A

Project Report

on

SENTIMENT ANALYSIS USING MACHINE LEARNING

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2023-24

in

COMPUTER SCIENCE & ENGINEERING

By

Vishu Agarwal (2000290100192)

Ashmit Tayal (2000290100033)

Rajani Dixit (2000290100110)

Under the supervision of

Ms. Bharti Chugh

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2024

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Name:

Vishu Agarwal (2000290100192)

Ashmit Tayal (2000290100033)

Rajani Dixit (2000290100110)

Date:

CERTIFICATE

This is to certify that Project Report entitled “SENTIMENT ANALYSIS USING ML” which is submitted by Vishu Agarwal, Ashmit Tayal, Rajani Dixit in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Ms. Bharti Chugh **Dr. Vineet Sharma**

(Assistant Professor) **(Head of Department)**

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Ms. Bharti Chugh, Department of Computer Science & Engineering, KIET, Ghaziabad, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/ industry person/ any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date:

Name:

Vishu Agarwal (2000290100192)

Ashmit Tayal (2000290100033)

Rajani Dixit (2000290100110)

ABSTRACT

Data Analysis has become an important part of our life, as it helps us to draw useful information, decision making conclusions on any particular raw data. One component of digital data analysis is sentiment analysis. Sentiment analysis evaluates the emotional tone of textual data and classifies it as positive, negative or neutral.

Sentiment analysis, a cornerstone of Natural Language Processing (NLP), is instrumental in distilling subjective sentiments and opinions from vast repositories of textual data. In an era dominated by online communication, understanding sentiment has far-reaching implications across domains, including marketing, customer service, political analysis, and beyond. This study embarks on an extensive exploration of sentiment analysis, employing a diverse array of machine learning techniques to analyze and classify textual data across various contexts.

In business, this analysis is frequently utilized for market social media monitoring, market research, consumer feedback and many more. In order to automate the procedure, machine learning models that recognize patterns in labeled data are frequently used.

The objective is to derive meaningful insights from textual data so that decisions can be made with knowledge, effectively. In order to provide important insights into public opinion and behavior, this study uses machine learning techniques to investigate the sentiments conveyed in Twitter data. A wide range of Twitter posts, including ones about trending hashtags, events, and subjects, are gathered as part of the process.

Sentiment analysis is done using machine learning algorithms. Sentiment analysis and examples of human language are used to train machine learning models. Following this training, machine learning is used by sentiment analysis software to evaluate and rank human language according to its previous training.

The current study proposes a technique that best analyzes the sentiments of the text used in social media is gradient boosting classifier. With this method, an additive model can be built in a step-by-step, forward fashion and any differentiable loss function can be optimized.

There are n-class regression trees that are fitted on the negative gradient of the loss function, which could be the multiclass log loss or the binary loss. This process is repeated in each step. In the particular case of binary classification, just one regression tree is induced.

The computational results demonstrates that gradient boosting approach with an accuracy of 93.62% which outperforms is preferable to other classifiers.

Keywords— Sentiment analysis, Twitter, Dataset, Machine learning, Gradient boosting classifier,

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ML	Machine Learning
SA	Sentiment Analysis
NLP	Natural Language Processing
GBC	Gradient Boosting Classifier
SVM	Support Vector Machines
LR	Logistic Regression
NB	Naïve Bayes
kNN	k- Nearest Neighbour Classifier
LDA	Linear Discriminant Analysis

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

Sentiment analysis, a fascinating domain within machine learning, delves into the realm of understanding and extracting emotional nuances from textual data. By leveraging advanced algorithms and natural language processing techniques, sentiment analysis meticulously examines the intricacies of human language to decipher underlying emotions and perceptions. Furthermore, this sophisticated analytical process utilizes semantic context to gauge the public sentiment surrounding a particular brand and predict the market success of its offerings. To enhance the accuracy and efficacy of sentiment analysis, extensive datasets are employed for training various machine learning models such as Naïve Bayes, K-nearest neighbor classifier, Support Vector Machines, and many more. These models are adept at recognizing intricate patterns and uncovering meaningful correlations between specific words and associated sentiments. Moreover, the comprehensive evaluation of these models includes performance metrics like accuracy, F1 score, precision, and other relevant indicators to assess their effectiveness in sentiment analysis tasks.

The study delves into various challenges, such as effectively handling sarcasm, understanding slang expressions, and interpreting context-specific terms to gain a deeper understanding of public sentiment on Twitter. The insights gained from sentiment analysis have valuable implications across diverse sectors including market research, brand management, and political analysis, offering a wealth of benefits to businesses and scholars alike. By tapping into the vast array of opinions expressed on the Twitter platform, this analysis provides businesses and academics with invaluable and perceptive information to drive strategic decision-making and enhance their understanding of public sentiment and trends. One key potential application of this research is the ability to monitor public sentiment in real-time,

facilitating swift and proactive responses and strategic decision-making measures within various sectors. This can prove invaluable for organizations seeking to stay ahead of trends and effectively manage their public relations efforts.

In this era characterized by an overwhelming influx of information, the sheer volume of textual data being produced on diverse online platforms poses both a formidable challenge and a promising opportunity. Within the vast expanse of this digital landscape, there exists a rich reservoir of valuable insights pertaining to human emotions, viewpoints, and stances. Sentiment analysis, which stands as a crucial branch within the realm of natural language processing (NLP), serves as a powerful tool for extracting and quantifying these personal expressions intricately woven into written content. The profound significance of comprehending sentiment reverberates across a multitude of sectors, encompassing areas such as marketing strategies, customer interactions, political assessments, and beyond. The utilization of advanced machine learning methodologies in the realm of sentiment analysis has now surfaced as a compelling pathway to streamlining and enhancing the process of deciphering sentiments on a large scale.

Aggarwal et al. (2011) conducted an in-depth study that focused on the exploration of twitter that led them to devise sophisticated models designed to effectively categorize the myriad of "tweets" into distinct sentiment categories, encompassing the realms of positivity, negativity, and neutrality. The significance of sentiment analysis, in data management, cannot be understated as it offers a cost-effective and efficient means to process large volumes of information in real-time scenarios. Leveraging sentiment analysis allows for the swift resolution of various real-world situations, making it a pivotal tool across a wide array of industries. Businesses, for instance, rely heavily on sentiment analysis to dissect and comprehend customer feedback, bolster their brand image, enhance product quality, and deliver personalized content that resonates with their target audience.

Moreover, the utilization of sentiment analysis extends beyond the business landscape, extending its benefits to fuel market research endeavors, steer policymaking decisions, and drive automation towards greater efficiency and accuracy. The assessment of textual sentiment stands as a crucial process, facilitated by the application of metrics that classifies expressions

as positive, negative, or neutral. The evolving digital landscape has introduced diverse avenues for individuals to express their emotions online, prompting the need for automated systems that can decipher sentiments without human intervention. This is where machine learning steps in, enabling computers to assimilate vast datasets of emotional text samples, thus empowering them to identify and analyze sentiments with remarkable precision.

Furthermore, sentiment analysis models have transcended conventional boundaries, acquiring the capacity to comprehend context, detect nuances like misapplied words and sarcasm, and evolve beyond rigid definitions. The implementation of a variety of methodologies, alongside intricate equations and a systematic training regime, equips machines with the requisite skills to deliver insightful sentiment analysis results. While the process may seem complex, the amalgamation of these strategies yields impactful outcomes, illustrating the profound capabilities of sentiment analysis in the modern era.

In machine learning, the use of natural language and sentiment plays a crucial role in training models. After the initial training process, sentiment analysis software makes use of machine learning techniques to assess and assign scores to human language. Various algorithms, such as Linear Regression, Support Vector Machines (SVM), and Naïve Bayes, are extensively employed for this purpose. Each new model created is designed to minimize prediction errors by refining its decision-making within the vast space of potential predictions for every example included in the training set. This iterative process ensures that the models continuously improve in their ability to interpret and process language in an accurate and efficient manner.

Wang et al. (2022) explains that opinion mining and sentiment analysis, both fascinating fields of study, delve into deciphering people's perceptions and emotions towards specific topics. These two terms are often used interchangeably to depict the process of gauging public attitudes and feelings. Moreover, Chalothom et al. (2015) highlights the newfound interest in analyzing twitter content to determine the sentiment's positivity or negativity. Our experiments with sentiment analysis on tweet data have revealed that employing a group learning approach, which combines opinions from various classifiers such as stacking, support vector machine, SentiStrength, and naïve bayes through majority voting, can significantly enhance accuracy

levels. This amalgamation of algorithms has proven to be a more efficient strategy to achieve superior performance outcomes in sentiment analysis tasks involving tweets. By leveraging this combined approach, we have observed a notable improvement in the accuracy and overall success of sentiment analysis in the context of Twitter data.

Xia et al. (2016) aims to provide insight into the vital role played by popular text classification techniques such as maximum entropy, support vector machines, and naïve bayes and articulate why these techniques are indispensable. Saif et al. (2012) emphasizes the fact that the inherent characteristics of microblogs, like Twitter, characterized by their brevity and asymmetrical structure, pose numerous challenges that need to be tackled when conducting sentiment analysis. These obstacles include the need for adapting traditional methods to handle the unique nature of microblog data for accurate sentiment interpretation.

In Gautam et al.'s (2014) study, they extensively investigate the evolving landscape of online content across various domains. This analysis reveals a significant surge in the volume and diversity of websites focusing on specific product categories and proficiently aggregating customer feedback from platforms like Amazon and Twitter. Neelakandan et al. (2020) further elaborates on the methodology of sentiment analysis applied to data, particularly in evaluating reviews to determine their positive, negative, or neutral nature according to the expressed viewpoints. The primary objective of sentiment analysis (SA) is to identify the stance of review writers on multiple levels, be it at the document, sentence, or even aspect level within the context of online content evaluation.

These days, sentiment analysis (SA) is a valuable tool commonly used to evaluate reviews across various data domains. These domains encompass a wide range of industries, such as products, movies, hotels, restaurants, and more. In a study by Le et al. (2015), it was found that tweets on Twitter serve as an effective platform for users to freely express their opinions on diverse topics. Through this platform, marketers can strategically tailor their campaigns to align with consumers' viewpoints on products, companies, incidents of bullying, as well as current events, enabling them to gain deeper insights into public sentiment and preferences.

1.2. PROJECT DESCRIPTION

a) Creating an efficient sentiment analysis model for the twitter dataset, which can accurately identify the sentiments, moods, effects and biasness.

b) Assessing the effects of various resampling and optimization strategies on the overall performance of sentiment analysis model in order to determine the most effective classifier technique for enhancing its overall performance.

c) Examining and adjusting the model's hyperparameters for better results and accuracy.

CHAPTER 2

LITERATURE REVIEW

With the growing development in the field of social media alongside machine learning, various experiments amd research has been carried out in these recent years releasing the relevant significant papers. The following literature survey provides an overview of relevant studies amd research works conducted in the field of sentiment analysis prediction using machine learning algorithms. These studies have contributed to the understanding of risk factors, model development, and performance evaluation, providing valuable insights and benchmarks for the present report.

Sahayak et al. (2015) state that sentiment analysis has gained popularity as a field of study in computational linguistics due to the proliferation of sentiment data from blogs, online forums, and social media sites like Facebook and Twitter.

Neethu et al. (2013) state that the sentiment analysis include four phases: Collecting real-time tweets up to a given limit, tokenizing every tweet as part of pre-processing, comparing them with an available bag of words, and classifying the tweets as positive or negative. It includes the process of locating and categorizing viewpoints or feelings represented in the source text is known as sentiment analysis. By using sentiment analysis in a particular domain, the influence can be determined by information on sentiment classification.

