Sentiment Analysis on Twitter Data using Machine Learning and Deep Learning

1st Hardik Ghuge
Department of IT,NITK
Surathkal, India
hardikghuge@gmail.com

2nd Dhananjay Kumar
Department of IT,NITK
Surathkal, India
dhananjaykumar1304@gmail.com

3rd Atul Pandey
Department of IT,NITK
Surathkal, India
atulpandey.233it001@nitk.edu.in

I. Introduction

In the big data era, data is made in real-time or closer to real-time. Thus, businesses can utilize this ever growing volume of data for the data-driven or information driven decision-making process to improve their businesses.The era of deriving meaningful insights from social media data has now dawned with advancements in technology. Analyzing and interpreting this diverse data provides valuable insights into the public sentiment surrounding specific topics. The world has experienced a tremendous rise in the volume of textual data especially for the unstructured data generated from people who express opinions through various web and social media platforms for different reasons. Mountains of these textual data, initially could be equated to garbage which would need to be disposed from time to time. However, with the advancement in storage capacity accompanied by the increasing sophistication in data mining tools, opportunities and challenges have been created for analysing and deriving useful insights from these mountains of data

Sentiment analysis is the process used to determine the attitude/opinion/emotion expressed by a person about a particular topic. Sentiment analysis or opinion mining uses natural language processing and text analytics to identify and extract subjective information in source materials. The rise of social media such as blogs and social networks has fuelled interest in sentiment analysis. In order to identify the new opportunities and to manage the reputations, business people usually view the reviews/ ratings/ recommendations and other forms of online opinion. This allows to not only find the words that are indicative of sentiment, but also to find the relationships between words so that both words that modify the sentiment and what the sentiment is about can be accurately identified.

.

Emoticons, often overlooked as a means of communication, have gained significance in sentiment expression. These symbolic representations, including ”, ¯ ’, ”, ”, or (”, commonly convey facial expressions and are read either sideways, such as -(for a sad face, or with a different orientation like (ˆˆ)” for a happy face, depending on the context . Monitoring these emoticons in conjunction with textual content is crucial for accurately discerning sentiments like happiness, frustration, anger, sadness, etc., which can be further classified as positive, negative, or neutral. Social media, like Twitter, generates an enormous amount of such data. However, social media data are often unstructured and difficult to manage. Hence, this study proposes an effective text data preprocessing technique and develop an algorithm to train the Support Vector Machine (SVM),Naïve Bayes (NB) classifiers to process Twitter data. We develop an algorithm that weights the sentiment score in terms of weight of hashtag and cleaned text. In this study, we (i) compare different preprocessing techniques on the data collected from Twitter using various techniques such as (stemming) to obtain the efficient method (ii) develop an algorithm to weight the scores of the hashtag and cleaned text to obtain the sentiment. Using our data preprocessing algorithm and sentiment weight score algorithm, we train SVM, DL, NB models. The results show that stemming technique performed best in terms of computational speed. Additionally, the accuracy of the algorithm was tested against manually sorted sentiments and sentiments produced before text data preprocessing. The result demonstrated that the impact produced by the algorithm was close to the manually annotated sentiments.

While sentiment analysis (SA) research has predominantly focused on either textual or emoticon data, the combined analysis of both has been largely neglected due to resource constraints and the complexity associated with emoticons. Text analysis, particularly using machine learning (ML) algorithms, has been a prominent area of research, extracting sentiments with the aid of various ML and deep learning (DL) techniques facilitated by modern technologies . However, DL applications in SA, especially on the combined text and emoticon data, remain scarce.

This research addresses the gap by conducting separate and combined analyses of text and emoticons to unveil sentiments. Additionally, an emoticon lexicon has been developed, and sentiments were analyzed by integrating emoticon lexicons with various text features, such as TF–IDF, bag-of-words, and n-grams, using both Machine Learning and Deep Learning algorithms.The vecorized documents are eventually provided as input to a SVM based classifier model, which classifies the sentiments expressed in the documents as either positive or negative.

