**KISII UNIVERSITY**

**SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY**

**A PROJECT REPORT ON**

**SENTIMENT ANALYSIS ON X(TWITTER) USING MACHINE LEARNING TECHNIQUES**

**BY**

**KENNEDY OKONDA (IN13/00046/20)**

**KIMBERLY MWACHILUNGO (IN13/00101/21)**

**SUBMITTED**

**UNDER THE GUIDANCE OF:**

**MDM. TERESA ABUYA**

**SEPTEMBER, 2024.**

**Table of contents.**

# **Abstract**

Social media websites have emerged as one of the platforms to raise users’ opinions and influence how any business is commercialized. People's opinions matter a lot in analyzing how information propagation impacts lives in a large-scale network like X formerly Twitter. Data analysis of the tweets determines the polarity and inclination of a vast population toward a specific topic, item, or entity. These days, the applications of such analysis can be easily observed during public elections, movie promotions, brand endorsements, and many other fields. In this project, we will make a program that analyzes the nature of tweets on a particular topic. The primary aim is to provide a method for analyzing polarity scores in noisy twitter streams. This paper reports on the design of a data analysis, extracting a vast number of tweets.

Results classify user's perceptions via tweets into positive and negative. In this project, we will make a program that analyzes the nature of tweets on a particular topic. The user will be able to input a keyword(hashtag) and get the nature of it based on the latest tweets that contain the input keyword. Each tweet extracted is classified based on its sentiment whether it is positive or negative. Data were collected from movie reviews which were on the IMDB Website. Naïve Bayes machine learning algorithm was used. The result from this model was tested using various testing metrics. Moreover, our model demonstrates strong performance in mining texts extracted directly from X.

# **INTRODUCTION**

In today’s world, the internet has become a critical medium for individuals to express their opinions on various topics. Platforms such as blogs, discussion forums, review sites, and social media have enabled people to share their perspectives on a global scale. (Halibas et al., 2018) said people worldwide extensively rely on this user-generated content for insights, whether they're researching a product before purchasing or assessing public sentiment on key issues. Given the enormous volume of such data, it is practically impossible for individuals to read through every review or comment manually.

Sentiment Analysis (SA) is a key application of ML, designed to understand the sentiment behind a given piece of text. SA categorizes opinions as positive, negative, or neutral, helping organizations make informed decisions based on public sentiment, (Trupthi et al., 2017). It is proposed to go beyond traditional knowledge-based methods by utilizing sophisticated ML algorithms to achieve higher accuracy in determining sentiment polarity the extent to which a sentiment is positive or negative (Neethu & Rajasree, 2013 ).

To achieve accurate sentiment predictions, several supervised ML algorithms are used, including Decision tree, Logistic Regression, and Random forest. These models are trained on pre-processed X data to classify sentiments effectively. M. Trupthi, S.Pabboju, and G. Narasimha proposed the system's objective is to automatically categorize text data into positive, negative, or neutral sentiments, offering a graphical representation of the results using bar graphs and pie charts. (Joshi & Tekchandani, 2016 )stated Sentiment analysis has gained considerable attention from businesses, governments, and researchers, given its wide applications. Companies increasingly leverage social media data to monitor public opinion on their products and services. For instance, brands can gauge customer satisfaction, evaluate product launches, and enhance marketing strategies by analyzing customer feedback on platforms like X. (Agarwal et al), added, that sentiment analysis extends to various other domains such as politics, healthcare, and finance, providing insights into public opinions on political campaigns, medical treatments, or market trends.

This work focuses on implementing NLP techniques for sentiment analysis, particularly using X data, to support applications like brand reputation management and market research. (Pang & Lee, 2008) stated it outlines a systematic approach to collecting, cleaning, and analyzing social media data, using machine learning algorithms to detect patterns in user sentiment. The paper also delves into the challenges faced during sentiment analysis, including the complexity of informal language, sarcasm, and domain-specific language nuances. As a result, automating this process has become crucial, with Machine Learning (ML) playing a significant role in understanding and categorizing sentiments expressed online.

This paper proposes a system for sentiment analysis using data from X, a platform that generates an immense volume of data through short, unfiltered, and often noisy messages called tweets. (Pang and Lee 2002) proposed with over 561 million tweets published daily, X has become a vital source for capturing public sentiment on trending topics, making it an ideal platform for sentiment analysis in applications such as brand reputation management and market research. (Kumar & Sebastian,2017), expressed X character limit encourages the use of abbreviations, emoticons, and hashtags, making the text challenging to analyze. However, advanced NLP techniques, such as feature extraction and emoticon mapping, allow us to process and classify the sentiments expressed in tweets accurately.

# **Contribution to the work.**

The primary goal of sentiment analysis on tweets is to accurately classify the text into predefined sentiment categories, typically positive, negative, or neutral. In recent years, numerous approaches have been developed to enhance the accuracy and efficiency of such classifications. These approaches range from traditional machine learning algorithms to modern deep learning techniques, and their effectiveness is typically evaluated through rigorous testing and validation.

We aim to develop and implement a sentiment analysis algorithm that automatically classifies tweets into positive, negative, or neutral categories. By analyzing the sentiment expressed in tweets, we seek to gauge public opinion on various subjects, providing insight into the overall mood or stance of the population. Additionally, we strive to enhance the accuracy of our system by integrating advanced techniques, such as word embeddings and transformer models, to improve upon existing methods. Furthermore, we aim to represent the sentiment distribution visually, utilizing graphical representations like bar graphs, pie charts, and line graphs, which will offer a clear and intuitive interpretation of the results and trends.

This work will also explore the challenges of sentiment analysis, such as dealing with sarcasm or ambiguous language, and propose methods to overcome these obstacles, contributing to advancements in the field.

## **Related Work**

Sentiment analysis is the careful examination of how feelings and points of view can be identified with one’s feelings and mentality in regular language regarding an occasion. The principle motivation behind choosing X’s profile information is that subjective data can be obtained from this platform. Ongoing occasions show that sentiment analysis has reached incredible accomplishment which can outperform the positive versus negative and manage the entire field of behavior and feelings for various networks and themes. In the field of sentiment analysis utilizing various techniques, a great measure of exploration has been done for the expectation of social sentiments. (Pang and Lee 2012) proposed the framework, where an assessment can be positive or negative was discovered by the proportion of positive words to total words. Later in 2018, the creator built up a methodology in which tweet results can be chosen by term in the tweet.