Sankar et al. (2017) addresses several issues and obstacles in the field of sentiment analysis and presents an array of methodologies and concerns associated with the field.

Alsaeedi et al. (2019) presented that over the past 10 years, opinion study of Twitter data has received a lot of attention. This type of investigation involves examining the words that make up "tweets," or remarks. Because of this, the purpose of this study is to investigate the

numerous sentiment analyses that were performed on Twitter data and the results of those analyses.

Kharde et al. (2016) provide a survey and an examination of the most recent developments in option mining techniques, including lexicon-based approaches and machine learning, is presented in this comparative analysis. Also provide a study on twitter data streams by employing a number of different machine learning algorithms, such as Naïve Bayes, Support Vector Machine and Max Entropy.

Neelakandan et al. (2020) proposes that using a gradient-boosted decision tree classifier, sentiment analysis and sentiment classification of Twitter data are conducted.

Anjaria et al. (2014) suggested a hybrid method of opinion extraction from Twitter data that combines characteristics that are both direct and indirect. According to this method supervised classifiers that consist of a network of artificial neural connections, Naïve Bayes, Maximum Entropy and Support Vector Machines (SVM).

Jain et al. (2016) states Sentiment analysis is the method of automatically determining if an entity (i.e., product, people, topic, event, etc.) is the topic of either neutral, negative, or positive opinion expression in a user-generated text. The purpose for conducting this research is to offer a thorough examination of machine learning technique for sentiment analysis utilizing data from Twitter.

Da Silva et al. (2014) provides a method that employs lexicons and classifier ensembles to automatically categorize the tone expressed in tweets. Tweets on a query word are categorized as good or bad. This method has numerous applications, including helping businesses track public opinion about their brands and consumers who can use sentiment analysis to look for products.

Le et al. (2015), it was found that tweets on Twitter serve as an effective platform for users to freely express their opinions on diverse topics. Through this platform, marketers can strategically tailor their campaigns to align with consumers' viewpoints on products, companies, incidents of bullying, as well as current events, enabling them to gain deeper insights into public sentiment and preferences

Jagdale et al. (2019) states that they used machine learning methods to categorize reviews as favourable or negative after preprocessing. The conclusion of the paper states that the most accurate classification results for product reviews are achieved through the application of machine learning techniques.

Singh et al. (2017) presents that people's opinions about governments, events, goods and services on social media are expressed through words and phrases. Sentiment analysis in natural language processing is the technique of extracting positive and negative polarities from text that is shared on social media platforms.

Ain et al. (2017) proposes that online user sentiments have a significant impact on lawmakers, goods suppliers, and readers. Sentiment analysis has garnered substantial attention as a means of analyzing and organizing the unstructured data gleaned from social media.

Rathi et al. (2018) focuses on the categorization of tweets' emotional content using information obtained from Twitter. In the past, researchers used machine learning techniques that were already available for sentiment analysis, but the findings indicated that these methods were not producing superior sentiment categorization results.

Pang and Lee (2002) proposed the framework, where an assessment can be positive or negative was discovered by the proportion of positive words to total words. Later in 2008, a methodology was proposed in which tweet results can be chosen by term in the tweet.

CHAPTER 3

PRELIMINARIES

3.1. BENCHMARK DATASET

Initially, we collected a dataset for our twitter data sentiment analysis prediction. The dataset used for research and analysis are collected from-

https://www.kaggle.com/datasets/durgeshrao9993/twitter-analysis-dataset-2022

	A	B	C
1	id	label	tweet
2	1	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
3	2	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
4	3	0	bihday your majesty
5	4	0	#model i love u take with u all the time in urÃ°ÃŸÃ“Â±!!! Ã°ÃŸÃ˜Ã™Ã°ÃŸÃ˜ÃŽÃ°ÃŸÃ‘Â„Ã°ÃŸÃ‘Â…Ã°ÃŸÃ‘Â¦Ã°ÃŸÃ‘Â¦Ã°ÃŸÃ‘Â¦
6	5	0	factsguide: society now #motivation
7	6	0	[2/2] huge fan fare and big talking before they leave. chaos and pay disputes when they get there. #allshowandnogo
8	7	0	@user camping tomorrow @user @user @user @user @user @user @user dannyÃ¢Â€Â¦
9	8	0	the next school year is the year for exams.Ã°ÃŸÃ˜Â¨ can't think about that Ã°ÃŸÃ˜Â- #school #exams #hate #imagine #actorslife #revolutionschool #girl
10	9	0	we won!!! love the land!!! #allin #cavs #champions #cleveland #clevelandcavaliers Ã¢Â€Â¦
11	10	0	@user @user welcome here ! i'm it's so #gr8 !
12	11	0	Ã¢Â†Â #ireland consumer price index (mom) climbed from previous 0.2% to 0.5% in may #blog #silver #gold #forex
13	12	0	we are so selfish. #orlando #standwithorlando #pulseshooting #orlandoshooting #biggerproblems #selfish #heabreaking #values #love #
14	13	0	i get to see my daddy today!! #80days #gettingfed
15	14	1	@user #cnn calls #michigan middle school 'build the wall' chant '' #tcot
16	15	1	no comment! in #australia #opkillingbay #seashepherd #helpcovedolphins #thecove #helpcovedolphins
17	16	0	ouch...junior is angryÃ°ÃŸÃ˜Â#got7 #junior #yugyoem #omg
18	17	0	i am thankful for having a paner. #thankful #positive
19	18	1	retweet if you agree!
20	19	0	its #friday! Ã°ÃŸÃ˜Â€ smiles all around via ig user: @user #cookies make people
21	20	0	as we all know, essential oils are not made of chemicals.
22	21	0	#euro2016 people blaming ha for conceded goal was it fat rooney who gave away free kick knowing bale can hit them from there.
23	22	0	sad little dude.. #badday #coneofshame #cats #pissed #funny #laughs
24	23	0	product of the day: happy man #wine tool who's it's the #weekend? time to open up & drink up!
25	24	1	@user @user lumpy says i am a . prove it lumpy.
26	25	0	@user #tgif #ff to my #gamedev #indiedev #indiegamedev #squad! @user @user @user @user @user

Table 1: Dataset used

Column Name	Attribute Description
ID (numerical)	Serial number
Label (binary)	Good or Bad
Tweet (categorial)	Twitter post

Table 2: Dataset Attributes

After the collection of the dataset, the whole dataset is divided into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of data is used for training purpose and 30% of data is used for testing.

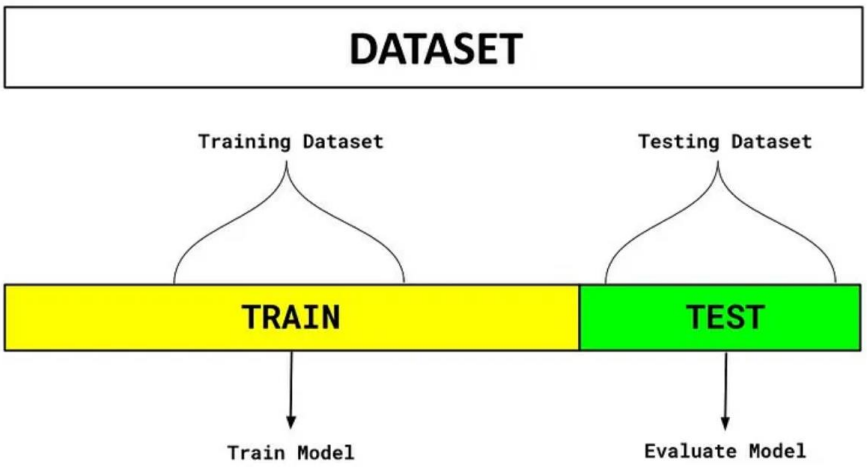

Figure 1: Training and Testing Data

3.2. EXPLORATORY DATA ANALYSIS

In order to construct predictive models, exploratory data analysis, or EDA, is requisite to machine learning since it helps to comprehend the traits, connections, and patterns found in the dataset. To acquire insights, spot problems with data quality, and guide preparation actions for the best possible model performance, it makes use of descriptive statistics, data visualization, and feature analysis.

Table given below gives the detailed description of the dataset, including target value i.e., “label” (binary value) and the number of data used.

	Description	Value
0	Session ID	123
1	Target	label
2	Target Type	Binary
3	Original Data Shape	(31962, 2)
4	Transformed Data Shape	(31962, 2)
5	Transformed Train Data Shape	(22373, 2)
6	Transformed Test Data Shape	(9589, 2)
7	Numeric Features	1
8	Preprocess	TRUE
9	Imputation Type	Simple
10	Numeric Imputation	Mean
11	Categorial Imputation	Mode

Table 3: Dataset Description

3.3. DATA PREPROCESSING

A tweet with a positive sentiment is represented by a value of 1, whereas a tweet with negative emotion is represented by a value of 0. Table 2 describes each attribute of the dataset that is ID is used as the serial number which contains the numeric values, The label contains binary values (good or bad) and the tweets are the categorial values.

Data preprocessing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes.

In pre-processing of data, we transform data into our required format. It is used to deal with noisses, duplicates, and missing values in the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

To ensure data quality, the dataset was carefully examined for missing, duplicate, and trash values before being subjected to additional analysis. In order to get the dataset ready for machine learning algorithms, the z-score method was utilized to standardize the data by taking the mean value (μ) and dividing it by the standard deviation (σ) of each feature as shown in equation 1. The label encoding method was then used to convert the categorial features into numerical representations.

$$z = \frac{(x - \mu)}{\sigma} \tag{1}$$

The data is then centered around zero and its standard deviation is measured to be one, making appropriate for ML algorithms that require normalized data. This technique makes the data suitable for machines.

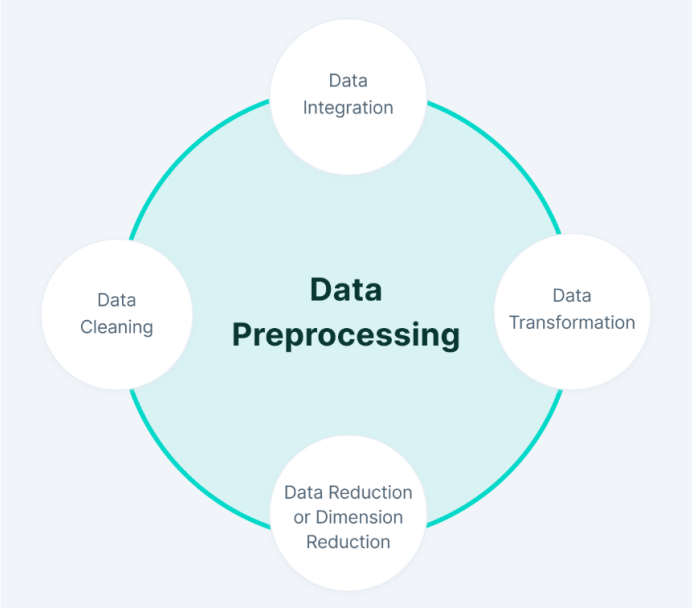

Figure 2: Data Preprocessing

CHAPTER 4

MACHINE LEARNING MODELS

4.1. LOGISTIC REGRESSION

Logistic regression is a statistical method used for binary classification, which means predicting the probability of an observation belonging to one of two classes. It's called "logistic" because it models the probability using the logistic (sigmoid) function.