TABLE I: Literature Survey

Paper Title, Author Name, Year	Methodology	Merits	Demerits
K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev and D. Trajanov, "Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers," in IEEE Access,2020.	Authors presents a comprehensive chronological study of sentiment analysis in finance, starting from lexicon-based approaches and progressing through word and sentence encoders to recent NLP transformers	While the approach is constructed for sentiment analysis in the finance domain, the paper suggests its potential extension to other areas such as healthcare, legal, and business analytics	The study focuses on reporting positive results, but it does not extensively explore potential challenges, limitations, or instances where the models may fail
B.Gokulakrishnan, P.Priyanthan, T.Ragavan, N.Prasath and A.Perera, "Opinion mining and sentiment analysis on a Twitter data stream," IEEE,2012	Authors proposed a method where publicized stream of tweets from Twitter microblogging site are preprocessed and classified based on their emotional content and then ,precision and recall are used to perform the analysis of performance in such case.	The study recognizes and addresses the issue of skewness in some datasets by introducing the SMOTE oversampling technique.	While the paper discusses the variations in accuracy between different samples and classifiers, it lacks in-depth comparative metrics for precision, recall, and F1-score
S. Zad, M. Heidari, J. H. Jones and O. Uzuner, "A Survey on Concept-Level Sentiment Analysis Techniques of Textual Data," 2021 IEEE.	Authors propose the techniques of text-based sentiment analysis pipeline including preprocessing, aspect extraction, feature selection, and classification techniques	Paper recognizes the interdisciplinary nature of text mining and sentiment analysis, incorporating concepts from computer science (e.g., linguistics), statistics, and artificial intelligence.	Paper focuses on current sentiment analysis techniques but does not delve into emerging trends or recent advancements in the field. Such as the use of transformer models or transfer learning.
N.Godbole,M.Srinivasaiah and S.Skiena,"Large-Scale Sentiment Analysis For News and Blogs," UVM,2007	Authors proposed a method that provide opinions expressed in newspaper and blogs while reporting on recent events. They assigned a score to each distinct entity in the text corpus indicating positive or negative opinion.	The paper provides statistical evidence of the validity of the sentiment evaluation by correlating the sentiment index with real-world events such as sports outcomes, stock market indices, and seasonal effects.	The paper does not thoroughly address how the sentiment analysis system handles ambiguous terms.
C.Lin and Y.He, "Joint sentiment/topic model for sentiment analysis,"ACM,2009	Authors proposed a model which detect sentiment and topic simultaneously from the text, a probabilistic modeling framework has been based on LDA(latent Dirichlet Allocation), called joint sentiment/topic model(JST).	The preliminary experiments show promising results achieved by the JST model. This suggests that the proposed approach has the potential to contribute positively to sentiment analysis tasks.	The absence of sentiment detection at more detailed levels, such as specific topics within a document, limits the applicability of the model in scenarios where more nuanced sentiment analysis is required.

II. Literature Survey

A. Related Work

In knowledge based approach and machine learning techniques were used for classifcation. It ensemble the applications common sense computing. This is also referred as concept level analysis. In a machine learning techniques have been used for polarity detection and to improve its accuracy. It combines lexical-based and machine learning techniques. It classifes the Facebook messages based on its polarity. In a lexicon based approach has been used for classifcation of twitter data which works on static polarity. In a natural language processing method is used to extract the features from the reviews. A score value is given to all the features which help to analyze the reviews with high accuracy. In a CNN with multiple flters have been used to extract features with varying the window size. It is an unsupervised learning method

In the features have been extracted from the text and then classifed by using supervised learning classifers such as support vector machine. In a knowledge based approach is used to classify the sentiments from the social media. An iterative algorithm was proposed in to predict sentiment polarities in twitter data sets. The algorithm was performed in two stages. In the first stage, sentiment reversal was done to analyze the tweets and retweets. The tweets were constructed into a tree by splitting into tweet (parent node) and retweet (child node). Both will have diferent polarities i, positive, negative and neutral.

