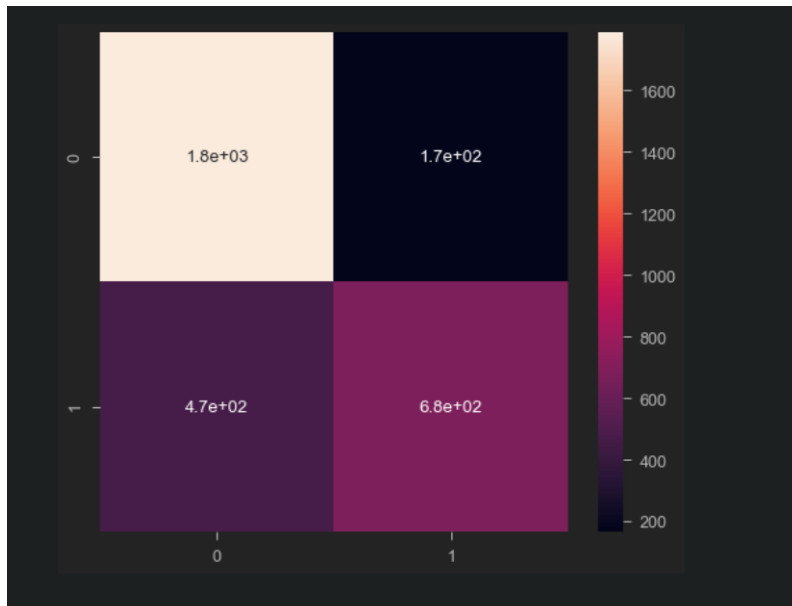


Figure 5.7: Confusion Report of Random forest

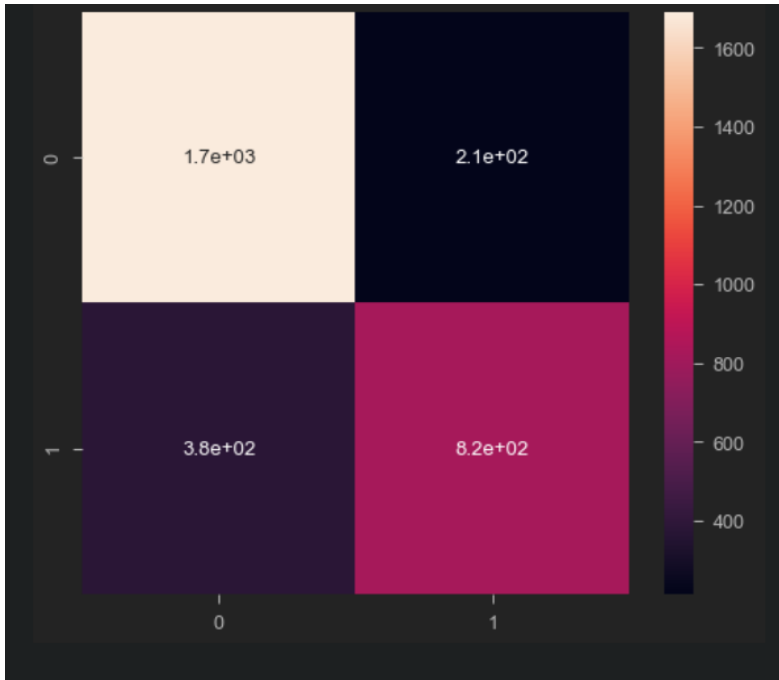


Figure 5.8: Confusion Matrix of SVM

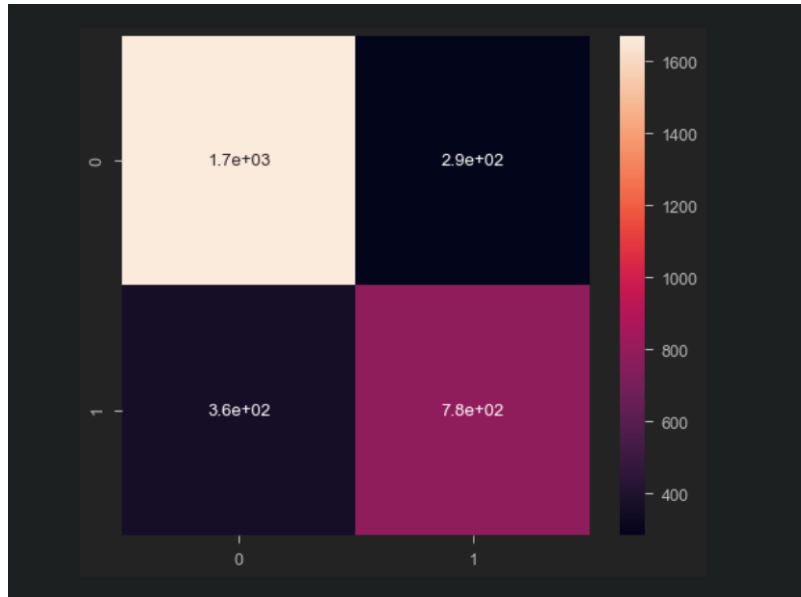


Figure 5.9: Confusion Matrix of XGBoost

we can see that among these eight classifiers of Machine Learning approaches, Logistic Regression is performing better accuracy, which is 98%, than the other seven classifiers. The K.N.N. classifier gives the lowest accuracy, which is 46.0%.

# Chapter 6

## Conclusion and Future Research

### 6.1 Conclusion

In today's data overload, environment companies can collect mountains of customer feedback. In this paper, our main objective is to detect the sentiment of reviews for different kinds of products. We used eight different supervised Machine Learning approaches, Naive Bayes Classifier, SVM, Random Forest Classifier, Logistic Regression (L.R.), K-nearest neighbors, XGBoost Classifier, Decision Tree Classifier, and gradient Boosting, to analyze which one gives perfect accuracy. We also conduct our research utilizing Ali-Express product reviews, employing a variety of N.L.P. approaches such as stop-word removal, stemming, word lemmatization, and TF-IDF vectorization. The accuracy and precision score of sentiment classification systems were investigated using our experimental methods.

Furthermore, all of our models could categorize negative and positive responses accurately and precisely. Among these eight classifiers of Machine Learning approaches, Logistic Regression performs better accuracy, 98%, than the other seven classifiers. The K.N.N. classifier gives the lowest accuracy, which is 46%. Logistic Regression outperforms the other seven Machine Learning techniques in terms of accuracy, with an accuracy of 98 percent. The K.N.N. classifier has the lowest accuracy (46%) of all the classifiers. The explanation for this could be attributed to particular stop words that generate a positive reaction and the fundamental class imbalance problem caused by the dataset's high proportion of favorable ratings.