In logistic regression, you typically have one or more independent variables (features) and a dependent variable (target) that is binary. The model estimates the probability that a given input belongs to the positive class (usually denoted as 1), based on the values of the independent variables. Equation 1 gives the formula logistic regression model as:-

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}} \quad (2)$$

Where:

- P (Y=1|X) is the probability of the dependent variable (Y) being 1 given the independent variables (X).
- $\beta 0$, $\beta 1$, $\beta 2$, ..., βn are the coefficients of the model.
- $X1$, $X2$, ..., Xn are the values of the independent variables.
- e is the base of natural logarithm.

Logistic regression estimates the coefficients (β) of the independent variables using maximum likelihood estimation. Once the coefficients are estimated, they can be used to make predictions by plugging the values of independent variables into the logistic function.

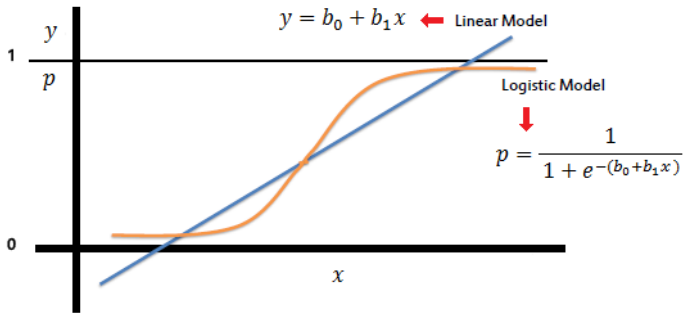

Figure 3: Logistic Regression Graph

There are three types of logistic regression models, which are defined as:-

a) **Binary Logistic Regression:** In this approach, response or dependent variable is dichotomous in nature- i.e. it has only two possible outcomes (e.g. 0 or 1). Some popular examples of its use include predicting if an e-mail is spam or not spam or if a tumor is malignant or not malignant.

b) **Multinomial logistic regression:** In this type of logistic regression model, the dependent variable has three or more possible outcomes; however, these values have no specified order. For example, movie studios want to predict what genre of film a moviegoer is likely to see to market films more effectively.

c) **Ordinal logistic regression:** This type of logistic regression model is leveraged when the response variable has three or more possible outcome, but in this case, these values do have a defined order. Examples of ordinal responses include grading scales from A to F or rating scales from 1 to 5.

4.2. LINEAR DISCRIMINANT ANALYSIS

Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification technique commonly used in statistics and machine learning. It's primarily employed for multi-class

classification problems, unlike logistic regression which is typically used for binary classification. Here's a high-level overview of how LDA works:-

a) **Assumption:** LDA assumes that the features are normally distributed and that the classes have identical covariance matrices.

b) **Dimensionality Reduction:** LDA seeks to reduce the dimensionality of the feature space while preserving as much of the class discriminatory information as possible.

c) **Discriminant Functions:** LDA finds discriminant functions of the features that best separate the classes. The number of functions is equal to the number of classes minus one.

d) **Decision Rule:** LDA uses a decision rule to classify new data points. This decision rule involves calculating the discriminant function values for each class and assigning the data point to the class with the highest discriminant function value.

e) **Optimization:** LDA maximizes the between-class variance while minimizing the within-class variance. This is typically done by maximizing a criterion called Fisher's Linear Discriminant.

Mathematically, LDA involves computing the mean vectors and covariance matrices of the different classes, and then solving eigenvalue problems to find the discriminant axes that maximize the separation between classes while minimizing the variation within each class.

LDA is widely used in various fields, including pattern recognition, image processing, and bioinformatics. It's particularly useful when the classes are well-separated and when the assumption of normally distributed classes holds true.

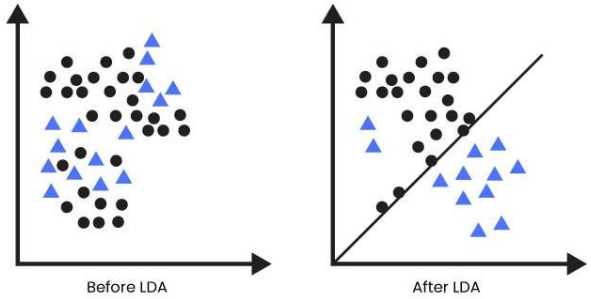

Figure 4: Classification using Linear Discriminant Analysis

4.3. NAÏVE BAYES CLASSIFIER

Naïve Bayes is a popular classification algorithm based on Bayes' Theorem with an assumption of independence between features. Despite its simplicity, Naive Bayes often performs well in practice and is particularly useful for text classification and other high-dimensional datasets.

Here's a breakdown of how Naïve Bayes works:-

a) Bayes' Theorem: At the core of Naive Bayes is Bayes' Theorem, which describes the probability of a hypothesis given the evidence. Equation 2 gives the mathematical equation for Bayes' Theorem.

$$P(A/B) = P(B/A) * \frac{P(A)}{P(B)} \quad (3)$$

Where:

- P(A/B) is the posterior probability of class A given the predictor B.
- P(B/A) is the likelihood of predictor B given class A.
- P(A) is the prior probability of class A.
- P(B) is the probability of predictor B.

b) Independence Assumption: Naïve Bayes assumes that the features are conditionally independent given the class. This means that the presence of one feature does not affect the presence of another feature. Although this assumption is often violated in real-world data, Naive Bayes can still perform well, especially when the features are approximately independent.

c) Classification Rule: Naive Bayes calculates the posterior probability of each class given the input features and then selects the class with the highest probability as the predicted class. This is expressed in equation 4 as:

$$\gamma = argmax_y \, P(y)\tau_{i=1}^{n} P(x_i|y) \quad (4)$$

Where:

- γ is the predicted class.
- n is the number of features.
- x_i is the ith feature.
- P(y)is the prior probability of class y.
- $P(x_i|y)$ is the likelihood of feature x_i given class y.

d) Different Variants: There are different variants of Naive Bayes, such as Gaussian Naive Bayes (when features are continuous and assumed to have a Gaussian distribution), Multinomial Naive Bayes (commonly used for text classification with discrete features), and Bernoulli Naive Bayes (similar to Multinomial Naive Bayes but for binary features).

Despite its simplicity and the "naive" assumption, Naive Bayes often performs surprisingly well in practice, especially for text classification tasks like spam detection, sentiment analysis, and document categorization. It's computationally efficient and can handle large datasets with high dimensionality. However, it may not perform well when the independence assumption is severely violated or when there is insufficient training data.

4.4. K- NEAREST NEIGHBOUR CLASSIFIER

The k-Nearest Neighbor (kNN) Algorithm is a widely used and intuitive machine learning algorithm that belongs to the category of instance-based learning or lazy learning. It is a non-parametric method used for both classification and regression tasks. The kNN algorithm is based on the principle of similarity, where the classification or prediction of a new instance is determined by its proximity to labeled instances in the training dataset.

The kNN algorithm follows the following general working principles:-

a) Training Phase: During the training phase, the algorithm stores the labeled instances of the training dataset. It typically involves loading the dataset into memory and organizing it in a suitable data structure, such as a KD-tree or a ball tree, to facilitate efficient nearest neighbor searches.

b) Prediction Phase: When a new, unseen instance is presented for classification or prediction, the k-NN algorithm performs the following steps-

- Distance Calculation- The algorithm calculates the distance between the new instance and all instances in the training dataset using the chosen distance metric. This step quantifies the similarity between instances based on their feature values. Equation 5 gives formula for distance calculation in kNN:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (5)$$

- Nearest Neighbor Selection- The algorithm selects the k instances with the smallest distances to the new instance. These k instances are considered the nearest neighbors.

- Class Label Assignment (Classification)- In classification tasks, the algorithm determines the class label for the new instance based on the majority class among the k nearest neighbors. It assigns the predicted class label to the new instance.

- Value Prediction (Regression)- In regression tasks, the algorithm calculates the average or weighted average of the target values of the k nearest neighbors. This value is assigned as the predicted value for the new instance.

The algorithm is a versatile and straight forward ML algorithm used for classification and regression tasks. Its reliance on instance-based learning and similarity-based principles makes it intuitive and easy to understand. By considering the k nearest neighbors and employing a voting scheme or averaging mechanism, the k-NN algorithm can provide accurate predictions for unseen instances.

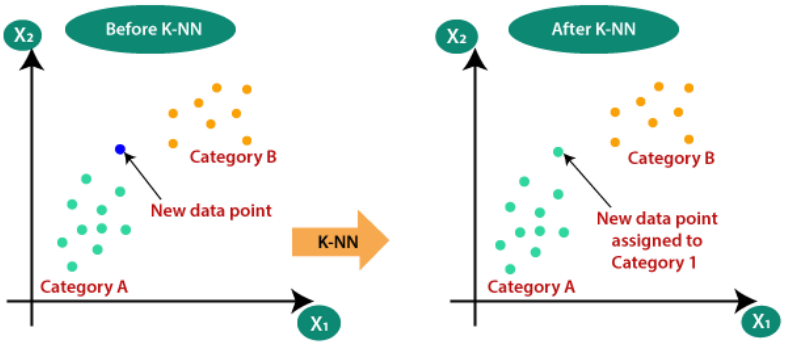

Figure 5: Classification using k- Nearest Neighbour Algorithm

4.5. DECISION TREES

Decision Tree is a supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a Decision Tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. A decision tree simply asks a question, and based on the answer (Yes/No), then it further split the tree into subtrees.

The most notable types of Decision Tree algorithms are:-

a) **IDichotomiser 3 (ID3):** This algorithm uses Information Gain to decide which attribute is to be used to classify the current subset of the data. For cach level of the tree. information gain is calculated for the remaining data recursively.

b) C4.5: This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifying attribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.

c) Classification and Regression Tree (CART): It is a dynamic learning algorithm which can produce a regression tree as well as a classification tree depending upon the dependent variable.

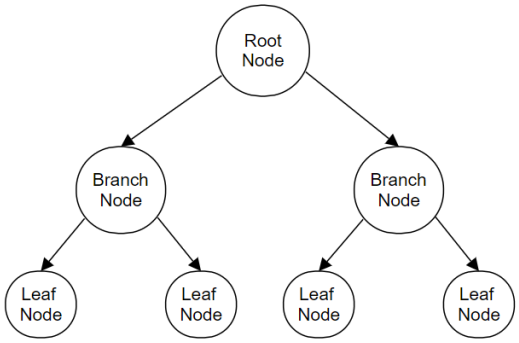

Figure 6: Decision Tree Structure

The algorithm starts from the root node of the tree. It compares the values of the root attribute with the record attribute and, based on the comparison, follows the branch and jumps to the next node. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood as:-

Step 1- Begin the tree with the root node, which contains the complete dataset.

Step 2- Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step 3- Divide the node into subsets that contains possible values for the best attributes.

Step 4- Generate the Decision Tree node, which contains the best attribute.

Step 5- Recursively make new decision trees using the subsets of the dataset created in above step.

Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

4.6. SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane's chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems.