Another study on X sentiment analysis was done by (Go et al 2019). who stated the issue as a two-class classification, meaning to characterize tweets into positive and negative classes.( M. Trupthi, S.Pabboju, and G. Narasimha) proposed a system that mainly makes use of Hadoop. The data is extracted using SNS services which are done using X’s streaming API. The tweets are loaded into Hadoop and are preprocessed using map-reduce functions. They have made use of uni-word naive Bayes classification. The paper analyzes the utilization of SA in business applications. Besides, this paper exhibits the text analysis process in auditing the popular assessment of clients toward a specific brand and presents hidden information that can be utilized for decision-making after the text analysis is performed. In the paper, the sentiment analysis has been done in 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.

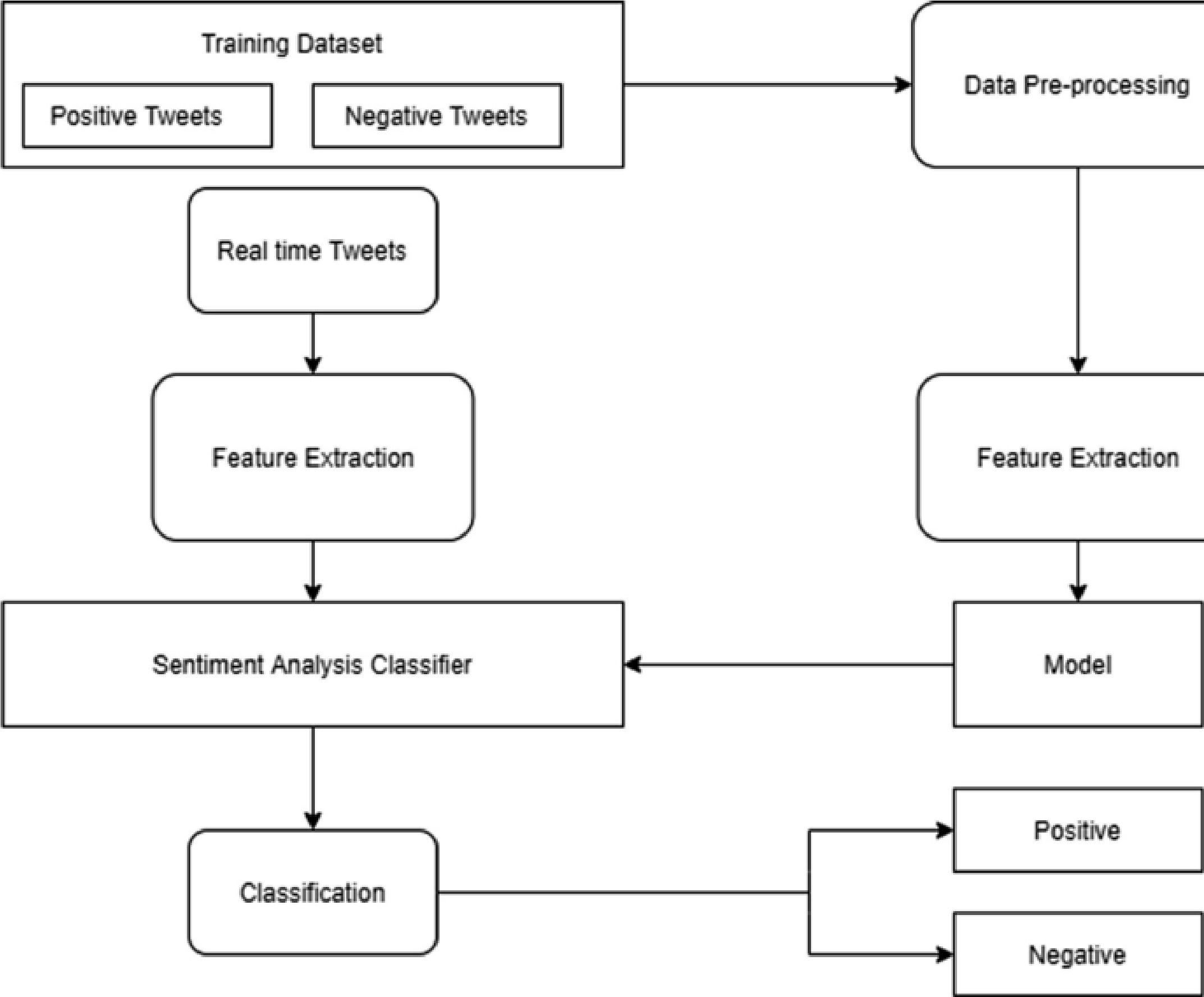
The proposed system is domain-specific. A user interactive GUI will be available for the users to type in the keywords related only to the commercial products. Not many existing systems have

Been made so specific. Also, the system aims to compare various ML algorithms and choose the one that will produce results with the highest accuracy. Making the system domain specific reduces processing time as tweets regarding specific products are only searched based on the keywords typed.

## **Proposed System**

The system intends to carry out sentiment analysis over tweets gathered from the Twitter dataset. Various algorithms have been utilized and tested against the available dataset, and the most appropriate algorithm has been chosen. This gives an idea of how the sentiment analysis will be carried out. Once the dataset has been cleaned and divided (isolated) into preparing (training) and testing datasets, it will be pre-processed using the techniques mentioned below. Features will be extracted to reduce the dimension of the dataset. The next stage is to create a model that will be given to the classifier to classify the tweets into positive and negative tweets. Again, real-time tweets will be given to the classifier for testing the real-time data. The proposed system does not engage in performing sentiment analysis on every tweet belonging to every other domain. The system is strictly domain-restricted, where the sentiment analysis is performed to classify the tweets related to products in the market into a negative or positive category. The end-user will be provided with an interactive GUI wherein he/she can type the keywords or sentences related to a particular product. All tweets which are identified with that product will be available to the user. The user will be able to see the number of positive and negative statements made by others. This will help them in revising their production and work strategies accordingly which will be useful in improving their businesses.

Outline of the proposed system



1.

i.

ii.

iii.

Below are the steps involved in handling large incoming data:

Data cleaning:

1. Use of various data tools that can help in cleaning the dataset.
2. Use of several AI tools that help identify duplicates in large data corporations and eliminate them.
3. For correcting the corrupted data, the source of errors should be tracked and monitored constantly.
4. Validate once, when the existing data is cleaned.

Data pre-processing:

1. Assessing data quality.
2. Identification of inconsistent values to know what the data type of the features should be.
3. Aggregate the features to give better performance.