### 6.2 Future Work

If we can collect many datasets in the future, we can achieve perfect accuracy and precision using Machine Learning approaches. Moreover, we want to expand this study to incorporate emojis in our texts in the future. The use of emojis in user-generated material has increased recently. All emoji are deleted from the texts during pre-processing. However, if emojis could be converted and processed, the accuracy of the predictions may be increased. Another way to improve our corpus would be to train using Word2vec, doc2vec, or paragraph2vec vectorization models rather than TF-IDF. In comparison to TF-IDF, these models take into account longer texts.

# References

- [1] T. U. Haque, N. N. Saber, and F. Shah, *Sentiment analysis on large scale amazon product reviews: Semantic scholar*, Jan. 1970. [Online]. Available: <https://www.semanticscholar.org/paper/Sentiment-analysis-on-large-scale-Amazon-product-Haque-Saber/58cfc4cb2644331eebf97db412f6d5f26b76188e>.
- [2] J. Webster and C. Kit, "Tokenization as the initial phase in nlp," Jan. 1992, pp. 1106–1110. DOI: 10.3115/992424.992434.
- [3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [4] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002, Nonlinear Methods and Data Mining, ISSN: 0167-9473. DOI: [https://doi.org/10.1016/S0167-9473\(01\)00065-2](https://doi.org/10.1016/S0167-9473(01)00065-2). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167947301000652>.
- [5] Y. Liao and V. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection11an earlier version of this paper is to appear in the proceedings of the 11th usenix security symposium, san francisco, ca, august 2002," *Computers Security*, vol. 21, no. 5, pp. 439–448, 2002, ISSN: 0167-4048. DOI: [https://doi.org/10.1016/S0167-4048\(02\)00514-X](https://doi.org/10.1016/S0167-4048(02)00514-X). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016740480200514X>.
- [6] J. Peng, K. Lee, and G. Ingersoll, "An introduction to logistic regression analysis and reporting," *Journal of Educational Research - J EDUC RES*, vol. 96, pp. 3–14, Sep. 2002. DOI: 10.1080/00220670209598786.
- [7] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*, Springer, 2005, pp. 878–887.
- [8] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 935–942.
- [9] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, IEEE, 2008, pp. 1322–1328.
- [10] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation.," *Encyclopedia of database systems*, vol. 5, pp. 532–538, 2009.