In the second stage, the relationship between the difusion and reversals was done to extract the patterns. Overall performance has been improved by respectable amount compared to other methods. Multivariate vehicle regression models were applied on the values of stock market and social media to predict monthly sales of the vehicles in . Three kinds of datasets were taken as inputs to analyze and predict the sales. They are values of stock market, scores of sentiments and hybrid model. By observation, it is noted that by applying regression models on three datasets, hybrid model is giving more accurate results compared to other datasets.

In seven machine learning techniques were applied to classify the airline dataset into three sentiment classes i.e., positive, negative and neutral. The input data is preprocessed and the tweets were represented as vectors to perform deep learning concepts. The dataset was divided into 80% of train data and 20% as test data. The main drawback is the limited numbers of tweets were considered. It would be good to get high accuracy if the number of tweets were increased.

B. Outcome Of Literature Survey

The literature survey reveals diverse approaches to sentiment analysis in different domains. In the finance sector,

Mishev et al. present a comprehensive study, emphasizing the potential extension of their approach to healthcare, legal, and business analytics. Gokulakrishnan et al. address sentiment analysis on Twitter data, introducing the SMOTE oversampling technique to handle skewed datasets. Zad et al.'s survey on concept-level sentiment analysis techniques covers preprocessing, aspect extraction, and classification methods but falls short in exploring recent trends like transformer models. Godbole et al.'s large-scale sentiment analysis for news and blogs correlates sentiment indices with real-world events but lacks comprehensive handling of ambiguous terms. Lin and He propose a joint sentiment/topic model, showing promise in sentiment analysis tasks but with limitations in detailed sentiment detection. Overall, the literature emphasizes diverse methodologies, acknowledges challenges, and suggests avenues for improvement in sentiment analysis.

C. Problem Statement

To address the need for robust sentiment analysis, the objective is to develop a method capable of accurately and efficiently classifying text data into distinct sentiment categories, specifically positive and negative. This endeavor is motivated by the growing importance of understanding user sentiments expressed in textual content, ranging from customer reviews to social media posts. The aim is to provide a reliable mechanism that automates the categorization of textual data, enabling businesses, researchers, and practitioners to gain valuable insights into public opinions.

D. Objectives

1. To enhance data quality, preprocess collected tweets by removing noisy symbols (e.g., emoticons, hashtags), resulting in cleaner, more consistent data.
2. To extract features and opinions, machine learning algorithms were applied, utilizing optimization techniques to improve performance.
3. To know whether the user has negative, positive and neutral opinion towards a product or something else.

III. Dataset

There is a publicly available dataset from Twitter, which we've had referred and used in our ML and DL models to train.

IV. Proposed Methodology

As the Data which we are using to train our models is not balanced means it is having many noise which will affect the performance of our model. It also contains unused or data which won't affect in predicting the user's mental health. Inorder, to make our data clean we are following certain steps which will help in correct prediction of user's health.

A. Data Preprocessing

This is first and foremost step while detecting depression on textual data. As the dataset which we referred is having so many liabilities which is not useful in prediction but it might effect the performance of ML and DL models. Whole data is converted to lowercase.