## **Data Pre-processing**

Data preprocessing is a crucial step in sentiment analysis, as it directly influences the quality of feature extraction and the performance of classification algorithms. Since tweets are often unstructured and contain a wide variety of noise, including slang, abbreviations, and special characters, preprocessing helps convert this raw data into a more structured format suitable for machine learning algorithms. This process involves several tasks such as cleaning, normalizing, and transforming the data to make it more understandable for models, ultimately improving classification accuracy and reducing computational complexity. The steps include:

**1. Data Quality Assessment and Cleaning** The first step in data preprocessing is assessing and improving the quality of the data. Tweets are typically collected from various sources, making it unrealistic to expect perfect consistency. An initial quality check should focus on identifying and correcting inconsistencies, such as missing values, duplicated entries, and irrelevant features (Pak & Paroubek, 2010). For instance, a common issue in Twitter data is the presence of usernames, URLs, and retweets, which do not carry sentiment information and can distort the analysis. Therefore, they should either be removed or replaced with placeholder tokens like "USER\_MENTION" or "URL" (Pak & Paroubek, 2010).

Next, the text is cleaned to remove any HTML tags, special characters, and numbers that may skew the results. Python’s regular expressions are often used to detect and eliminate non-alphanumeric characters (Barbosa & Feng, 2010). Additionally, tweets are converted to lowercase to ensure that the model does not differentiate between "Happy" and "happy." This step is essential because case-sensitive analysis can mistakenly interpret different capitalizations of the same word as separate features (Salton & McGill, 1983).

**2. Tokenization and Stop Words Removal** Tokenization is the process of breaking down a tweet into its individual words or tokens. This step is fundamental because it transforms a text into its component units, making it easier for machine learning algorithms to process. During tokenization, each tweet is split into words based on spaces and punctuation, though special handling is required for emoticons, URLs, and abbreviations, which are common in social media data (Pak & Paroubek, 2010).

Following tokenization, stop words are removed. Stop words are commonly used words like "the," "is," and "at," which appear frequently but do not contribute meaningfully to the sentiment of the text (Salton & McGill, 1983). By removing these words, the dimensionality of the dataset is reduced, allowing the model to focus on more relevant features. Stop word removal is especially useful in sentiment analysis because it helps eliminate noise and improves computational efficiency.

**3. Stemming and Lemmatization** Stemming and lemmatization are processes used to reduce words to their base or root forms. Stemming involves chopping off affixes from words to arrive at the stem, while lemmatization maps words to their dictionary form based on the context in which they are used (Porter, 1980). For example, words like "running," "ran," and "runner" are reduced to "run" during stemming. This reduction helps minimize the vocabulary size and improves the generalization of the model (Manning & Schutze, 1999).

Lemmatization, on the other hand, is more sophisticated than stemming because it considers the grammatical structure of the sentence. For instance, the words "better" and "good" may appear in different forms, but lemmatization maps them to their appropriate base forms based on their role in the sentence (Manning & Schutze, 1999). Although lemmatization requires more computational resources than stemming, it generally produces better results for sentiment analysis.

**4. Handling Negations** Negations are words like "not" and "no" that can reverse the sentiment of a sentence. Handling negations is a critical preprocessing step because failing to account for them can lead to incorrect sentiment classification (Councill et al., 2010). For instance, the sentence "I don’t like this movie" has a negative sentiment, but if the negation word "don’t" is ignored, the sentiment may be wrongly classified as positive due to the word "like."

A simple approach to handling negations is to reverse the polarity of all words between the negation word and the next punctuation mark (Councill et al., 2010). For example, in the phrase "I don’t enjoy watching TV," the polarity of "enjoy" and "watching" would be flipped from positive to negative. More advanced methods use contextual valence shifters, which not only handle negations but also consider intensifiers and diminishers that alter the strength of the sentiment (Polanyi & Zaenen, 2006).

**5. Part-of-Speech Tagging (POS)** Part-of-speech tagging assigns a grammatical label, such as noun, verb, or adjective, to each word in a sentence. POS tagging is used in sentiment analysis to identify patterns that could provide deeper insights into the sentiment of the text. For example, subjective texts often contain more pronouns and modal verbs, while neutral or factual texts are likely to contain more common nouns (Pak & Paroubek, 2010).

The effectiveness of POS tagging in sentiment analysis is still a subject of debate. While some studies report that POS tagging improves the accuracy of sentiment classification (Barbosa & Feng, 2010), others suggest that it may reduce performance, depending on the dataset and algorithm used (Kouloumpis et al., 2011). Therefore, its application should be evaluated on a case-by-case basis.

**6. N-grams and Feature Extraction** N-grams are contiguous sequences of words in a text. Unigrams consist of single words, bigrams consist of two words, and trigrams consist of three words. N-gram models are used in sentiment analysis to capture word combinations that are often missed by single-word analysis (Zhang, 2003). For instance, the phrase "not good" conveys a negative sentiment, but if the words "not" and "good" are analyzed separately, the model may misinterpret the sentiment as positive (Tan et al., 2002).

Feature extraction in sentiment analysis involves selecting the most relevant n-grams and other features from the dataset. Common features include the frequency of unigrams, bigrams, and trigrams, as well as the presence of specific words or symbols like hashtags, emoticons, and user mentions (Pak & Paroubek, 2010). These features are then used to train machine learning models to predict the sentiment of new tweets.

**7. Handling Special Characters, URLs, and Usernames** Tweets often contain non-sentiment-bearing features like URLs, usernames, and hashtags, which can add noise to the dataset. Special characters, digits, and non-alphanumeric symbols should be removed because they do not contribute to the sentiment analysis (Pak & Paroubek, 2010). Similarly, usernames and URLs can be replaced with placeholder tokens like "USER\_MENTION" and "URL" to maintain the sentence structure without affecting the sentiment (Pak & Paroubek, 2010).

**8. Spelling Correction and Slang Expansion** Since Twitter data often contains slang, abbreviations, and spelling errors, it is essential to correct these issues during preprocessing. Spelling correction ensures that common misspellings are normalized, while slang expansion maps informal expressions to their formal equivalents (Barbosa & Feng, 2010). For example, "gonna" can be expanded to "going to," and "luv" can be expanded to "love." These transformations help improve the model’s understanding of the text and ensure that important features are not overlooked.

The above-stated steps improve the efficiency of the data preprocessing and are critical for achieving high accuracy in sentiment analysis, particularly when working with unstructured data like tweets. By following a series of structured steps—such as cleaning, tokenization, stop word removal, stemming, lemmatization, negation handling, and feature extraction—researchers can significantly enhance the quality of the input data and, consequently, the performance of the machine learning model. While each preprocessing step has its benefits and challenges, their combined application creates a more structured and meaningful dataset for sentiment classification.