- [11] W. H. Walters, “Google scholar search performance: Comparative recall and precision,” *portal: Libraries and the Academy*, vol. 9, no. 1, pp. 5–24, 2009.
- [12] S.-J. Yen and Y.-S. Lee, “Cluster-based under-sampling approaches for imbalanced data distributions,” *Expert Systems with Applications*, vol. 36, no. 3, pp. 5718–5727, 2009.
- [13] A. Auger and B. Doerr, *Theory of randomized search heuristics: Foundations and recent developments*. World Scientific, 2011, vol. 1.
- [14] L. Liu, A. Wang, M. Sha, X. Sun, and Y. Li, “Optional svm for fault diagnosis of blast furnace with imbalanced data,” *ISIJ international*, vol. 51, no. 9, pp. 1474–1479, 2011.
- [15] H. M. Nguyen, E. W. Cooper, and K. Kamei, “Borderline over-sampling for imbalanced data classification,” *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011.
- [16] M. Rushdi Saleh, M. Martín-Valdivia, A. Montejó-Ráez, and L. Ureña-López, “Experiments with svm to classify opinions in different domains,” *Expert Systems with Applications*, vol. 38, no. 12, pp. 14 799–14 804, 2011, ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2011.05.070>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417411008542>.
- [17] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, “Confusion matrix-based feature selection,” *MAICS*, vol. 710, pp. 120–127, 2011.
- [18] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of machine learning research*, vol. 13, no. 2, 2012.
- [19] V. Rodríguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. Rigol-Sánchez, “An assessment of the effectiveness of a random forest classifier for land-cover classification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 93–104, 2012, ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2011.11.002>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271611001304>.
- [20] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” *Advances in neural information processing systems*, vol. 25, 2012.
- [21] R. Varghese and M. Jayasree, “Aspect based sentiment analysis using support vector machine classifier,” in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2013, pp. 1581–1586. DOI: 10.1109/ICACCI.2013.6637416.
- [22] B. Das, N. C. Krishnan, and D. J. Cook, “Racog and wracog: Two probabilistic oversampling techniques,” *IEEE transactions on knowledge and data engineering*, vol. 27, no. 1, pp. 222–234, 2014.
- [23] B. Komer, J. Bergstra, and C. Eliasmith, “Hyperopt-sklearn: Automatic hyperparameter configuration for scikit-learn,” in *ICML workshop on AutoML*, Citeseer, vol. 9, 2014, p. 50.
- [24] X. Fang and J. Zhan, “Sentiment analysis using product review data,” *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, 2015.



- [25] R. G. Mantovani, A. L. Rossi, J. Vanschoren, B. Bischl, and A. C. De Carvalho, “Effectiveness of random search in svm hyper-parameter tuning,” in *2015 International Joint Conference on Neural Networks (IJCNN)*, Ieee, 2015, pp. 1–8.
- [26] D. Zhang, J. Wang, and X. Zhao, “Estimating the uncertainty of average f1 scores,” in *Proceedings of the 2015 International conference on the theory of information retrieval*, 2015, pp. 317–320.
- [27] Z. Zheng, Y. Cai, and Y. Li, “Oversampling method for imbalanced classification,” *Computing and Informatics*, vol. 34, no. 5, pp. 1017–1037, 2015.
- [28] P. Baid, A. Gupta, and N. Chaplot, “Sentiment analysis of movie reviews using machine learning techniques,” *International Journal of Computer Applications*, vol. 179, pp. 45–49, Dec. 2017. DOI: 10.5120/ijca2017916005.
- [29] C. Chauhan and S. Sehgal, “Sentiment analysis on product reviews,” in *2017 International Conference on Computing, Communication and Automation (IC-CCA)*, 2017, pp. 26–31. DOI: 10.1109/CCA.2017.8229825.
- [30] D. Devi, B. Purkayastha, *et al.*, “Redundancy-driven modified tokek-link based undersampling: A solution to class imbalance,” *Pattern Recognition Letters*, vol. 93, pp. 3–12, 2017.
- [31] Y. Liu, J.-W. Bi, and Z.-P. Fan, “A method for ranking products through online reviews based on sentiment classification and interval-valued intuitionistic fuzzy topsis,” *International Journal of Information Technology Decision Making*, vol. 16, Sep. 2017. DOI: 10.1142/S021962201750033X.
- [32] R. Safrin, K. R. Sharmila, T. S. S. Subangi, and E. A. Vimal, “Sentiment analysis on online product,” 2017.
- [33] Z. Singla, S. Randhawa, and S. Jain, “Sentiment analysis of customer product reviews using machine learning,” Jun. 2017, pp. 1–5. DOI: 10.1109/I2C2.2017.8321910.
- [34] R. Ireland and A. Liu, “Application of data analytics for product design: Sentiment analysis of online product reviews,” *CIRP Journal of Manufacturing Science and Technology*, vol. 23, pp. 128–144, 2018, ISSN: 1755-5817. DOI: <https://doi.org/10.1016/j.cirpj.2018.06.003>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1755581718300336>.
- [35] A. Mayr, H. Binder, O. Gefeller, and M. Schmid, *The evolution of boosting algorithms*, Jan. 2018. [Online]. Available: <https://www.thieme-connect.com/products/ejournals/abstract/10.3414/ME13-01-0122>.
- [36] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, *A comparative analysis of xgboost*, Nov. 2019. [Online]. Available: <https://arxiv.org/abs/1911.01914v1>.
- [37] E. Coyne, J. Smit, and L. Güner, *Sentiment analysis for amazon.com reviews*, Mar. 2019. DOI: 10.13140/RG.2.2.13939.37920.
- [38] R. Jagdale, V. Shirsath, and S. Deshmukh, “Sentiment analysis on product reviews using machine learning techniques: Proceeding of cisc 2017,” in Jan. 2019, pp. 639–647, ISBN: 978-981-13-0616-7. DOI: 10.1007/978-981-13-0617-4\_61.