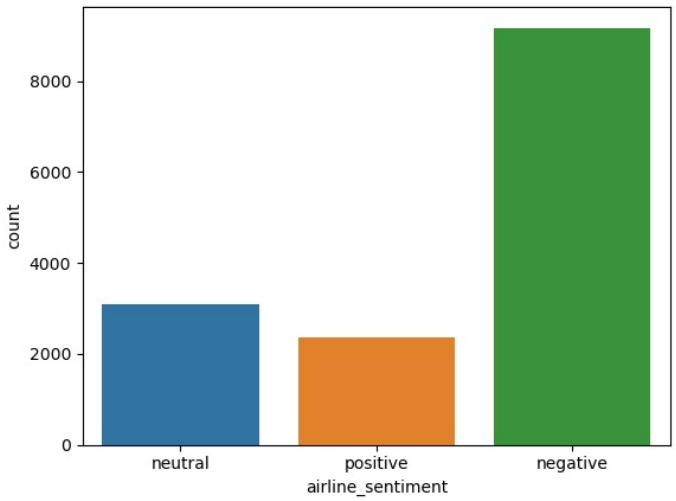

Fig. 1: Airline Sentiment

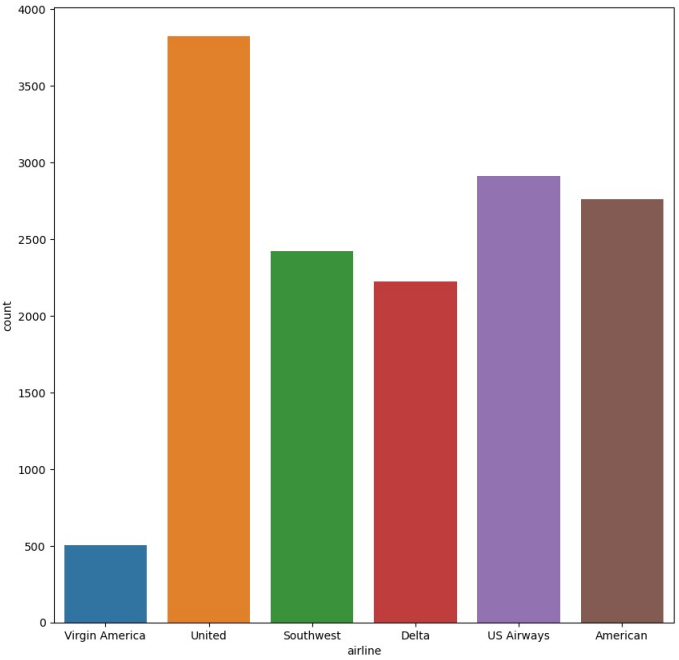

Fig. 2: Airlines

1) Removing html tags : HTML tags such as this <.*?> won't help us in predicting the mental health of user which is having symptoms of depression

2) Removing URL's : URL links which are used in comments or used while chatting are of no use in above text classification which is helping in predicting the user's health

3) Removing Punctuation Marks : Punctuation marks such as (.,:?!””-[,]) are nothing but useless in this context of the data

4) Expanding chat acronyms : Nowadays, while having conversation with friends or posting thoughts on social media most of the user's use short-forms so to convey their msg in easy ways and don't want the burden of writing the whole

words. Some acronyms like(lol means "Laugh out Loud",asap means "As soon as possible") which might create confusion for our model in prediction. So we've replaced this acronyms with their actual meaning while preprocessing

5) Removing Stop Words: While preprocessing the removal of stop words is also necessary.Some Stop words are (the,a,me,i) that occur frequently in conversation which carries little semantic meaning. By removing stop words, we can focus on the words that contribute more to overall meaning of a text.

6) Treatment of Emoji's : There are two approaches we can use in this case where we can remove the emojis or we can replace them with their original meaning.But replacing the emoji's with their actual meaning will be quiet helpful in training our models. When we demojize it gives the added advantage in correct predictions.

B. Tokenization

It is one of the crucial step while performing text classification and in our case depression detection.Tokenization breaks down a text into individual words or tokens. This provides a more granular representation of the text, allowing the model to understand the meaning of the text at the word level. Each token becomes a feature that the model can use for classification.Tokenization standardizes the representation of text, making it easier to compare and analyze documents. It ensures that each piece of text is broken down into the same basic units, which contributes in consistent and reliable analysis.

C. Stemming

Here we've two apprach to convert words to their root form, one is Stemming and other is lemmatization. Depending on requirement we use either of these methods, here we are going with stemming due to it's performance for large datasets compare to lemmatization. Stemming is the process of reducing inflection in words to their root forms such as mapping a group of of words to the same stem even if the stem itself is not a valid word in the language. It is used when we need speed or want our job to be done quickly as possible.Advantage of doing this step before feature selection is the the words which are having same meaning are mapped to their root word and not treated as separate. If it is treated as different words even though context is same, it could lead to inaccurate representations and analysis.Here we are using Porter stemmer algorithm which was invented in 1980 by Martin F.Porter.