The followings are important points in SVM:-

- **Support Vectors:** Data Points that are closest to the hyperplane are called support vectors. Separating line will be defined with the help of these data points.
- **Hyperplane:** As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
- **Margin:** It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

There are two important types of SVM:-

a) **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

b) **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

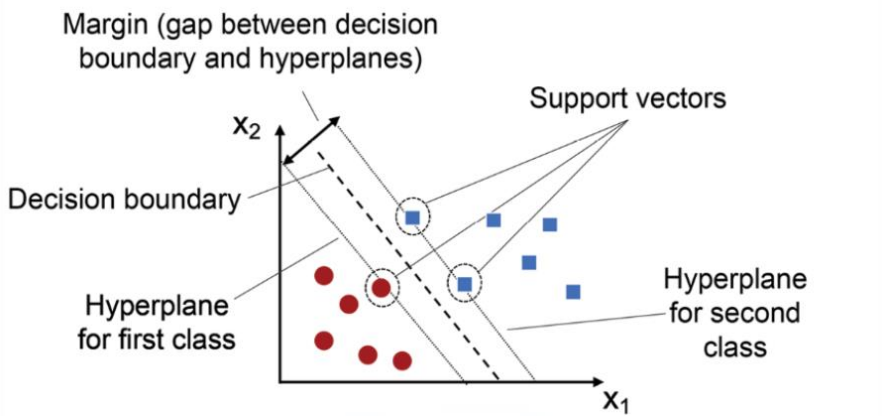

Figure 7: Classification using Support Vector Machines

4.7. RANDOM FOREST CLASSIFIER

Random Forest is a supervised machine learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vector which is independently sampled. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. Random Forests create decision trees on randomly selected data samples, get predictions from each tree and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random Forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from cach tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. It works in four steps:-

Step 1- Select random samples from a given dataset.

Step 2- Construct a Decision Tree for each sample and get a prediction result from each Decision Tree.

Step 3- Perform a vote for each predicted result.

Step 4- Select the prediction result with the most votes as the final prediction.

4.8. GRADIENT BOOSTING CLASSIFIER

Gradient Boosting Classifier (GBC) is a popular machine learning technique used for both classification and regression tasks. It belongs to the ensemble learning methods, which means it combines the predictions of multiple base estimators (typically decision trees) to improve the accuracy of the model.

Here's a brief overview of how Gradient Boosting Classifier works:-

a) **Base Learners:** Initially, it starts with an initial weak learner, often a decision tree with a very limited depth, known as a shallow tree or a stump.

b) **Sequential Training:** Unlike random forests, which train multiple trees independently, gradient boosting trains trees sequentially. Each new tree aims to correct errors made by the previous ones.

c) **Gradient Descent:** In each iteration, the algorithm fits a new weak learner to the residuals (the differences between the actual values and the predictions of the previous model). This is done by minimizing a loss function, such as the mean squared error for regression or the log loss for classification, using gradient descent.

d) **Gradient Calculation:** The gradient is calculated by finding the negative gradient of the loss function with respect to the model's prediction. This indicates the direction and magnitude of the change needed to minimize the loss.

e) **Weighted Combination:** Once the new tree is trained, its predictions are combined with the predictions of the previous trees, with each tree's contribution weighted according to its performance.

f) **Regularization:** To prevent overfitting, gradient boosting typically includes regularization parameters like learning rate, which controls the contribution of each tree to the final prediction, and tree-specific parameters like maximum depth, minimum samples per leaf, etc.

g) **Stopping Criteria**: The process continues for a specified number of iterations or until a certain level of performance is achieved.

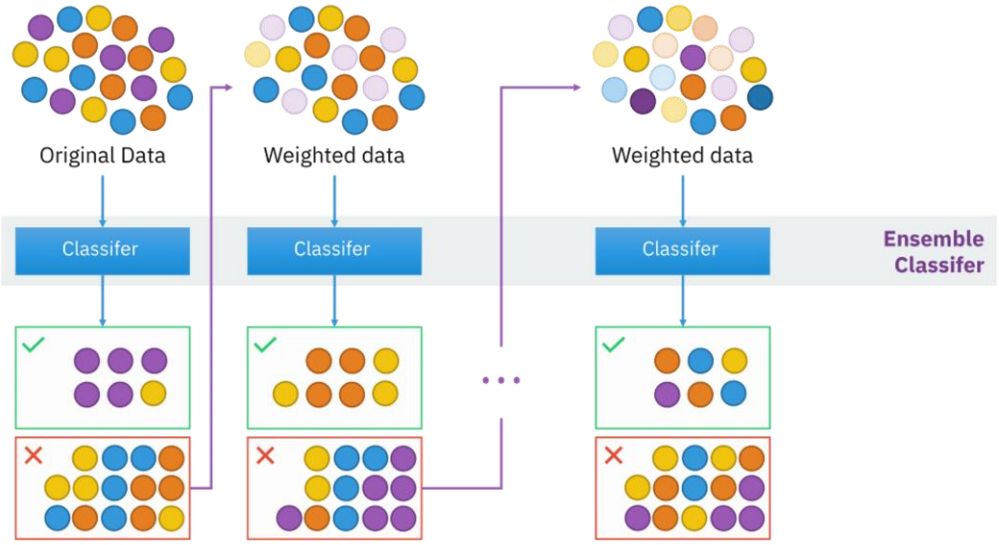

Figure 8: Classification using Gradient Boosting Classifier

Gradient Boosting Classifier tends to yield very accurate models, often outperforming other algorithms in many situations. However, it can be computationally expensive and prone to overfitting if not carefully tuned. Regularization techniques and cross-validation are often used to address these issues. Popular implementations include scikit-learn's GradientBoostingClassifier and XGBoost.

4.9. RIDGE CLASSIFIER

A Ridge Classifier is a linear classifier that uses Ridge Regression for classification tasks. It's a variant of logistic regression that incorporates L2 regularization, also known as ridge regularization, to prevent overfitting and improve the generalization of the model.

Here's how it works:

a) **Linear Model**: Like logistic regression, a ridge classifier is based on a linear model. It assumes that the decision boundary between classes is a linear function of the input features.

b) **Ridge Regularization:** In addition to minimizing the logistic loss function, which measures the difference between predicted probabilities and actual labels, ridge regularization adds a penalty term to the optimization objective. This penalty term is proportional to the squared magnitude of the coefficients (weights) of the model.

c) **Regularization Strength:** The strength of the regularization is controlled by a hyperparameter called the regularization parameter (often denoted as alpha or lambda). A higher value of alpha results in stronger regularization, which penalizes large coefficients more heavily, making the model simpler and less prone to overfitting.

d) **Training:** The model is trained using optimization algorithms like gradient descent or its variants. The goal is to find the optimal coefficients that minimize the combined loss function and regularization term.

e) **Decision Boundary:** Once trained, the ridge classifier computes a weighted sum of the input features to make predictions. If the result is above a certain threshold, the classifier assigns the sample to one class; otherwise, it assigns it to the other class.

Ridge classifiers are particularly useful when dealing with high-dimensional datasets where the number of features is close to or greater than the number of samples. The regularization helps prevent overfitting in such scenarios by shrinking the coefficients towards zero. However, it's essential to tune the regularization parameter appropriately to achieve the right balance between bias and variance.

In practice, ridge classifiers are implemented in various machine learning libraries, including scikit-learn in Python, where you can find the RidgeClassifier class for binary classification tasks and RidgeClassifierCV for cross-validated hyperparameter tuning.

4.10. ADABOOST CLASSIFIER

AdaBoost (Adaptive Boosting) Classifier is an ensemble learning method that combines multiple weak learners to create a strong classifier. It's particularly effective for binary classification problems but can be extended to multi-class classification as well.

Here's how AdaBoost works:

a) **Base Learners:** AdaBoost starts by training a base classifier (often a decision tree) on the original dataset. This base classifier is typically a weak learner, meaning it performs slightly better than random guessing but isn't very accurate on its own.

b) **Weighted Samples:** Initially, all samples in the dataset are given equal weights. During each iteration, the algorithm adjusts the weights of incorrectly classified samples, giving higher weights to those that were misclassified by the previous classifiers. This allows subsequent classifiers to focus more on the difficult-to-classify samples.

c) **Sequential Training:** AdaBoost trains multiple weak classifiers sequentially. In each iteration, it focuses on the training instances that are misclassified by the previous classifiers. The weight of each classifier's vote is determined based on its accuracy in classifying the training data.

d) **Weighted Voting:** The final prediction of the AdaBoost classifier is obtained by combining the individual predictions of all weak classifiers, with each classifier's contribution weighted by its accuracy. Typically, classifiers with higher accuracy are given more weight in the final decision.

e) **Stopping Criteria:** The process continues for a predefined number of iterations or until a perfect classifier is achieved. AdaBoost can be sensitive to noise and outliers, so it's common to limit the number of iterations or impose other stopping criteria to prevent overfitting.

AdaBoost tends to perform well in practice and is relatively resistant to overfitting. However, it's essential to choose appropriate weak learners and tune hyperparameters such as the number of iterations and the learning rate to achieve optimal performance.

In Python, AdaBoost classifiers are implemented in libraries like scikit-learn, where you can find the AdaBoostClassifier class. This class allows you to specify the base estimator, the number of estimators (iterations), and other hyperparameters. Additionally, there are variations of AdaBoost, such as SAMME and SAMME.R, which differ in how they calculate the classifier weights during training.

CHAPTER 5

RESAMPLING METHODS

Resampling Method is a statical method that is used to generate new data points in the dataset by randomly picking data points from the existing dataset. It helps in creating new synthetic datasets for training machine learning models and to estimate the properties of a dataset when the dataset is unknown, difficult to estimate, or when the sample size of the dataset is small.

5.1. OVER SAMPING

Oversampling involves increasing the number of instances in the minority class to balance the class distribution. This process is considered when the amount of data is inadequate. Oversampling can be beneficial for improving the performance of machine learning models, especially when the class imbalance is severe. However, it's essential to be cautious as oversampling can also introduce bias and overfitting, especially if not applied judiciously. This can be done in several ways. Some of them are:

5.1.1. SMOTE

In order to address the issue of class imbalance, the popular machine learning approach known as SMOTE generates synthetic samples of the minority class. It interpolates additional instances between samples of the minority class that already exist in order to increase the representation of the minority class in the training data. This results in the minority class being represented more accurately.

5.1.2. ADASYN

This is an addition to the SMOTE algorithm created especially to deal with the unbalanced datasets. By concentrating on the samples that are more difficult to accurately identify, it intelligently creates synthetic samples for the minority class, enhancing the classifier's overall performance. It is a powerful algorithm for handling class imbalance in machine learning datasets. It's particularly useful when the distribution of the minority class is not uniform and varies across different regions of the feature space.

5.2. UNDER SAMPING

Undersampling involves reducing the number of instances in the majority class to balance the class distribution. This process is considered when the amount of data is adequate. Undersampling can be beneficial for improving the performance of machine learning models, especially when the class imbalance is severe. However, it's essential to be cautious as undersampling can also discard potentially useful information and reduce the representativeness of the dataset. This can be done in several ways. Some of them are:

5.2.1. ALLkNN

This is a classification based strategy which finds and duplicates each minority class instance's k closest neighbors. It seeks to improve the representation of the minority class by duplicating these nearby examples, which may help classifiers perform better on unbalanced datasets. The ALLKNN algorithm addresses this issue by considering all instances in the dataset as potential neighbors during the classification process, rather than limiting the search to the k nearest neighbors. By doing so, ALLKNN aims to improve the classification accuracy for minority class instances. To get the best results, parameter tuning must be done carefully as close duplicates of the nearest neighbors may lead to overfitting.

5.2.2. TomekLinks

The TomekLinks technique finds and eliminates the pairs of samples from several classes that are most similar to one another. It seeks to strengthen the divide between the majority and minority classes by getting rid of these occurrences, which could improve algorithmic classification performance. It focuses on the borderline instances, which are those instances that are close to the decision boundary between classes. The idea is to remove instances that form Tomek Links, which are pairs of instances from different classes that are nearest neighbors to each other. It might not work well in situations when the classes overlap much, though, and it might result in the loss of some potentially helpful samples.