- [39] E. Ndiaye, T. Le, O. Fercoq, J. Salmon, and I. Takeuchi, “Safe grid search with optimal complexity,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 4771–4780.
- [40] B. Jonathan, P. H. Putra, and Y. Ruldeviyani, “Observation imbalanced data text to predict users selling products on female daily with smote, tomek, and smote- tomek,” in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, IEEE, 2020, pp. 81–85.
- [41] S. Sharma, I. Sharma, and A. Sharma, “Automated system for detecting mental stress of users in social networks using data mining techniques,” in Mar. 2020, pp. 769–777, ISBN: 978-3-030-43191-4. DOI: 10.1007/978-3-030-43192-1\_85.
- [42] B. Zhao, X. Zhang, H. Li, and Z. Yang, “Intelligent fault diagnosis of rolling bearings based on normalized cnn considering data imbalance and variable working conditions,” *Knowledge-Based Systems*, vol. 199, p. 105971, 2020.
- [43] M. T. Akter, M. Begum, and R. Mustafa, “Bengali sentiment analysis of e-commerce product reviews using k-nearest neighbors,” in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, 2021, pp. 40–44. DOI: 10.1109/ICICT4SD50815.2021.9396910.
- [44] P. Chitra, T. Karthik, S. Nithya, *et al.*, “Sentiment analysis of product feedback using natural language processing,” *Materials Today: Proceedings*, 2021, ISSN: 2214-7853. DOI: <https://doi.org/10.1016/j.matpr.2020.12.1061>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214785320407795>.
- [45] Y. Guang, “Generalized xgboost method,” *CoRR*, vol. abs/2109.07473, 2021. arXiv: 2109.07473. [Online]. Available: <https://arxiv.org/abs/2109.07473>.
- [46] D. Khyani and S. B S, “An interpretation of lemmatization and stemming in natural language processing,” *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology*, vol. 22, pp. 350–357, Jan. 2021.
- [47] P. Kumar and A. Wahid, “Social media analysis for sentiment classification using gradient boosting machines,” in *Proceedings of International Conference on Communication and Computational Technologies*, S. Kumar, S. D. Purohit, S. Hiranwal, and M. Prasad, Eds., Singapore: Springer Singapore, 2021, pp. 923–934, ISBN: 978-981-16-3246-4.
- [48] N. M. Mqadi, N. Naicker, and T. Adeliyi, “Solving misclassification of the credit card imbalance problem using near miss,” *Mathematical Problems in Engineering*, vol. 2021, 2021.
- [49] S. Saxena, *Tokenization and text normalization*, Mar. 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/03/tokenization-and-text-normalization/>.
- [50] B. R. R. .-. II, *Halving gridsearch*, Apr. 2022. [Online]. Available: <https://bobrupakroy.medium.com/halving-gridsearch-736b13898327>.

- [51] Z. Lateef, *A beginners guide to boosting machine learning algorithms*, Jan. 2022. [Online]. Available: <https://www.edureka.co/blog/boosting-machine-learning/>.