D. Feature Extraction

For Extraction of Features from text data set we are using TF-IDF vectorizer. TF-IDF(Term Frequency - Inverse Document Frequency)

After tokenizing TF is calculated which measures how often the tem is used in a specific document

IDF is calculated for each term in entire corpus.This measure helps to asses the importance of term in the context of the entire collection of the documents.Terms that are common across many documents receive lower IDF score,while terms that are more unique to specific documents receive higher IDF score.

- $\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$
- $\text{IDF}(t, D) = \log\left(\frac{\text{Total number of documents in the corpus } N}{\text{Number of documents containing term } t}\right)$
- $\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$

- t is the term (word) for which we are calculating the IDF.
- D is the collection of documents.
- N is the total number of documents in the corpus.
- $df(t, D)$ is the document frequency of the term t, i.e., the number of documents in which the term t occurs.

TF-IDF score is calculated by multiplying TF and IDF score. The process results in numerical representaion of importance of each term in document. Higher TF-IDF values indicates greater importance of a term in a specific document.

The TF-IDF scores for all terms form a feature vector for each document. These feature vectors can then be used as input for machine learning algorithms in text classification tasks such as Depression Detection. Each dimension in the vector corresponds to a unique term in the vocabulary, and the TF-IDF score represents the importance of that term in the document.

E. Training on various Machine Learning models

1) Naive Bayes Classifier: The Naive Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification. It is also part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category.

For each class, calculate the prior probability, which is the probability of a document belonging to that class without considering the features. $P(C_i)$ represents the prior probability of class C_i

For each term (feature) in the vocabulary and for each class, calculate the conditional probability of observing that term given the class label. This involves calculating $P(T_j|C_i)$, the probability of term T_j given class C_i.

2) Decision Tree: Decision Trees are a popular and intuitive machine learning algorithm used classification. They work by recursively partitioning the data into subsets based on the most significant attribute at each step. The result is a tree-like structure where each internal node represents a decision based on a particular feature.

The algorithm evaluates different features and selects the one that best splits the data into homogeneous subsets based on the target variable. The chosen feature is used to split the data into subsets at each internal node, and the process is repeated for each subset. This recursive partitioning continues until a stopping criterion is met. To make a prediction for a new data point, it traverses the tree from the root to a leaf node based on the feature values of the data point. Without proper control,