CHAPTER 6

PERFORMANCE EVALUATION METRICES

Performance evaluation metrics in machine learning are used to assess the quality and effectiveness of a model's predictions. They help in understanding how well a model generalizes to new, unseen data. Here are some commonly used evaluation metrics:

6.1. ACCURACY

Accuracy is a fundamental evaluation metric in machine learning that measures the overall correctness of the model's predictions. It is calculated as the ratio of the number of correctly predicted instances to the total number of instances in the dataset as given in equation 6. The frequency with which a ML model predicts the result accurately is measured by its accuracy.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{6}$$

6.2. RECALL

Recall, also known as sensitivity or true positive rate, measures the ability of a classification model to identify all relevant instances from the dataset. In other words, it answers the question: “Of all the actual positive instances, how many did the model correctly predict as positive?”. The percentage of data samples that a ML model correctly classifies as belonging to a particular class i.e., the "positive class", out of the total sample. It is also called as True Positive Rate (TPR). Mathematical formula for calculating recall is given in equation 7 as:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{7}$$

6.3. PRECISION

Precision is a key evaluation metric in machine learning that measures the accuracy of positive predictions made by the model. It answers the question: “Of all the instances that the model predicted as positive, how many were actually positive?”. It is a measurement of how effective a machine learning model is. The ratio of true positives to total positive predictions, which is the sum of true positives and false positives, is what we mean when we talk about this particular metric. Mathematical formula for calculating precision is given in equation 8 as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{8}$$

6.4. F1 SCORE

The F1 score is a popular evaluation metric in machine learning that balances both precision and recall. It is the harmonic mean of precision and recall, providing a single score that considers both false positives and false negatives. The accuracy of a model can be quantifies using the F1 score as given in equation 9. The accuracy metric is a system that determines the number of times a model has correctly predicted across the entirety of the dataset under consideration. It is the measure of predictive performance.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

6.5. KAPPA

Kappa is a statistical measure that assesses the agreement between two raters (or in the context of machine learning, two sets of labels) while taking into account the possibility of the agreement occurring by chance. A statistic that is utilized for the purpose of evaluating the efficiency of ML classification models is known as the Kappa Coefficient. The traditional 2x2 confusion matrix serves as the foundation for its formula, as given in equation 10, which is utilized in the field of statistics and ML for the purpose of evaluating binary classifiers.

$$k = \frac{p_0 - p_e}{1 - p_e} \tag{10}$$

6.6. MCC

MCC stands for Matthews Correlation Coefficient. It's a single metric that summarizes the confusion matrix of a binary classification problem. MCC takes into account true and false positives and negatives and is especially useful when dealing with imbalanced datasets.

The Matthews correlation coefficient is calculated using the formula given in equation 11:

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{11}$$

Where:

- TP is the number of True Positives.
- TN is the number of True Negatives.
- FP is the number of False Positives.
- FN is the number of False Negatives.

CHAPTER 7

PROPOSED METHODOLOGY

7.1. ALGORITHM

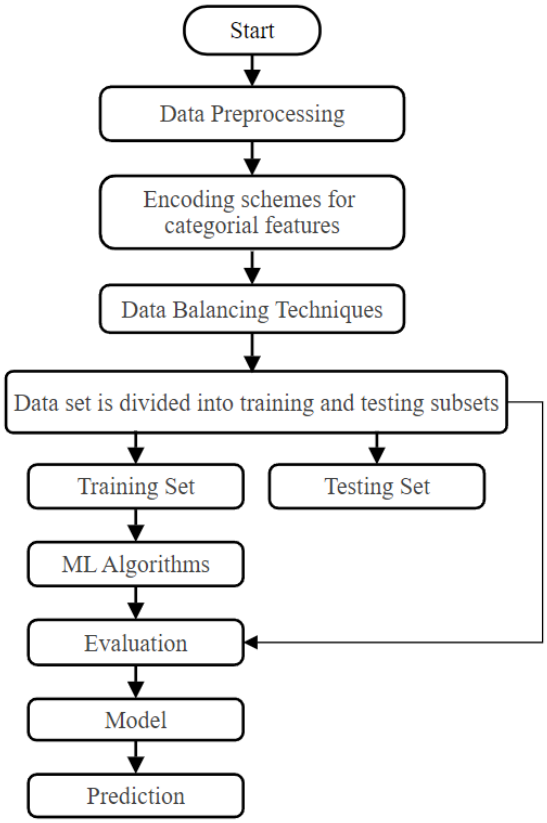

Figure 9: Flowchart of model

Step I: Data Preprocessing and Encoding-

The Twitter dataset have been preprocessed to improve the quality of the data. It improves the accuracy and reliability of a dataset by removing missing or inconsistent data values resulting from human or computer error. This makes the data more suitable for prediction process. The dataset is then encoded for further analysis. Data encoding is the process of changing raw data

into a format that can be read and interpreted by an algorithm. Here, we have used label encoding. Label Encoding is the technique to convert categorical columns into numerical ones so that they can be fitted by machine learning models which only take numerical data.

Step II: Generate models for an imbalanced dataset-

Initally, all the models are generated for the raw imbalanced dataset based on training data. An imbalanced dataset is a classification dataset with an uneven distribution of class proportions, where the majority class has the highest number of samples, and the minority class has the lowest. Then, performance of all models are evaluated using various performance evaluation metrices like accuracy, precision, F1 score, etc.

Step III: Select the top models for the evaluation of the results-

The top-performing models undergo further analysis to determine the best among them. This process involves thorough examination and evaluation to identify the model that demonstrates superior performance. The most effective model is selected for implementation and further refinement.

Step VI: Perform the Resampling techniques to balance the dataset-

Data resampling is a statistical technique that involves repeatedly drawing samples from a dataset to gather more information about it. Resampling can help identify bias or issues in the data, improve accuracy, and estimate uncertainty. Various resampling techniques are performed to reduce imbalance in the dataset. Different techniques used are: SMOTE, ADASYN, ALLkNN, TomekLinks.

Step V: Optimization of the model by applying hyperparameters-

Hyperparameter optimization is the process of finding the best combination of hyperparameters for a machine learning model to maximize its performance. Hyperparameters are configuration variables that control a model's learning process and general behavior, such as its architecture, regularization strengths, and learning rates. It can have a significant impact on a model's accuracy, efficiency and generalization.

Step VI: Prediction based on test data-

The final model generated is applied to predict outcomes on testing data, offering valuable insights into its performance and potential outcomes. Utilizing the testing dataset allows for thorough assessment of the model's efficacy in real-world scenarios, shedding light on its predictive capabilities and generalization ability. By analyzing the model's predictions against the test data, researchers and practitioners gain a deeper understanding of its strengths, weaknesses, and overall reliability. These insights inform decision-making processes and refine the model for optimal performance in practical applications.

CHAPTER 8

RESULTS AND DISCUSSIONS

Table 4 compares the various ML models on the processed unbalanced datasets various performance metrics including precision, recall, F1 score and recall for 14 different classifiers.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
gbc	Gradient Boosting Classifier	0.9362	0.795	0.0988	0.9289	0.1776	0.1664	0.2889
knn	K Neighbors Classifier	0.9342	0.7526	0.2587	0.5665	0.3547	0.3251	0.3533
lightgbm	Light Gradient Boosting Machine	0.9314	0.7951	0.0516	0.634	0.0944	0.085	0.1642
xgboost	Extreme Gradient Boosting	0.9309	0.7987	0.0478	0.6295	0.0879	0.0784	0.1562
lr	Logistic Regression	0.9299	0.5184	0	0	0	0	0
ridge	Ridge Classifier	0.9299	0	0	0	0	0	0
qda	Quadratic Discriminant Analysis	0.9299	0.6163	0	0	0	0	0
ada	Ada Boost Classifier	0.9299	0.762	0	0	0	0	0
lda	Linear Discriminant Analysis	0.9299	0.5184	0	0	0	0	0
dummy	Dummy Classifier	0.9299	0.5	0	0	0	0	0
svm	SVM - Linear Kernel	0.9293	0	0	0	0	-0.0011	-0.0054
et	Extra Trees Classifier	0.9143	0.7032	0.3607	0.383	0.3713	0.3254	0.3256
rf	Random Forest Classifier	0.9136	0.7567	0.3684	0.3799	0.3737	0.3273	0.3275
dt	Decision Tree Classifier	0.9134	0.662	0.3697	0.3795	0.3742	0.3277	0.3279

Table 4: Performance analysis of all the models for imbalanced data

Table 5 represents the resultant values of the sentiment analysis for various sampling strategies. SMOTE and ADASYN are used as the oversampling techniques and ALLKNN and TomekLinks are used as the under-sampling techniques.

Gradient Boosting Classifier		
	Sampling Technique	**Accuracy**
Over-Sampling	SMOTE	0.7735
	ADASYN	0.7193
Under-Sampling	ALLKNN	0.9338
	TomekLinks	0.9366

Table 5: Performance analysis of Gradient Boosting Classifier without using hyper-parameter

Table 6 shows the resultant values of the sentimental analysis of the given dataset after balancing and tuning the dataset and using the different sampling techniques. The sampling techniques used for over-sampling dataset are the SMOTE and ADASYN and the sampling technique used for fetching the resultant values for the under-sampling dataset are ALLKNN and Tomek Links.

Gradient Boosting Classifier		
	Sampling Technique	**Accuracy**
Over-Sampling	SMOTE	0.796
	ADASYN	0.7894
Under-Sampling	ALLKNN	0.9343
	TomekLinks	0.9368

Table 6: Performance analysis of Gradient Boosting Classifier with using hyper-parameter

Table 7 briefs us about the confusion matrix for the imbalanced dataset. In relation to the numbers of actual no and actual yes. Specifically, it displays the number of erroneous predictions as well as the number of true predictions. Both of their values are as:

True- Positives = 51

True- Negatives = 622

False- Positives = 7

False- Negatives = 8909

	Predicted No	**Predicted Yes**
Actual No	8909	7
Actual Yes	622	51

Table 7: Confusion Matrix

CHAPTER 9

CONCLUSION

9.1. CONCLUSION

Improving the accuracy of the model that has been suggested is the primary goal of the research that has been proposed. Consequently, in order to accomplish this goal, the presence of a balanced dataset is one of the primary prerequisites. This is because an imbalanced dataset can result in bias towards classes that have a greater number of samples. Every classifier has its unique set of benefits, which are determined by the kind of data that is used for training. It is seen that the results of the Gradient boosting classifier were superior to those of the other classifiers. The purpose of this endeavour is to employ a variety of optimization strategies aimed at selecting the most advantageous features, in conjunction with resampling techniques, in order to ensure that classifiers produce satisfactory outcomes. The first step is to select the five most accurate classifiers and nominate them for further consideration. Each model is assessed according to its level of accuracy. In conclusion, the Gradient boosting classifier beats the other classifiers in terms of accuracy, surpassing prior efforts that were performed with the same dataset by a margin of 93.62%. The work that will be done in the future can involve other datasets to work on.

9.2. FUTURE SCOPE

The future scope of sentiment analysis using machine ML is broad and promising, driven by advancements in technology and the increasing importance of understanding human emotions in digital interactions.

In the next few years, sentiment analysis of text data will likely advance in both depth and breadth. Techniques will become more nuanced, incorporating contextual understanding and cultural sensitivities. Integration with AI-driven systems will enable real-time analysis at scale, facilitating personalized experiences in customer service, market research, and political analysis. Additionally, the convergence of sentiment analysis with other AI disciplines like natural language understanding and emotion recognition will unlock richer insights into human behavior and preferences.