**Feature selection**

Since the main features of a text classifier are N-grams, the dimensionality of the feature space grows proportionally to the size of the dataset. This dramatical growth of the feature space makes it in most cases computationally infeasible to calculate all the features of a sample. Many features are redundant or irrelevant and do not significantly improve the results. Feature selection is the process of identifying a subset of features that have the highest predictive power. This step is crucial for the classification process, since elimination of irrelevant and redundant features allows to reduce the size of feature space increasing the speed of the algorithm, avoiding overfitting as well as contributing to the improved quality of classification. Here we used the following two feature selection techniques:

**Recursive Feature Elimination (RFE)** , where we iteratively train the model and eliminate the least important features based on the model's performance. In a sentiment analysis context, we might start with a large set of words and iteratively remove those that contribute the least to the prediction accuracy. For example, if terms like "the" or "is" are found to have minimal impact, RFE would remove them, allowing the model to focus on more meaningful features.

Here, Two types of features are extracted from the dataset, namely unigrams and bigrams. A frequency distribution is created for the extracted features. Later, the top N unigrams and bigrams are chosen to carry out the analysis. Also, tweets contain special features like URLs, user names, emoticons, etc. Retweets are also a feature of tweets. These features are not required while performing sentiment analysis.

Hence, these features are replaced with common keywords or markers like “URL,” “USER\_MENTION,” “EMO,” respectively

**L1 Regularization (Lasso)** is a method that used also to perform feature selection during model training. In sentiment analysis, a logistic regression model could be applied with L1 regularization to encourage sparsity in the feature set. This means that irrelevant features (e.g., common stop words or unrelated terms) would receive zero coefficients, effectively removing them from the model. This helps streamline the analysis and improve the focus on features that actually influence sentiment.

## **Classifiers to Be Used**

**Random Forest**

This classifier comprises of a classification tree T(J,Ki),i=1,2,⋯,qT(J,Ki),i=1,2,⋯,q . Here, KiKi represents a vector that is identically and independently distributed (IID) to each tree vote at its input *J*. In short, a random forest combines several decision trees to minimize overfitting. Suppose that T1(S),T2(S),⋯,Tq(S)T1(S),T2(S),⋯,Tq(S) is an ensemble classifier with arbitrary training data got from vector *S* and *Q* (the prediction class), *f* is the indicator function, A⌣A⌣ is the mean, the margin function is formulated as in (24):

Mg(S,Q) =f(T1(s)=Q)- T1(S)=R)

In (24), Ti(S)=QTi(S)=Q denote classification result while Ti(S)=RTi(S)=R is classification result with *R*. In RF, the margin is utilized to establish the mean value of votes *S* and *Q*, such that the greater the margin, the more accurate is the classification. Here, the generalization error GˆG^ is derived as in (25):

= WS,Q mg(S,Q)<0)

In (25), *WS*,*Q* signifies that the probability is more than *S*, *Q* dimension. Considering training sample *Ţ*p={(α1,β1),⋯,(αp,βp)}Ţp={(α1,β1),⋯,(αp,βp)} of IID [0,1]l[0,1]l . Using *Ţp*, the objective is to estimate the regression function RF(α)=E[β[α=g]]RF(α)=E[β[α=g]] for some fixed g∈[0,1]lg∈[0,1]l . Generally, RF classifier consists of a set of stochastic regression tree {RT(g,hq,*Ţ*p),q≥1}{RT(g,hq,Ţp),q≥1} . Here, h1,h2,⋯h1,h2,⋯ Denotethe IID outputs of a randomization construct *h*. By combining these random trees (*RT*s), an amalgamated regression estimate is obtained as below:

T(α, TP) = Eh[RT(g,h, Tp)]

In (26), EhEh is conditionally associated with random constructs on gg and *Ţp*. Here, the dependency of sample estimates is denoted as T(g)R¯T(g) and *h* is utilized to establish how successive divisions are executed when building individual trees.

**Decision tree**

This shows the classification report of the Decision Tree model. The model achieved a training accuracy of 99.92% and a validation accuracy of 93.20%. This indicates that while the model performs nearly perfectly on the training data, its performance on unseen validation data is slightly lower but still strong.

A classification report was generated to evaluate the performance on the validation set. The confusion matrix showed that the model correctly classified most of the examples, with a slight number of misclassifications as reflected in the validation F1 score of 0.54

**Classification Report for Decision Tree**  
ROC curves are appropriate when the observations are balanced between each class. Since the dataset used for training and testing is balanced, ROC curves were used to measure model performance. The Decision Tree model's ROC curve showed competitive performance compared to other algorithms.

AUROC is a superior measure of classifier performance as it provides an unbiased summary of the model's performance across different thresholds. Figure 5 shows a comparison between the three algorithms used for sentiment analysis; Decision Tree achieved the highest validation accuracy of 93.20%. However, the F1 score and the AUROC for the Decision Tree suggest it slightly underperformed compared to logistic regression in terms of balanced performance across precision, recall, and F1 score.

Therefore, while the Decision Tree excels in accuracy, logistic regression may offer more balanced performance in this context.

**Logistic regression.**

Logistic regression predicts a binary outcome, i.e., (Y/N) or (1/0) or (True/False). It also works as a special case of linear regression. It produces an S-shaped curve better known as a sigmoid. It takes real values between 0 and 1.

The model of logistic regression is given by:

|  |
| --- |
| Output:0or1  Hypothesis:Z = WX + B  hθ (x) = sigmoid(Z) |

Basically, logistic regression has a binary target variable. There can be categories of target variables that can be predicted by it. The logistic classifier uses a cross-validation estimator.

Support vector machine. It is a non-probabilistic model that utilizes a portrayal of text models as focuses in a multidimensional space. These examples are mapped with the goal that the instances of the diverse categories (sentiments) have a place in particular areas of that space. Later, the new messages are mapped onto that equivalent space and are predicted to have a place with a classification dependent on which category they fall into. In the SVM algorithm, the fundamental goal is to boost the edge between information points and the hyperplane. The loss function that helps with this is called a hinge loss.

|  |  |
| --- | --- |
|  |  |

(

x

)

)

=

{

0

,

i

f

y

∗

f

(

x

)

≥

1

1

−

y

∗

f

(

x

)