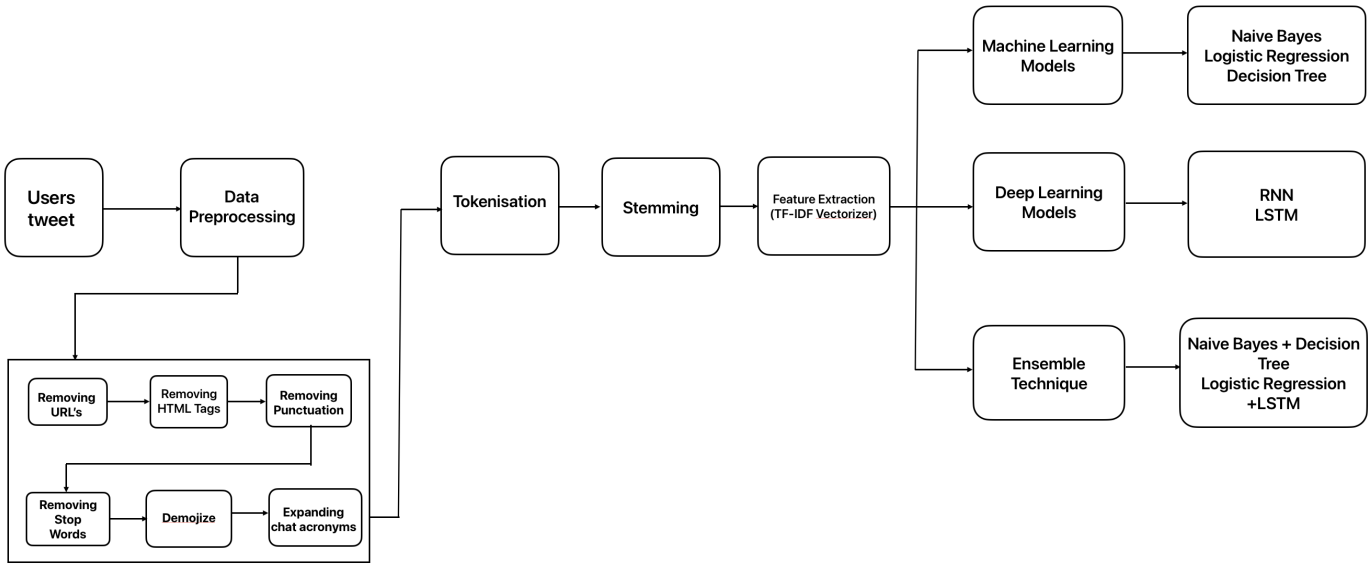

Fig. 3: Proposed Architecture

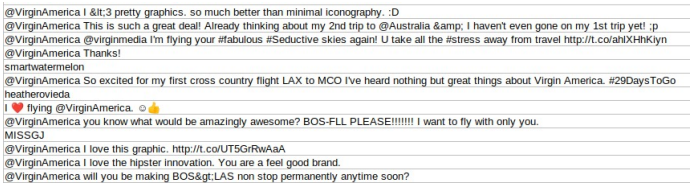
@VirginAmerica I <3 pretty graphics. so much better than minimal iconography. :D
@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! :p
@VirginAmerica @virginmedia I'm flying your #fabulous #Seductive skies again! U take all the #stress away from travel http://t.co/ahlXHhKiyn
@VirginAmerica Thanks!
smartwatermelon
@VirginAmerica So excited for my first cross country flight LAX to MCO I've heard nothing but great things about Virgin America. #29DaysToGo
heatherovieda
I ❤️ flying @VirginAmerica. ☺👍
@VirginAmerica you know what would be amazingly awesome? BOS-FLL PLEASE!!!!!!! I want to fly with only you.
MISSGJ
@VirginAmerica I love this graphic. http://t.co/UT5GrRwAaA
@VirginAmerica I love the hipster innovation. You are a feel good brand.
@VirginAmerica will you be making BOS>LAS non stop permanently anytime soon?

Fig. 4: Before Preprocessing

```
i lt pretti graphic so much better than minim ...
thi is such a great deal alreadi think about m...
virginmedia im fli your fabul seduct sky again...
                                          thank
                        sfopdx schedul is still mia
so excit for my first cross countri flight lax...
i flew from nyc to sfo last week and couldnt f...
                                   fli virginamerica
you know what would be amazingli awesom bosfll...
whi are your first fare in may over three time...
                                  i love thi graphic
i love the hipster innov you are a feel good b...
will you be make bosgtla non stop perman anyti...
```

Fig. 5: After Preprocessing

Decision Trees are prone to overfitting, capturing noise in the training data.

3) Logistic Regression: Logistic Regression is a statistical method used for binary classification problems, where the outcome variable is categorical and has two classes (usually denoted as 0 and 1). Despite its name, logistic regression is a classification algorithm, not a regression algorithm. It estimates the probability that a given input belongs to a particular category. The logistic regression model is based on the logistic function (also known as the sigmoid function). The logistic function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

where: - $\sigma(z)$ is the logistic (sigmoid) function. - e is the base of the natural logarithm. - z is a linear combination of the input features.

In logistic regression, the linear combination z is defined as:

$$z = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n$$

where: - b_0 is the intercept. - $b_1, b_2, \ldots, b_n$ are the coefficients associated with each feature $x_1, x_2, \ldots, x_n$.

The logistic regression equation can be written as:

$$P(Y = 1) = \sigma(b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n)$$

Here, $P(Y = 1)$ represents the probability of the dependent variable (Y) being equal to 1.

F. Performance Measure

The performance measurement of datasets is carried out with the help of a confusion matrix from where we get the results, which calculates the values of accuracy, precision, and recall with the help of positive and negative values of datasets. The formulas used for the calculation of values are,

True Positive (TP): Number of correctly predicted positive instances.