REFERENCES

1. Agarwal, Apoorv, et al. "Sentiment analysis of twitter data." Proceedings of the workshop on language in social media (LSM 2011). 2011.

2. Wang, Yili, et al. "Sentiment analysis of Twitter data." Applied Sciences 12.22 (2022): 11775.

3. Chalothom, Tawunrat, and Jeremy Ellman. "Simple approaches of sentiment analysis via ensemble learning." information science and applications. Springer Berlin Heidelberg, 2015.

4. Xia, Rui, Chengqing Zong, and Shoushan Li. "Ensemble of feature sets and classification algorithms for sentiment classification." Information sciences 181.6 (2011): 1138-1152.

5. Saif, Hassan, Yulan He, and Harith Alani. "Semantic sentiment analysis of twitter." The Semantic Web–ISWC 2012: 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I 11. Springer Berlin Heidelberg, 2012.

6. Gautam, Geetika, and Divakar Yadav. "Sentiment analysis of twitter data using machine learning approaches and semantic analysis." 2014 Seventh international conference on contemporary computing (IC3). IEEE, 2014.

7. Neelakandan, S., and D. Paulraj. "A gradient boosted decision treebased sentiment classification of twitter data." International Journal of Wavelets, Multiresolution and Information Processing 18.04 (2020): 2050027.

8. Le, Bac, and Huy Nguyen. "Twitter sentiment analysis using machine learning techniques." Advanced Computational Methods for Knowledge Engineering: Proceedings of 3rd International Conference on Computer Science, Applied Mathematics and ApplicationsICCSAMA 2015. Springer International Publishing, 2015.

9. Sahayak, Varsha, Vijaya Shete, and Apashabi Pathan. "Sentiment analysis on twitter data." International Journal of Innovative Research in Advanced Engineering (IJIRAE) 2.1 (2015): 178-183.

10. Neethu, M. S., and R. Rajasree. "Sentiment analysis in twitter using machine learning techniques." 2013 fourth international conference on computing, communications and networking technologies (ICCCNT). IEEE, 2013.

11. Sankar, H., and V. Subramaniyaswamy. "Investigating sentiment analysis using machine learning approach." 2017 International conference on intelligent sustainable systems (ICISS). IEEE, 2017.

12. Alsaeedi, Abdullah, and Mohammad Zubair Khan. "A study on sentiment analysis techniques of Twitter data." International Journal of Advanced Computer Science and Applications 10.2 (2019): 361-374.

13. Kharde, Vishal, and Prof Sonawane. "Sentiment analysis of twitter data: a survey of techniques." arXiv preprint arXiv:1601.06971 (2016).

14. Anjaria, Malhar, and Ram Mohana Reddy Guddeti. "Influence factor based opinion mining of Twitter data using supervised learning." 2014 sixth international conference on communication systems and networks (COMSNETS). IEEE, 2014.

15. Jain, Anuja P., and Padma Dandannavar. "Application of machine learning techniques to sentiment analysis." 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). IEEE, 2016.

16. Sahayak, Varsha, Vijaya Shete, and Apashabi Pathan. "Sentiment analysis on twitter data." International Journal of Innovative Research in Advanced Engineering (IJIRAE) 2.1 (2015): 178-183.

17. Da Silva, Nadia FF, Eduardo R. Hruschka, and Estevam R. Hruschka Jr. "Tweet sentiment analysis with classifier ensembles." Decision support systems 66 (2014): 170-179.

18. Jagdale, Rajkumar S., Vishal S. Shirsat, and Sachin N. Deshmukh. "Sentiment analysis on product reviews using machine learning techniques." Cognitive Informatics and Soft Computing: Proceeding of CISC 2017. Springer Singapore, 2019.

19. Singh, Jaspreet, Gurvinder Singh, and Rajinder Singh. "Optimization of sentiment analysis using machine learning classifiers." Humancentric Computing and information Sciences 7 (2017): 1-12.

20. Ain, Qurat Tul, et al. "Sentiment analysis using deep learning techniques: a review." International Journal of Advanced Computer Science and Applications 8.6 (2017).

21. Rathi, Megha, et al. "Sentiment analysis of tweets using machine learning approach." 2018 Eleventh international conference on contemporary computing (IC3). IEEE, 2018.

22. Chugh, Bharti, and Nitin Malik. "Machine Learning Classifiers for Detecting Credit Card Fraudulent Transactions." Information and Communication Technology for Competitive Strategies (ICTCS 2021) ICT: Applications and Social Interfaces. Singapore: Springer Nature Singapore, 2022. 223-231.

23. Qi, Yuxing, and Zahratu Shabrina. "Sentiment analysis using Twitter data: a comparative application of lexicon-and machine-learning-based approach." Social Network Analysis and Mining 13.1 (2023): 31.

24. Pasrija, Paridhi, Utkarsh Singh, and Mehak Khurana. "Performance Analysis of Intrusion Detection System Using ML Techniques." Applying Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection (2024): 135-150.

APPENDIX 1

Exploring ML Methods for Sentiment Analysis in Text Data

Vishu Agarwal
Department of Computer Science and Engineering
KIET Group of Institutions
Ghaziabad, India
vishuagarwal183@gmail.com

Ashmit Tayal
Department of Computer Science and Engineering
KIET Group of Institutions
Ghaziabad, India
ashmittayal196@gmail.com

Rajani Dixit
Department of Computer Science and Engineering
KIET Group of Institutions
Ghaziabad, India
rajanidixit024@gmail.com

Bharti Chugh
Department of Computer Science and Engineering
KIET Group of Institutions
Ghaziabad, India
bharti.kathpalia@gmail.com

Shikha Jain
School of Engineering and Technology
Vivekananda Institute of Professional Studies- Technical Campus
New Delhi, India
ssjainshikha@gmail.com

***Abstract*— Data Analysis has become an important part of our life, as it helps us to draw useful information, decision-making conclusions on any particular raw data. One component of digital data analysis is sentiment analysis. Sentiment analysis evaluates the emotional tone of textual data and classifies it as neutral, negative, or positive. In business, this analysis is frequently utilized for market social media monitoring, market research, consumer feedback and many more. In order to automate the procedure, machine learning models that recognize patterns in labeled data are frequently used. The objective is to derive meaningful insights from textual data so that decisions can be made with knowledge, effectively. In order to provide important insights into public opinion and behavior, this study uses machine learning techniques to investigate the sentiments conveyed in Twitter data. A wide range of Twitter posts, including ones about trending hashtags, events, and subjects, are gathered as part of the process.**

Sentiment analysis is done using machine learning algorithms. Sentiment analysis and examples of human language are used to train machine learning models. Following this training, machine learning is used by sentiment analysis software to evaluate and rank human language according to its previous training.

The current study proposes a technique that best analyzes the sentiments of the text used in social media is gradient boosting classifier. With this method, an additive model can be built in a step-by-step, forward fashion and any differentiable loss function can be optimized. There are n-class regression trees that are fitted on the negative gradient of the loss function, which could be the multiclass log loss or the binary loss. This process is repeated in each step. In the particular case of binary classification, just one regression tree is induced.

The computational results demonstrates that gradient boosting approach with an accuracy of 93.62% which outperforms is preferable to other classifiers.

Keywords—Sentiment analysis, Twitter, Dataset, Machine learning, Gradient boosting classifier

I. INTRODUCTION

An intriguing area of machine learning, sentiment analysis extracts and understands textual emotions. Sentiment analysis analyzes human language using machine learning. Sentiment analysis uses contextual meaning to estimate a brand's social sentiment and if its product will sell in the market. The dataset is used to train a variety of ML models, including Naïve bayes, K-nearest neighbour classifier, Support vector machines, etc to find patterns and correlations between words and sentiments. The performance level is analysed with the metrics in mind like accuracy, F1 score, precision. The study tackles issues including managing sarcasm, slang, and context-specific terms in an effort to better grasp public emotion on Twitter. Market research, brand management, political analysis, and other fields can all benefit from the sentiment analysis's results. They offer businesses and scholars insightful information gleaned from the enormous volume of viewpoints expressed on the Twitter network.

Potential uses of this research include real-time public sentiment monitoring, which enables proactive response and decision-making tactics across a range of industries. Reference [1] examines a well-known microblog called Twitter and develops models to categorize "tweets" into sentiment categories that are good, negative, and neutral. Sentiment Analysis is needed to store data efficiently and cheaply. You can solve all real-time scenarios with sentiment analysis. Businesses use sentiment analysis to analyze customer opinions, build brand reputation, create better goods, and personalize content. It supports market research, policymaking, and automation. Text sentiment is measured using metrics to determine positive, negative, or neutral. Today, the internet offers several means to convey feelings. Machines are trained with text examples of emotions to identify sentiment without human input. To put it briefly, computers can learn new tasks using machine learning without the need for explicit programming. Sentiment analysis models may learn context, misapplied words and sarcasm beyond definitions. As many as methods and complicated system of equation commands and train machines to carry out sentiment analysis. However, combined they can produce great effects.

Natural language and sentiment are employed in machine learning model training. Sentiment analysis software uses machine learning to score human language after this training. Among the algorithms that are utilized the most are Linear Regression, Support vector machines (SVM), Naïve Bayes, and many more. Every new model proceeds in the direction of least prediction error within the space of possible predictions for every training example. Reference [2] states that opinion mining and sentiment analysis are two exciting new areas of research that are used to find out what people think and feel about particular subjects. As a result, opinion mining and

sentiment analysis are frequently used synonymously to convey the same concept. Reference [3] illustrates that the content of tweets has emerged as a hot research topic concerning the positive or negative polarity of sentiment. Our work with sentiment analysis of tweet contexts demonstrates that group learning, which is made up of the majority vote of the Stacking, support vector machine, SentiStrength, and naïve bayes, can enhance and be more successful in achieving accuracy performance. Reference [4] tries to illustrate why popular text classification techniques like maximum entropy, support vector machines and naïve bayes are necessary. Reference [5] states the characteristic short length and asymmetrical structure of microblogs like Twitter present various additional obstacles for sentiment analysis.

The research literature on sentiment analysis on microblogs identifies two primary study avenues. Reference [6] encompasses the examination of the content on the Internet spanning numerous domains that are seeing a rapid increase in both quantity and number as websites are devoted to particular product categories and have expertise gathering customer reviews from other websites, including Amazon, twitter, etc. Reference [7] states that the data sentiment analysis is a categorization method based on finding if a review is good, negative, or neutral based on the given opinion. Finding the review writer's position at the document, sentence, or aspect levels is SA's main goal.

These days, SA is frequently used to analyze reviews from a variety of different data domains, including reviews of products, movies, hotels, restaurants, and many more. Reference [8] Tweets on Twitter allow users to voice their thoughts on a variety of subjects. This can assist marketers target their campaigns to convey consumers' opinions about products and companies, bullying incidents, and events.

A. Key Contributions

The key contributions in this research are:

a) Creating an efficient sentiment analysis model for the twitter dataset, which can accurately identify the sentiments, moods, effects and biasness.

b) Assessing the effects of various resampling and optimization strategies on the overall performance of sentiment analysis model in order to determine the most effective classifier technique for enhancing its overall performance.

c) Examining and adjusting the model's hyperparameters for better results and accuracy.

II. Literature Review

According to [9] sentiment analysis has gained popularity as a field of study in computational linguistics due to the proliferation of sentiment data from blogs, online forums, and social media sites like Facebook and Twitter. Reference [10] states that the process of locating and categorizing viewpoints or feelings represented in the source text is known as sentiment analysis. By using sentiment analysis in a particular domain, the influence can be determined by information on sentiment classification. Reference [11] addresses several issues and obstacles in the field of sentiment analysis and presents an array of methodologies and concerns associated with the field.

