National College of Ireland

A Comparative Study of Machine Learning and Lexicon-Based Methods for Sentiment Classification

MSc Research Project
MSc Data Analytics

Ammaiappan Srinivasan
Student ID: 23270900

School of Computing
National College of Ireland

Supervisor: Dr. Bharat Agarwal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ammaiappan Srinivasan
Student ID:	23270900
Programme:	MSc Data Analytics
Year:	2026
Module:	MSc Research Project
Supervisor:	Dr. Bharat Agarwal
Submission Due Date:	28/01/2026
Project Title:	A Comparative Study of Machine Learning and Lexicon-Based Methods for Sentiment Classification on Twitter Data
Word Count:	8157
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Ammaiappan Srinivasan
Date:	28/01/2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator office must be placed

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

into the assignment box located outside the office.

A COMPARATIVE STUDY OF MACHINE LEARNING AND LEXICON- BASED METHODS FOR SENTIMENT CLASSIFICATION ON TWITTER DATA

Ammaiappan Srinivasan
23270900
MSc. Data Analytics
School of Computing
National College of Ireland

Abstract

This study is a comparison between machine learning and lexicon-based sentiment classification on Twitter data using Sentiment140 dataset of 1.6 million tweets. The paper has constructed an entire sentiment analysis pipeline including preprocessing, feature extraction, training of the model and evaluation. TF-IDF vectorization was done to clean, tokenize and lemmatize the text data which was then classified using four supervised models which included Naive Bayes, Logistic Regression, Support Vector Machine (SVM) and random forest. Accuracy, precision, recall and F1-score were used to measure the performance. Findings indicated that both Logistic Regression and Linear SVM scored the highest accuracy and F1-scores of about 78 and better than lexicon-based methods on contextual adaptability and predictive reliability. Even though lexicon-based methods were interpretable and cheap to compute, they could not handle informal and ambiguous language. The paper finds that linear machine learning fits offer intermediate between scalability and accuracy and that the next research should adopt hybrid or transformer-based systems to capture the contextual sentiment in greater detail.

Keywords: Sentiment Analysis, Twitter Data, Machine Learning, Lexicon-Based Methods, Natural Language Processing, TF-IDF, Logistic Regression, Support Vector Machine, Text Classification, Opinion Mining

1. Introduction

1.1. Overview

The widespread use of social media has redefined communication, information transfer and the expression of opinions worldwide. Twitter is a dynamic sentiment-analysis platform that provides real-time sentiment analysis and is known for being concise and instantaneous. Sentiment analysis or opinion mining is aimed at locating attitudes, feelings and opinions in text data. Twitter sentiment categorization has obtained significant academic and business attention due to its potential applications in social health, politics and marketing. However, the informal words, acronyms, emoticons and situational nuances make it hard to extract emotions from tweets. The two principal approaches to handling such problems have been lexicon-based methods and machine learning. Some machine learning models used to identify sentiment patterns include Naive Bayes, Support Vector Machine and Logistic Regression, which operate with annotated data [1]. Deep learning models such as LSTM and BERT are further enhanced by understanding sequential correlations and semantic representations, which raise their performance. The methods based on lexicon are effective and easy to use, but they cannot

effectively tackle the context, sarcasm and informal expressions, which makes them significant in terms of their comparison with machine learning and deep learning models.

1.2. Aims

The research is primarily aimed at comparing a lexicon, machine learning and deep learning sentiment classification on Twitter data. The researchers examine the impact of varying levels of preprocessing on the efficiency, interpretability and performance of the model. It also seeks flaws in methodological frameworks and provides recommendations to improve the accuracy and applicability of sentiment analysis to natural language processing.

1.3. Objectives

RO 1: To compare and contrast the computational efficiency, scalability and accuracy of lexicon-based and machine learning methods.

RO 2: To gauge the effect of simple, moderate and sophisticated preprocessing on the determination of sentiment categorization results.

RO 3: To determine the interpretability of lexicon-based and machine learning models and their ability to adapt to the context.