,

else

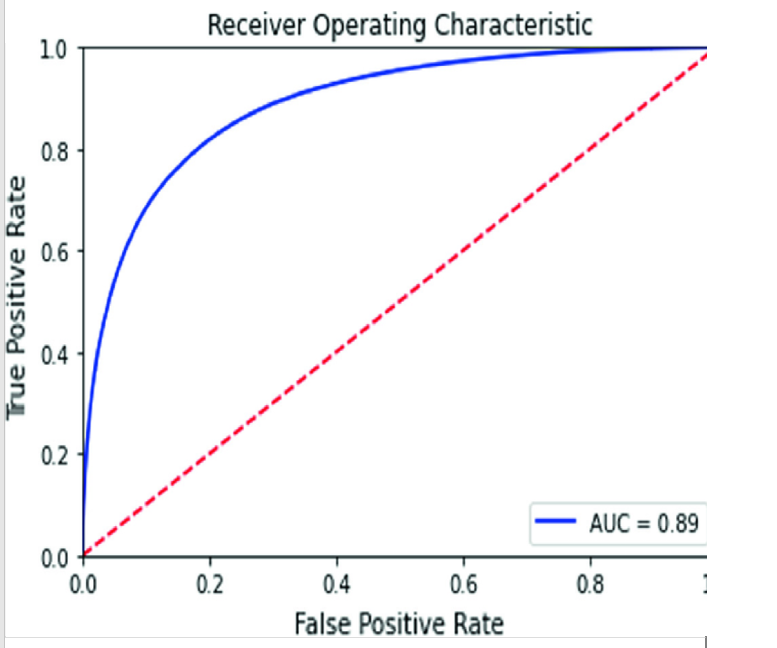
The cost is 0 if the predicted, and the actual value is of a similar sign. On the off chance that they are not, at that point, figure the loss esteem.

For performing sentiment analysis, logistic regression is considered over naive Bayes because naive Bayes assumes all the features used in model building to be conditionally independent whereas logistic regression splits feature space linearly and typically works reasonably well even when some of the variables are correlated, and on the other hand, logistic regression and SVM with a linear kernel have similar performance but depending on the features, one may be more efficient than the other.

## **Plotting Results**

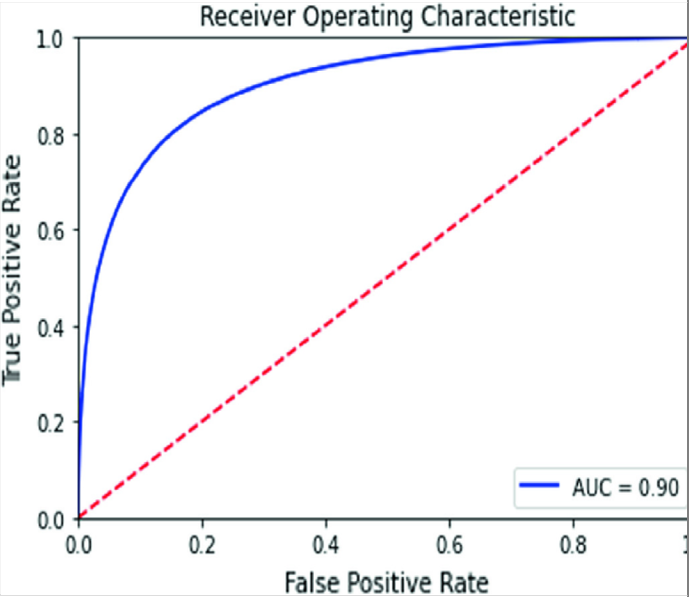
The results will be visualized using graphs, with the comparison of algorithms presented using a ROC curve. The ROC (Receiver Operating Characteristic) curve is a plot of the true positive rate (TPR) against the false positive rate (FPR). The ROC curve for the Random Forest, Decision Tree, and Logistic Regression models are shown below alongside other representations. The Area Under the ROC curve (AUROC) for these models provides a measure of their performance.

For instance, the AUROC for the Random Forest model is 0.89, demonstrating its ability to distinguish between classes. Similar ROC curves and AUROC values will be generated for the Decision Tree and Logistic Regression classifiers, allowing for a comprehensive comparison of their performance in terms of classification accuracy.



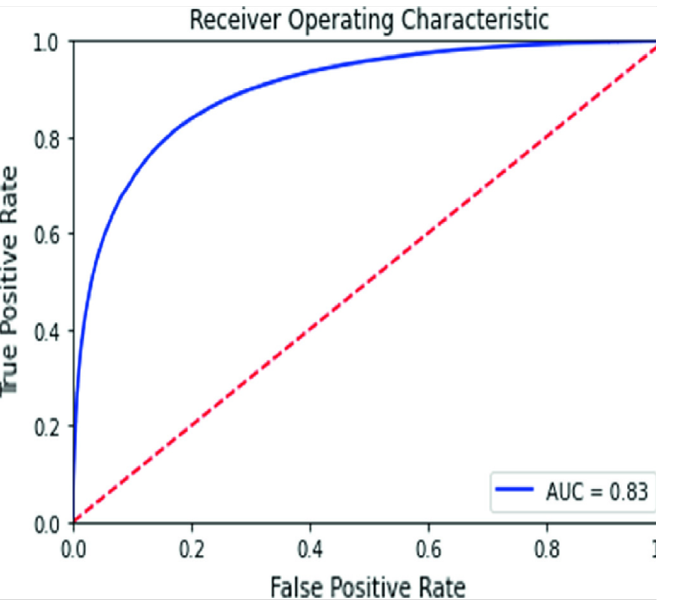
This shows the ROC curve for a logistic regression model. The Area Under the Receiver Operating Characteristics curve (AUROC) curve for the logistic regression model is 0.89.

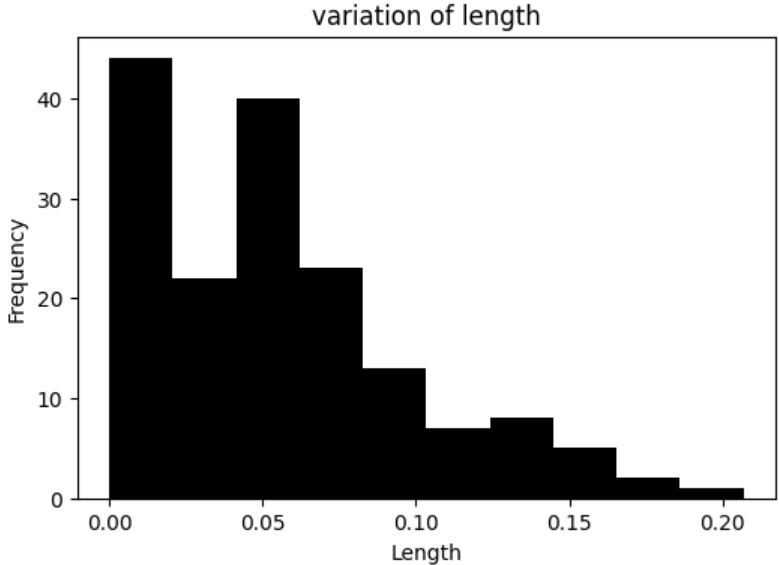
ROC curve for logistic regression classifier