Reference [12] presented that over the past 10 years, opinion study of Twitter data has received a lot of attention. This type of investigation involves examining the words that make up "tweets," or remarks. Because of this, the purpose of this study is to investigate the numerous sentiment analyses that were performed on Twitter data and the results of those analyses. Reference [13] provide a survey and an examination of the most recent developments in option mining techniques, including lexicon-based approaches and machine learning, is presented in this comparative analysis. Also provide a study on twitter data streams by employing a number of different machine learning algorithms, such as Naïve Bayes, Support Vector Machine and Max Entropy. Reference [7] proposes that using a gradient-boosted decision tree classifier, sentiment analysis and sentiment classification of Twitter data are conducted.

Reference [14] suggested a hybrid method of opinion extraction from Twitter data that combines characteristics that are both direct and indirect. According to this method supervised classifiers that consist of a network of artificial neural connections, Naïve Bayes, Maximum Entropy and Support Vector Machines (SVM).

Reference [15] states Sentiment analysis is the method of automatically determining if an entity (i.e., product, people, topic, event, etc.) is the topic of either neutral, negative, or positive opinion expression in a user-generated text. The purpose for conducting this research is to offer a thorough examination of machine learning technique for sentiment analysis utilizing data from Twitter. Reference [16] discusses the current data mining analysis of the Twitter dataset, including the sentiment analysis using machine learning algorithms. Reference [17] provides a method that employs lexicons and classifier ensembles to automatically categorize the tone expressed in tweets. Tweets on a query word are categorized as good or bad. This method has numerous applications, including helping businesses track public opinion about their brands and consumers who can use sentiment analysis to look for products. Reference [18] states that they used machine learning methods to categorize reviews as favourable or negative after preprocessing. The conclusion of the paper states that the most accurate classification results for product reviews are achieved through the application of machine learning techniques. Reference [19] presents that people's opinions about governments, events, goods and services on social media are expressed through words and phrases. Sentiment analysis in natural language processing is the technique of extracting positive and negative polarities from text that is shared on social media platforms. Reference [20] proposes that online user sentiments have a significant impact on lawmakers, goods suppliers, and readers. Sentiment analysis has garnered substantial attention as a means of analyzing and organizing the unstructured data gleaned from social media. Reference [21] focuses on the categorization of tweets' emotional content using information obtained from Twitter. In the past, researchers used machine learning techniques that were already available for sentiment analysis, but the findings indicated that these methods were not producing superior sentiment categorization results. Reference [22] states that in the foreseeable future, financial sector fraud is anticipated to have significant repercussions.

III. Preliminaries

A. Exploratory Data Analysis

In order to construct predictive models, exploratory data analysis, or EDA, is requisite to machine learning since it helps to comprehend the traits, connections, and patterns

found in the dataset. To acquire insights, spot problems with data quality, and guide preparation actions for the best possible model performance, it makes use of descriptive statistics, data visualization, and feature analysis.

Table 1 gives detailed description of the dataset, including target value i.e., "label" (binary value) and the number of data used.

TABLE I. THE DETAIULED DESCRIPTION OF THE DATASET USED

	Description	Value
0	Session ID	123
1	Target	Label
2	Target Type	Binary
3	Original Data Shape	(31962, 2)
4	Transformed Data Shape	(31962, 2)
5	Transformed Train Data Shape	(22373, 2)
6	Transformed Test Data Shape	(9589, 2)
7	Numeric Features	1
8	Preprocess	True
9	Imputation Type	Simple
10	Numeric Imputation	Mean
11	Categorial Imputation	Mode

B. Data Preprocessing

A tweet with a positive sentiment is represented by a value of 1, whereas a tweet with negative emotion is represented by a value of 0.

Table 2 describes each attribute of the dataset that is ID is used as the serial number which contains the numeric values, The label contains binary values (good or bad) and the tweets are the categorial values.

TABLE II. DESCRIPTION OF EACH ATTRIBUTR OF THE DATASET

Column Name	Attribute Description
ID (numerical)	Serial number
Label (binary)	Good or Bad
Tweet (categorial	Twitter Post

To ensure data quality, the dataset was carefully examined for missing, duplicate, and trash values before being subjected to additional analysis. In order to get the dataset ready for machine learning algorithms, the z-score method was utilized to standardize the data by taking the mean value (μ) and dividing it by the standard deviation (σ) of each feature as shown in (1). The label encoding method was then used to convert the categorial features into numerical representations.

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

The data is then centered around zero and its standard deviation is measured to be one, making appropriate for ML algorithms that require normalized data. This technique makes the data suitable for machines.

IV. MACHINE LEARNING MODELS

Reference [23] states machine learning models are able to construct classifiers with extraction of feature vectors, which allows them to complete sentiment categorization. The primary components of these methods are the collection and cleaning of data, the extraction of features, the formation of the classifier through training data, and the evaluation of the results. Through the application of machine learning strategies, the dataset is partitioned into a training dataset and a test dataset. A performance evaluation of the classifier is carried out using the test dataset, whereas the training sets are designed to provide assistance to the classifier in learning the text features.

A. Logistic Regression

It ascertains the likelihood that a specific instance will belong to a class for binary classification tasks. It's a classification algorithm, not regression. It uses the logistic function to model the independent factors and dependent variable log-odds as shown in (2). A threshold is used to make binary predictions from output probabilities. LR is popular for its simplicity, interpretability and versatility.

$$y = \beta_0 + \beta_1 x \quad (2)$$

B. Linear Discriminant Analysis

It reduces dimensionality and classifies in machine learning. The combinations that are linear in characteristics that most separate two or more classes while conserving class discriminating information are its goal. Multi-class classification issues benefit from linear discriminant analysis. The maximizing of between-class variance compared to within-class variance determines class labels and discriminant functions. Linear discriminant analysis assumes Gaussian data and the same covariance matrix for all classes.

C. Naïve Bayes Classifier

They are often classified using the probabilistic machine learning method Naïve Bayes. It is assumed that characteristics given the class designation, are conditionally independent based on the Bayes theorem, as given in (3), simplifying likelihood computation regarding text categorization and spam filtering, NB often executes well despite its "naive" premise. It uses little training data and is computationally efficient.

$$P(A/B) = P(B/A) * \frac{P(A)}{P(B)} \quad (3)$$

D. K- Nearest Neighbour Classifier

It is a basic and simple categorization machine learning model. It classifies new data points using the feature space class majority of their K closest neighbors. How many neighbors there are is determined by user-defined parameter K. The non-parametric, lazy KNN doesn't make assumptions about the data distribution or develop an explicit model during training, making it adaptable for varied datasets.

E. Decision Trees

It is a regression and classification technique for supervised machine learning. Recursively partitioning the dataset by the most important feature at each node yields a tree-like model. The idea is to partition data to reduce impurity and variation. DT are interpretable, visualizable, and can capture complex data relationships.

F. Support Vector Machines

It is a sophisticated and supervised ML method for classification and regression. Data points are divided into groups by identifying a hyperplane that optimizes their distance from one another. SVM performs admirably in high-dimensional areas and handles non-linear interactions with kernel functions. Data points closest to the border make up support vectors of the decision- are selected to find the

best decision boundary. SVM's resilience and capacity to handle complex datasets make them popular.

G. Random Forest Classifier

It is a popular ensemble learning algorithm for categorization. It generates many decision trees during training and delivers class mode for categorization. RF reduces overfitting by employing subsets of characteristics and training data for each tree. It is durable, manages high-dimensional data, and has less variance than individual decision trees, making it a popular and effective machine learning alternative.

H. Gradient Boosting Classifier

The technique is known as ensemble learning, and it involves the generation of a number of unsuccessful learners in a sequential manner, typically decision trees, in order to repair errors that were produced by earlier machine learning models. Using GBC optimization, it reduces errors to a minimum while simultaneously updating forecasts at each stage. Implementations such as XGBoost, LightGBM, and CatBoost are becoming increasingly popular. A powerful and accurate prediction model is produced as a result of the combination of these weak learners, which is the final model.

I. Ridge Classifier

It is a linear grouping technique that regularizes LR to reduce overfitting. It adds L2 regularization (ridge penalty) to the cost function during training. This penalty discourages large coefficients, making the model more stable and generalizable. Ridge Classifier is effective for multicollinearity in feature space, and its regularization parameter determines regularization strength.

J. AdaBoost (Adaptive Boosting) Classifier

It uses ensemble learning for classification. It takes many weak learners, usually decision trees, and turns them into a potent prediction model. AdaBoost weights each instance based on weak learners' performance. Iteratively highlighting misclassified cases improves accuracy. AdaBoost is adaptable, outperforms models, and overfits less.

V. Resampling Methods

A. SMOTE (Synthetic Minority Over-sampling Technique)

In order to address the issue of class imbalance, the popular machine learning approach known as SMOTE generates synthetic samples of the minority class. It interpolates additional instances between samples of the minority class that already exist in order to increase the representation of the minority class in the training data. This results in the minority class being represented more accurately.

B. ADASYN (Adaptive Synthetic Sampling)

This is an addition to the SMOTE algorithm created especially to deal with the unbalanced datasets. By concentrating on the samples that are more difficult to accurately identify, it intelligently creates synthetic samples for the minority class, enhancing the classifier's overall performance.

C. TomekLinks

The TomekLinks technique finds and eliminates the pairs of samples from several classes that are most similar to one another. It seeks to strengthen the divide between the majority and minority classes by getting rid of these occurrences, which could improve algorithmic classification performance. It might not work well in situations when the classes overlap much, though, and it might result in the loss of some potentially helpful samples.

D. ALLKNN (All k- Nearest Neighbors)

This is a classification-based under-sampling strategy which finds and duplicates each minority class instance's k closest neighbors. It seeks to improve the representation of the minority class by duplicating these nearby examples, which may help classifiers perform better on unbalanced datasets. To get the best results, parameter tuning must be done carefully as close duplicates of the nearest neighbors may lead to overfitting.

VI. Performance Evaluation Metrics

The performance of each individual model is based on Accuracy, F1 score, AUC and Precision.

A. Accuracy

The frequency with which a ML model predicts the result accurately is measured by its accuracy.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of predictions}$$

B. Recall

The percentage of data samples that a ML model correctly classifies as belonging to a particular class i.e., the "positive class", out of the total sample. It is also called as True Positive Rate (TPR).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

C. Precision

It is a measurement of how effective a machine learning model is. The ratio of true positives to total positive predictions, which is the sum of true positives and false positives, is what we mean when we talk about this particular metric.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

D. F1 Score

The accuracy of a model can be quantifies using the F1 score. The accuracy metric is a system that determines the number of times a model has correctly predicted across the entirety of the dataset under consideration. It is the measure of predictive performance.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

E. Kappa

A statistic that is utilized for the purpose of evaluating the efficiency of ML classification models is known as the Kappa Coefficient, which is also widely referred to as Cohen's Kappa Score. The traditional 2x2 confusion matrix serves as the foundation for its formula, which is utilized in the field of statistics and ML for the purpose of evaluating binary classifiers.

$$k = \frac{p_o - p_e}{1 - p_e}$$

F. MCC

An example of the þ coefficient that is a special case is the MCC. In the context of this discussion, a value of one that is positive indicates that the classification is perfect, a value that is close to zero indicates that the prediction was made by chance, and a value that is negative one indicates that the opposite prediction is perfect. Furthermore, all of the negative samples were predicted to be positive, and vice versa.