RO 4: To indicate existing gaps in the comparative evaluation of sentiment analysis techniques and to indicate future research directions.

RO 5: To provide valuable recommendations on how to simplify the sentiment analysis procedures on huge Twitter databases.

RO 6: To evaluate the performance of deep learning models, including LSTM and BERT and compare them with other methods based on lexicon and machine learning.

1.4. Research Question

RQ 1: What are the differences in the computing capacities and the accuracy of lexicon-based, machine learning and deep learning algorithms using Twitter data?

RQ 2: How do the various preprocessing techniques affect the efficacy of sentiment classification models?

RQ 3: What is the most appropriate sentiment analysis approach with a reasonable balance between accuracy, scalability and interpretability on datasets?

1.5. Research Problem

The increased reliance on social media analytics demands powerful and efficient frameworks of sentiment categorization. Twitter is informal and terse, which is a critical problem for traditional natural language processing solutions. Even so, much remains debated regarding the comparative merits of lexicon-based methods and machine learning methods in spite of a vast amount of research. Even though machine learning models are highly accurate in prediction, the model often requires significant quantities of labelled data and computing resources. Although interpretable and resource-efficient, lexicon-based methods are less effective in more complex language structures and lack the ability to be more flexible at the contextual level [2]. Besides, similar studies have failed to adequately focus on the effect of preprocessing techniques on the performance of models. The sentiment analysis of Twitter is precise, scalable and understandable when the preprocessing discrepancies are managed via one framework. Deep learning technologies, though being very precise, also present difficulties associated with the cost of computation and the lack of interpretability that should be taken into account when comparing these approaches.

2. Related Work

Sentiment analysis is an important domain in artificial intelligence and natural language processing. The prevalence of user-created content on social media, particularly Twitter, has fuelled academic research interest in understanding the emotions and thoughts of the collective public. As a means of advancing analytical precision, researchers have investigated an array of computational approaches, including lexicon-based and machine-learning methods. The literature indicates substantial momentum in new algorithms and the increasingly wide and growing application of sentiment analysis. We will examine key research that advances methodologically and comparatively the landscape of sentiment analysis using Twitter.

2.1. The Progression of Sentiment Analysis and Its Applications with Twitter Data

Sentiment analysis has developed in response to a growing interest in understanding opinions expressed in a variety of internet contexts. According to Bhakuni et al. (2022), sentiment analysis is an important part of artificial intelligence and natural language processing, which has evolved from traditional opinion mining to more complex, computer-based models that can detect subtle feelings such as sarcasm in tweets. Their research examined how sarcasm and ambiguity in tweets present challenges to the appropriate classification of sentiment, which adds complexity in preprocessing, feature selection and classifier choice [3]. Wang et al. (2022), in a related study on Twitter sentiment analysis, describe how Twitter's crowded data set of user-generated content, as well as its short sentences and phrases, are IDs type in accurately measuring public opinion. They note that Twitter's briefness presents unique challenges that often include slang, abbreviations, relativity and contextual understanding [4]. From their analysis and methodology, it is clear that early sentiment analysis of social media messages targeted longer messages. Collectively, these studies highlight how the influence of improved methods of natural language processing (NLP) and machine learning for social media has transitioned sentiment analysis from simple polarity analysis to a complex and domain-centric model of emotion. Continuing from the developments described above, sentiment analysis has become an evolving and interdisciplinary method utilized across various domains such as marketing, finance and health [5].

In addition, this paper sheds light on advances in artificial intelligence and natural language processing that have driven the rapid growth of sentiment analysis research since 2008. It noted that despite the great success of traditional machine learning techniques (both hand-crafted models and lexical ones), now more attention is also being paid to hybrid models and deep learning ones. The authors also highlighted that the time and cause elements of sentiment analysis increased understanding of user emotions in several domains where it applies -i.e., politics and financial markets. They pointed out that though academic work is burgeoning, patent generation by companies is significant, demonstrating the industrial applicability of emotion analysis. The study concluded that for facilitating the reconciliation of academic and commercial progress, it is important to have a more appropriate domain-adaptation and transfer assessment framework. Finally, the authors predict continuous momentum in research with sentiment analysis as a significant building block of social media analytics for the next decade [5].