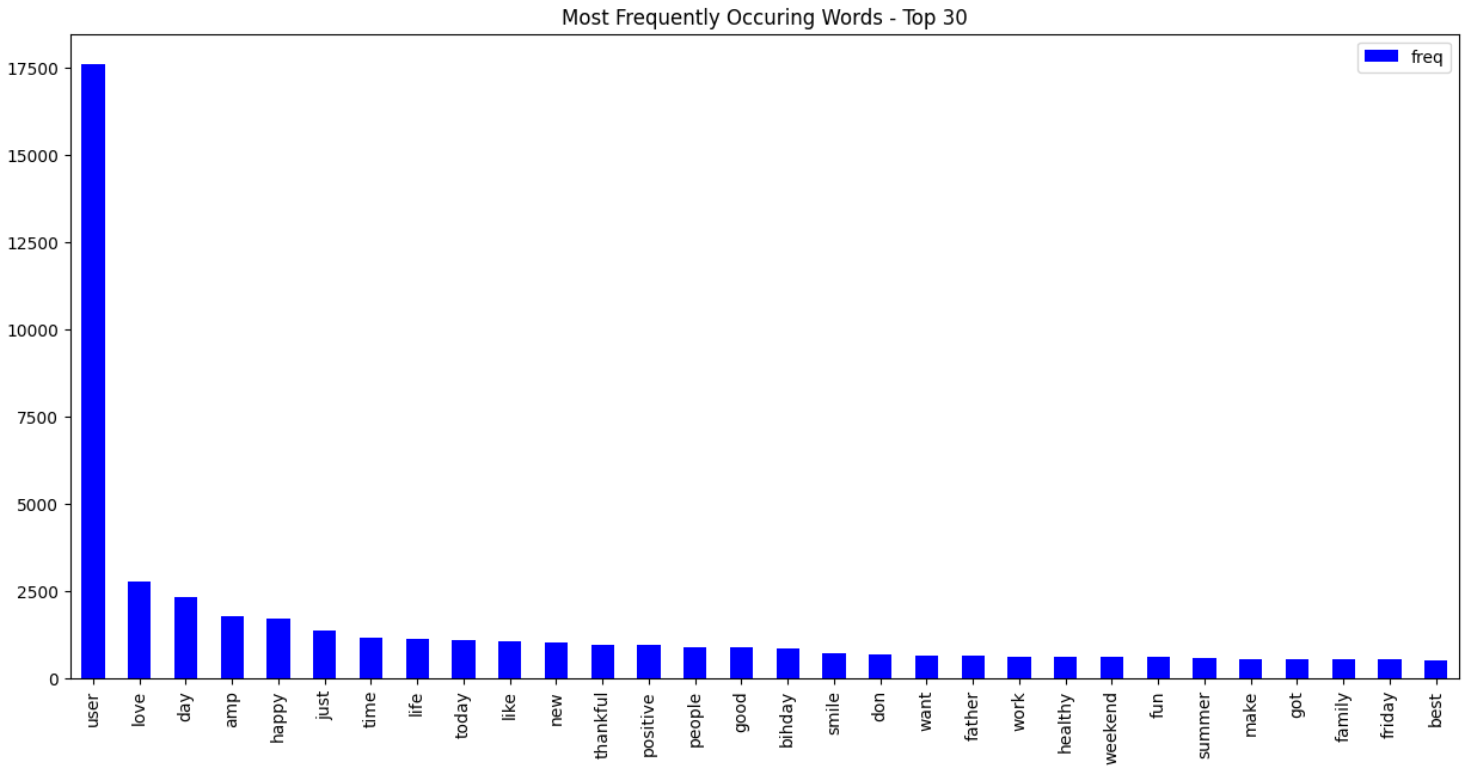
shows the ROC curve for a linear SVC model. The Area Under the Receiver Operating Characteristics curve (AUROC) for the linear SVC model is