True Negative (TN): Number of correctly predicted negative instances.

False Positive (FP): Number of incorrectly predicted positive instances (Type I error).

False Negative (FN): Number of incorrectly predicted negative instances (Type II error).

1) Accuracy: The ratio of correctly predicted instances to the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2) Precision: The ratio of correctly predicted positive instances to the total predicted positive instances.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3) Recall: The ratio of correctly predicted positive instances to the total actual positive instances.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) $F1_Score$: The harmonic mean of precision and recall, providing a balance between the two.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

G. Ensemble Techniques

Ensemble techniques in text classification involve combining the predictions of multiple base classifiers to improve overall performance. These methods aim to trigger the strengths of different models and mitigate their individual weaknesses. Here are some popular ensemble techniques used in text classification.

1) Bootstrapping: Train multiple instances of the same base classifier on different subsets of the training data (bootstrap samples) and combine their predictions through averaging or voting.

2) Voting Classifier: Combine the predictions of multiple base classifiers through voting (e.g., majority or weighted voting).

Hard Voting : Where the classification is based on majority number models giving same output.

Soft Voting : Where the classifiers provide probability estimates for each class of outcome

H. Deep Learning Models

1) Recurrent Neural Network(RNN): Recurrent Neural Networks (RNNs) are a type of neural network architecture commonly used for sequential data, making them suitable for text classification tasks. In text classification, the goal is to assign predefined categories or labels to text documents.

2) Long Short Term Memory(LSTM): LSTMs are a type of recurrent neural network (RNN) designed to address the vanishing gradient problem in traditional RNNs. They are well-suited for handling sequences of data, making them effective for tasks like text classification.

LSTMs have a unique architecture that includes a memory cell. This memory cell allows the network to capture and remember information over long sequences, making it particularly useful for understanding the context of words in text.

I. Results

The results suggest that deep learning models, specifically LSTM, outperform traditional machine learning models such as Naive Bayes and Decision Tree in terms of accuracy, precision, F1 score, and recall. Logistic Regression also demonstrates high performance, indicating its effectiveness in text classification tasks.

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Naive Baye's	61.93	71.10	61.93	65.91
Logistic Reg.	88.43	88.34	88.43	86.99
SVM	89.30	88.87	89.30	88.62
SVM + DT	88.45	88.43	85.75	86.98
DT	83.54	81.68	83.54	80.57
RNN	89.90	88.29	89.28	89.27
LSTM	89.69	88.82	88.78	88.75

TABLE II: Result Analysis

V. Conclusion

The presented results showcase the performance of various sentiment analysis models, including Naive Bayes, Logistic Regression, Support Vector Machine (SVM), Decision Trees (DT), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM). The evaluation metrics such as accuracy, precision, recall, and F1 score provide valuable insights into the efficacy of each model.

The findings indicate that RNN,SVM and LSTM exhibit superior performance, outperforming other models in terms of accuracy. These results suggest that these models are well-suited for sentiment analysis tasks, showcasing their reliability in extracting sentiments from diverse textual data sources

References

[1] A. Naresh, P. Venkata Krishna, An efficient approach for sentiment analysis using machine learning algorithm , Published online: 3 June 2020,Evolutionary Intelligence (2021) 14:725–731

[2] C. N. dos Santos eta M. Gatti, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts" in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 69–78.

[3] P. Ray eta A. Chakrabarti, A Mixed approach of Deep Learning method and Rule-Based method to improve Aspect Level Sentiment Analysis, Appl. Comput. Informatics, 2019.

[4] S. Lai, L. Xu, K. Liu, eta J. Zhao, Recurrent Convolutional Neural Networks for Text Classification, Twenty-ninth AAAI Conf. Artif. Intell., pp. 2267–2273, 2015.

[5] D. Tang, B. Qin, eta T. Liu, Document Modeling with Gated Recurrent Neural Network for Sentiment Classification, in Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 1422–1432.

[6] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, eta Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, arXiv Prepr. arXiv1406.1078, 2014