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

VII. Proposed Methodology

ALGORITHM-

Step I: Twitter dataset has been pre-processed and then encoded.

Step II: Generate models for an unbalanced dataset.

Step III: Select the top models for the evaluation of the results.

Step VI: Perform the Re-sampling techniques to balance the dataset.

Step V: Optimization of the model by applying hyperparameters.

Step VI: Prediction based on test data.

Fig. 1. Flow Chart of the proposed model.

Fig.1 gives block diagram of the model that is being suggested. These are the ways in which the model that was suggested operates: data is gathered and processed, and models are investigated to see which ones are the most appropriate for the type of data; then trains, tests, and evaluates the models. The important thing is to increase the accuracy of machine learning models. Providing labeled data to an AI system. This indicates that a label has been assigned to each piece of information in accordance with its unique significance.

VIII. Results and Discussions

Table 3 compares the various ML models on the processed unbalanced datasets various performance metrics including precision, recall, F1 score and recall for 13 different classifiers.

TABLE III. Performance Analysis of All the Models for Imbalanced Data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.9362	0.795	0.0988	0.9289	0.1776	0.1664	0.2889	1.587
knn	K Neighbors Classifier	0.9342	0.7526	0.2587	0.5665	0.3547	0.3251	0.3533	0.184
lightgbm	Light Gradient Boosting Machine	0.9314	0.7951	0.0516	0.634	0.0944	0.085	0.1642	0.458
xgboost	Extreme Gradient Boosting	0.9309	0.7987	0.0478	0.6295	0.0879	0.0784	0.1562	0.195
lr	Logistic Regression	0.9299	0.5184	0	0	0	0	0	0.612
ridge	Ridge Classifier	0.9299	0	0	0	0	0	0	0.034
qda	Quadratic Discriminant Analysis	0.9299	0.6163	0	0	0	0	0	0.037
ada	Ada Boost Classifier	0.9299	0.762	0	0	0	0	0	0.439
lda	Linear Discriminant Analysis	0.9299	0.5184	0	0	0	0	0	0.037
dummy	Dummy Classifier	0.9299	0.5	0	0	0	0	0	0.03
svm	SVM - Linear Kernel	0.9293	0	0	0	0	-0.0011	-0.0054	0.437
et	Extra Trees Classifier	0.9143	0.7032	0.3607	0.383	0.3713	0.3254	0.3256	1.307
rf	Random Forest Classifier	0.9136	0.7567	0.3684	0.3799	0.3737	0.3273	0.3275	3.604
dt	Decision Tree Classifier	0.9134	0.662	0.3697	0.3795	0.3742	0.3277	0.3279	0.133

Table 4 represents the resultant values of the sentiment analysis for various sampling strategies. SMOTE and ADASYN are used as the oversampling techniques and ALLKNN and TomekLinks are used as the under-sampling techniques.

TABLE IV. Performance Analysis of Gradient Boosting Classifier Without Using Hyper- Parameters

Gradient Boosting Classifier		
	Sampling Technique	**Accuracy**
Over-Sampling	SMOTE	0.7735
	ADASYN	0.7193
Under-Sampling	ALLKNN	0.9338
	TomekLinks	0.9366

Table 5 shows the resultant values of the sentimental analysis of the given dataset after balancing and tuning the dataset and using the different sampling techniques. The sampling techniques used for over-sampling dataset are the SMOTE and ADASYN and the sampling technique used for fetching the resultant values for the under-sampling dataset are ALLKNN and Tomek Links.

TABLE V. Performance Analysis of Gradient Boosting Classifier With Using Hyper- Parameters

Gradient Boosting Classifier		
	Sampling Technique	**Accuracy**
Over-Sampling	SMOTE	0.796
	ADASYN	0.7894
Under-Sampling	ALLKNN	0.9343
	TomekLinks	0.9368

Table 6 briefs us about the confusion matrix for the imbalanced dataset. In relation to the numbers of actual no and actual yes. Specifically, it displays the number of erroneous predictions as well as the number of true predictions. Both of their values are as:

True- Positives = 51

True- Negatives = 622

False- Positives = 7

False- Negatives = 8909

TABLE VI. Confusion Matrix for Imbalanced Data

	Predicted No	**Predicted Yes**
Actual No	8909	7
Actual Yes	622	51

IX. Conclusion

Improving the accuracy of the model that has been suggested is the primary goal of the research that has been proposed. Consequently, in order to accomplish this goal, the presence of a balanced dataset is one of the primary prerequisites. This is because an imbalanced dataset can result in bias towards classes that have a greater number of samples. Every classifier has its unique set of benefits, which are determined by the kind of data that is used for training. It is seen that the results of the Gradient boosting classifier were superior to those of the other classifiers. The purpose of this endeavour is to employ a variety of optimization strategies aimed at selecting the most advantageous features, in conjunction with resampling techniques, in order to ensure that classifiers produce satisfactory outcomes. The first step is to select the five most accurate classifiers and nominate them for further consideration. Each model is assessed according to its level of accuracy. In conclusion, the Gradient boosting classifier beats the other classifiers in terms of accuracy, surpassing prior efforts that were performed with the same dataset by a margin of 93.62%. The work that will be done in the future can involve other datasets to work on.

References

[1] Agarwal, Apoorv, et al. "Sentiment analysis of twitter data." *Proceedings of the workshop on language in social media (LSM 2011)*. 2011.

[2] Wang, Yili, et al. "Sentiment analysis of Twitter data." *Applied Sciences* 12.22 (2022): 11775.

[3] Chalothom, Tawunrat, and Jeremy Ellman. "Simple approaches of sentiment analysis via ensemble learning." *information science and applications*. Springer Berlin Heidelberg, 2015.

[4] Xia, Rui, Chengqing Zong, and Shoushan Li. "Ensemble of feature sets and classification algorithms for sentiment classification." *Information sciences* 181.6 (2011): 1138-1152.

[5] Saif, Hassan, Yulan He, and Harith Alani. "Semantic sentiment analysis of twitter." *The Semantic Web–ISWC 2012: 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I 11*. Springer Berlin Heidelberg, 2012.

[6] Gautam, Geetika, and Divakar Yadav. "Sentiment analysis of twitter data using machine learning approaches and semantic analysis." *2014 Seventh international conference on contemporary computing (IC3)*. IEEE, 2014.

[7] Neelakandan, S., and D. Paulraj. "A gradient boosted decision tree-based sentiment classification of twitter data." *International Journal of Wavelets, Multiresolution and Information Processing* 18.04 (2020): 2050027.

[8] Le, Bac, and Huy Nguyen. "Twitter sentiment analysis using machine learning techniques." *Advanced Computational Methods for Knowledge Engineering: Proceedings of 3rd International Conference on Computer Science, Applied Mathematics and Applications-ICCSAMA 2015*. Springer International Publishing, 2015.

[9] Sahayak, Varsha, Vijaya Shete, and Apashabi Pathan. "Sentiment analysis on twitter data." *International Journal of Innovative Research in Advanced Engineering (IJIRAE)* 2.1 (2015): 178-183.

[10] Neethu, M. S., and R. Rajasree. "Sentiment analysis in twitter using machine learning techniques." *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*. IEEE, 2013.

[11] Sankar, H., and V. Subramaniyaswamy. "Investigating sentiment analysis using machine learning approach." *2017 International conference on intelligent sustainable systems (ICISS)*. IEEE, 2017.

[12] Alsaeedi, Abdullah, and Mohammad Zubair Khan. "A study on sentiment analysis techniques of Twitter data." *International Journal of Advanced Computer Science and Applications* 10.2 (2019): 361-374.

[13] Kharde, Vishal, and Prof Sonawane. "Sentiment analysis of twitter data: a survey of techniques." *arXiv preprint arXiv:1601.06971* (2016).

[14] Anjaria, Malhar, and Ram Mohana Reddy Guddeti. "Influence factor based opinion mining of Twitter data using supervised learning." *2014 sixth international conference on communication systems and networks (COMSNETS)*. IEEE, 2014.

[15] Jain, Anuja P., and Padma Dandannavar. "Application of machine learning techniques to sentiment analysis." *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. IEEE, 2016.

[16] Sahayak, Varsha, Vijaya Shete, and Apashabi Pathan. "Sentiment analysis on twitter data." *International Journal of Innovative Research in Advanced Engineering (IJIRAE)* 2.1 (2015): 178-183.

[17] Da Silva, Nadia FF, Eduardo R. Hruschka, and Estevam R. Hruschka Jr. "Tweet sentiment analysis with classifier ensembles." *Decision support systems* 66 (2014): 170-179.

[18] Jagdale, Rajkumar S., Vishal S. Shirsat, and Sachin N. Deshmukh. "Sentiment analysis on product reviews using machine learning techniques." *Cognitive Informatics and Soft Computing: Proceeding of CISC 2017*. Springer Singapore, 2019.

[19] Singh, Jaspreet, Gurvinder Singh, and Rajinder Singh. "Optimization of sentiment analysis using machine learning classifiers." *Human-centric Computing and information Sciences* 7 (2017): 1-12.

[20] Ain, Qurat Tul, et al. "Sentiment analysis using deep learning techniques: a review." *International Journal of Advanced Computer Science and Applications* 8.6 (2017).

[21] Rathi, Megha, et al. "Sentiment analysis of tweets using machine learning approach." *2018 Eleventh international conference on contemporary computing (IC3)*. IEEE, 2018.

[22] Chugh, Bharti, and Nitin Malik. "Machine Learning Classifiers for Detecting Credit Card Fraudulent Transactions." *Information and Communication Technology for Competitive Strategies (ICTCS 2021) ICT: Applications and Social Interfaces*. Singapore: Springer Nature Singapore, 2022. 223-231.

[23] Qi, Yuxing, and Zahratu Shabrina. "Sentiment analysis using Twitter data: a comparative application of lexicon-and machine-learning-based approach." *Social Network Analysis and Mining* 13.1 (2023): 31.

[24] Pasrija, Paridhi, Utkarsh Singh, and Mehak Khurana. "Performance Analysis of Intrusion Detection System Using ML Techniques." *Applying Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection* (2024): 135-150.

APPENDIX 2

Similarity Report

18% Overall Similarity

Top sources found in the following databases:

- 11% Internet database
- Crossref database
- 12% Submitted Works database
- 5% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	cse.anits.edu.in Internet	3%
2	KIET Group of Institutions, Ghaziabad on 2024-04-19 Submitted works	2%
3	coursehero.com Internet	2%
4	KIET Group of Institutions, Ghaziabad on 2024-05-15 Submitted works	1%
5	geeksforgeeks.org Internet	1%
6	Canterbury Christ Church University on 2020-05-30 Submitted works	<1%
7	Somaiya Vidyavihar on 2022-05-03 Submitted works	<1%
8	turcomat.org Internet	<1%

Sources overview

9	University of Glamorgan on 2023-08-24 Submitted works	<1%
10	sinergiejournal.eu Internet	<1%
11	Onaizah Colleges on 2024-05-12 Submitted works	<1%
12	medium.com Internet	<1%
13	ibm.com Internet	<1%
14	WorldQuant University on 2023-07-04 Submitted works	<1%
15	subhrankurretail.com Internet	<1%
16	Aston University on 2023-09-29 Submitted works	<1%
17	University of Southampton on 2022-05-30 Submitted works	<1%
18	Uttaranchal University, Dehradun on 2023-12-30 Submitted works	<1%
19	Arts, Sciences & Technology University In Lebanon on 2024-02-27 Submitted works	<1%
20	ijcjournal.org Internet	<1%

Sources overview