ROC curve for linear SVC

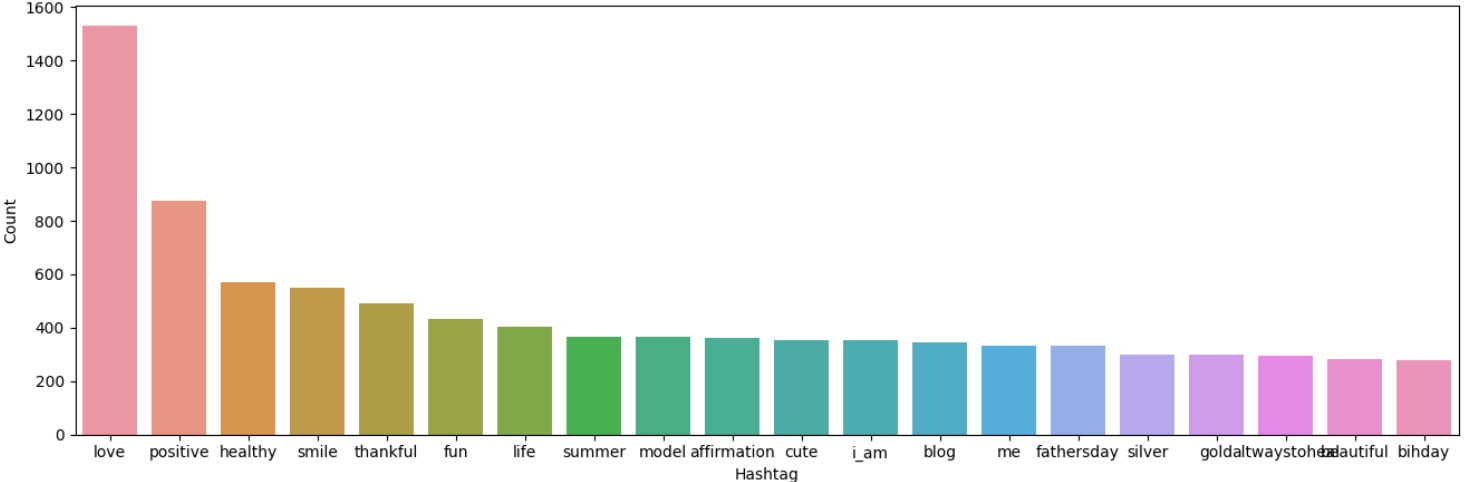


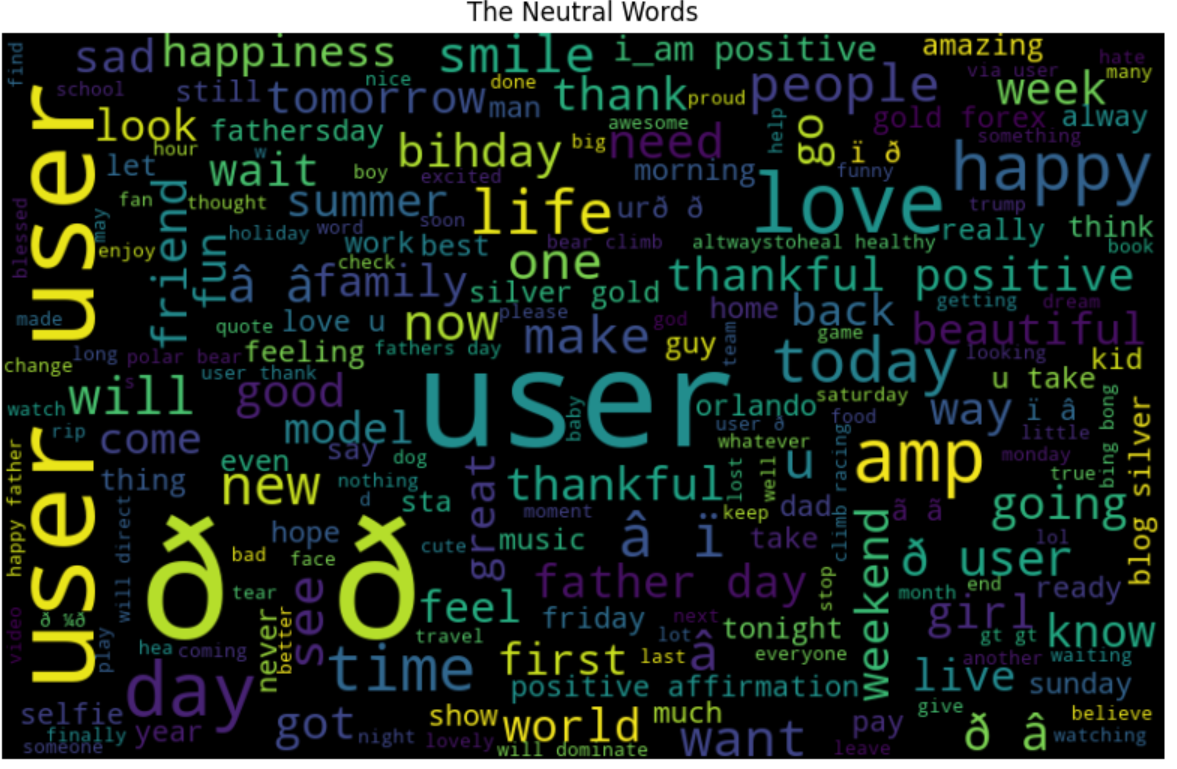


This plot depicts the relationship between word length and their frequency of occurrence in the dataset. Shorter words, typically those between 3 to 5 characters, are more frequent, while longer words appear less often. The trend reflects common language usage patterns, where shorter words like "the," "and," or "love" tend to be used more frequently than longer, more complex words. This variation in word length contributes to understanding the structure and nature of the text data.



This bar chart illustrates the 30 most frequently occurring words in the dataset. The word "user" appears most often, followed by words like "love," "day," "happy," and "amp." This visualization highlights common themes in the dataset, likely related to positive or neutral sentiments expressed in the text.



This bar chart shows the most frequently used hashtags in the dataset. The hashtag "#love" dominates the chart, followed by others like "#positive," "#healthy," "#smile," and "#thankful." These hashtags indicate prevalent positive themes and self-affirmation topics in the data, suggesting a focus on well-being and positivity in user-generated content.

The word cloud represents the distribution of neutral words within the text data. The size of each word corresponds to its frequency of occurrence. Words such as "user," "love," "thankful," "positive," and "happy" appear prominently, suggesting that these concepts are central in the analyzed content. The variety of words reflects a broad range of sentiments and common terms used in neutral contexts.

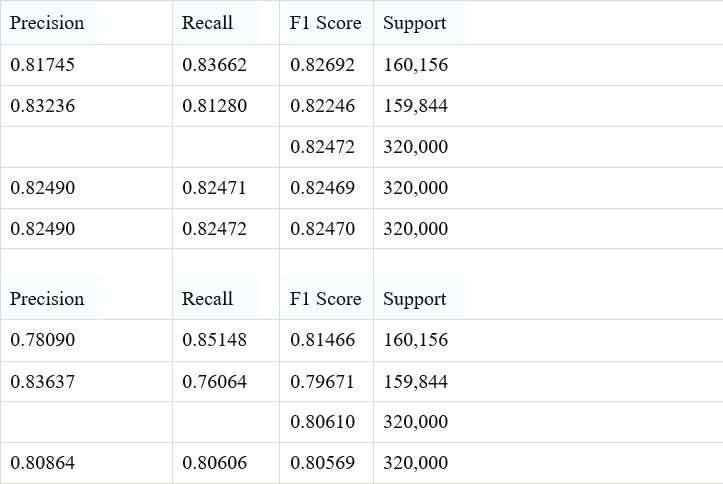
## **9. Results**

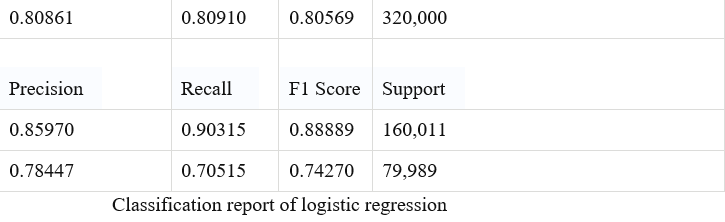
The dataset used for the training model is the Sentiment140 dataset. It is a balanced dataset with 1.6 million tweets among which 800,000 tweets belong to the positive class, and the remaining 800,000 tweets belong to negative class. The splitting is done using the train\_test\_split method with a test size of 0.20. 1.2million tweets are used for training the model, and the remaining 400,000 tweets are used for testing the model.

The classification report of the logistic regression model is given below. The accuracy of the model is 82.47 percent. It also shows the precision and recall of the model. Precision is the positive predictive value, and recall is the sensitivity of the model.

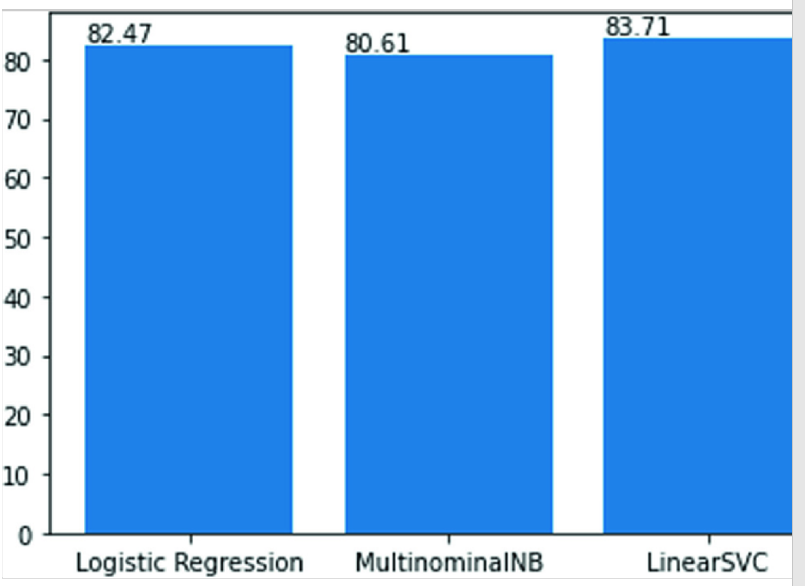
Here precision and Recall were used to measure the performance of the classifies but however F1-Score was used because in the extreme cases the recall can reach 100%, but precision can be very low.

Therefore, f-score=





AUROC is a superior measure of classifier performance as it provides an unbiased summary of the model's performance across different thresholds. Figure 5 shows a comparison between the three algorithms used for sentiment analysis; Decision Tree achieved the highest validation accuracy of 93.20%. However, the F1 score and the AUROC for the Decision Tree suggest it slightly underperformed compared to logistic regression in terms of balanced performance across precision, recall, and F1 score.



.

# **14. Conclusion and Future Work**

This paper presented an in-depth analysis of Twitter sentiment using machine learning approaches, focusing on the implementation of Random Forest, Decision Tree, and Logistic Regression classifiers. The study highlighted the critical steps involved in sentiment analysis, including data collection, preprocessing, sentiment detection, and model training and testing. Our results indicated that machine learning models can effectively classify sentiments into positive and negative categories, achieving accuracies between 82% and 93%. The Decision Tree model, in particular, demonstrated a remarkable validation accuracy of 93.2%, showcasing its potential for handling complex datasets.

The exploration of various machine learning algorithms revealed the strengths and limitations of each approach. While the Random Forest model provided robust classification capabilities through ensemble learning, Decision Trees offered interpretability and ease of use, making them accessible for stakeholders who may not have a technical background. Logistic Regression served as a baseline model, providing valuable insights into the relationship between features and sentiment classification. By comparing these models, we emphasized the importance of selecting the right algorithm based on the specific requirements of the task at hand, such as accuracy, interpretability, and computational efficiency.

Despite achieving promising results, several challenges remain in the field of Twitter sentiment analysis. The diversity of language used in tweets, including slang, abbreviations, and emoticons, can significantly impact model performance. Current models often struggle with ambiguous phrases and varying contexts, leading to misclassification of sentiments. Additionally, the presence of multiple classes and complex emotions, such as sarcasm, presents further hurdles that need addressing. Our research suggests that future advancements in sentiment analysis should focus on developing models that can adapt to linguistic diversity and better understand nuanced expressions.

Future work will involve expanding the dataset to include a broader range of topics and languages, allowing for a more comprehensive evaluation of model performance across different contexts. We aim to incorporate advanced natural language processing techniques, such as word embeddings and transformers, to enhance feature representation and capture contextual meanings more effectively. Additionally, exploring hybrid models that combine lexicon-based features with machine learning classifiers may improve sentiment classification accuracy, particularly in complex scenarios.

Furthermore, we intend to investigate the application of deep learning methods to capture intricate patterns within textual data. Techniques such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) may prove beneficial in understanding the sequential nature of language and improving sentiment analysis results. Evaluating the performance of cost-sensitive classifiers in unbalanced datasets remains another crucial area for exploration, as it may yield significant improvements in predictive accuracy.

In conclusion, the evolving landscape of social media sentiment analysis presents a bright future for research and application. By addressing the existing challenges and continuously refining our models, we can create robust systems that provide valuable insights into public sentiment, benefiting businesses, organizations, and researchers alike.

# **References**

1. Neethu MS, Rajasree R (2013) Sentiment analysis in twitter using machine learning techniques. In: 2013 Fourth international conference on computing, communications and networking technologies (ICCCNT), Tiruchengode, pp 1–5
2. Kumar A, Sebastian TM, Sentiment analysis on twitter. Department of Computer Engineering, Delhi Technological University Delhi, India
3. Joshi R, Tekchandani R (2016) Comparative analysis of Twitter data using supervised classifiers. In: 2016 International conference on inventive computation technologies (ICICT). ISBN: 978-1-5090-1285-5
4. Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R, Sentiment analysis of twitter. Passonneau Department of Computer Science Columbia University New York, NY 10027 USA
5. Hasan A, Moin S, Karim A, Shamshirband S, Machine learning-based sentiment analysis for twitter accounts. Department of Computer Science, Air University, Multan Campus, Multan 60000
6. Pang B, Lee L (2008) Opinion mining and sentiment analysis
7. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford 1, no 12
8. Trupthi M, Pabboju S, Narasimha G (2017) Sentiment analysis on twitter using streaming API. In: 2017 IEEE 7th international advance computing conference (IACC), Hyderabad, pp 915–919
9. Halibas AS, Shaffi AS, Mohamed MAKV (2018) Application of text classification and clustering of twitter data for business analytics. In: 2018 Majan international conference (MIC), ISBN: 978-1-5386-3761-6
10. Neethu MS, Rajasree R (2013) Sentiment analysis in twitter using machine learning techniques. In: 2013 Fourth international conference on computing, communications and networking technologies (ICCCNT), Tiruchengode, pp 1– 5
11. Shamantha RB, Shetty SM, Rai P (2019) Sentiment analysis using machine learning classifiers: evaluation of performance. In: 2019 IEEE 4th international conference on computer and communication systems (ICCCS), Singapore, pp 21–25
12. Tyagi P, Tripathi RC (2019) A review towards the sentiment analysis techniques.
13. Barbosa, L., & Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. *Proceedings of the 23rd International Conference on Computational Linguistics*, 36-44.
14. Councill, I. G., McDonald, R., & Velikovich, L. (2010). What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. *Proceedings of the ACL 2010 Conference Short Papers*, 51-59.
15. Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter sentiment analysis: The good the bad and the OMG! *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 538-541.
16. Manning, C. D., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.
17. Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, 1320-1326.
18. Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters. In J. Shanahan, Y. Qu, & J. Wiebe (Eds.), *Computing attitude and affect in text: Theory and applications* (pp. 1-10). Springer.
19. Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137.
20. Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill.
21. Tan, C. M., Wang, Y. F., & Lee, C. D. (2002). The use of bigrams to enhance text categorization. *Information Processing & Management*, 38(4), 529-546.
22. Zhang, T. (2003). Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 31(3), 823-859.

sta