2.2. Methods for Classification of Sentiment on Twitter with Machine Learning

Machine learning models' adaptability and accuracy have made them indispensable for performing sentiment classification on Twitter. According to Qi and Shabrina (2023), classifiers such as Random Forest, Support Vector Machine and Naïve Bayes use structured feature vectors to classify tweets in relation to sentiment polarity efficiently. They emphasized that semantic awareness of Twitter is considerably enhanced through other feature representations such as Bag of Words, TF-IDF and Word2Vec. For example, preprocessing

techniques such as tokenization, stemming and part-of-speech tagging provide better classification efficiency [6]. This research has shown that when supervised classifiers are integrated with feature representation over large and noisy datasets, the sentiment prediction is improved.

In their study, AlBadani et al. (2022) introduced an advanced hybrid paradigm for sentiment analysis that combines Support Vector Machine and Universal Language Model Fine-Tuning (ULMFiT) for efficient sentiment detection. Their proposed approach was able to generalize better and had lower training complexity, attaining an accuracy of 99.78% on the Twitter US Airlines dataset [7]. Aljedaani et al. (2022) used CNN, LSTM and GRU, among several different architectures and reported that LSTM-GRU had a 0.97 accuracy and 0.96 F1 score. Their study indicated that performance and context awareness are improved when deep models and transfer learning are utilized [8].

2.3. Comparative Evaluations with Machine Learning Models and Lexicon-based Methods

Lexicon sentiment analysis techniques employ already established dictionaries to evaluate the polarity of a word. Lexicon-based approaches are interpretable, efficient with resources and are effective in multilingual or low-resource settings. Srivastava et al. (2022) applied reviews from hotels to evaluate machine learning models with lexicon techniques, including VADER and AFINN. VADER's accuracy of 88.7% was lower than SVM's 96.3% accuracy, suggesting that there are differences in performance related to methods [9]. They found that lexicon models are faster to compute but less contextually sensitive than supervised classifiers. These are comparisons showing how crucial the quality of feature representation and data pre-treatment is to the result of performance.

Catelli et al. (2022) also compared LexTs to lexicographers and against a neural language model based on BERT in similar experimental settings for NooJ-based lexicons using Italian textual data. They found that on small and/or domain-specific sets, lexicon-based models were superior, but BERT achieved higher accuracy for complex usage of the language [10]. They showed that the rule-based stability of lexicon approaches makes them applicable in less-resourced languages. Besides, the work also proved the easy interpretability of lexicons for polarity identification by explaining them with SHAP analysis. These results also substantiate the use of lexicon models in narrow-coverage or contextually less-prominent scenarios.

Hybrid strategies have recently been analyzed to augment this comparative study. Van der Veen and Bleich (2025) introduced MultiLexScaled to achieve a fine-grained sentiment scoring, which surpassed ML models in generalization capabilities [11]. Similarly, Raees and Fazilat (2024) confirmed neutrality detection with TextBlob and VADER over 1.6 million tweets, comparing the results against Random Forests and SVM. Viewed as a whole, these results demonstrate that corpus-based models are still required for accurate, domain-independent sentiment analysis and especially so in cases where computational or annotation resources are scarce [12].

3. Methodology

3.1. Introduction and Background

Sentiment analysis (also referred to as opinion mining) is a vital sub-discipline of Natural Language Processing (NLP) which is concerned with the recognition and classification of the opinions within a text. As the social media platforms, like Twitter, explode in their growth, the users exchange their opinions on a variety of topics and mass data is created out of unstructured information. This gives researchers a chance to elicit information upon public opinion and

emotional colouring based on computational means. This study focuses on Twitter data analysis and machine learning techniques which used to categorize tweets based on positive or negative sentiments. This paper has shown by using the Sentiment140 dataset how conventional NLP and supervised learning algorithms can be used successfully to estimate the polarity of sentiment in short informal text such as tweets.

The project aims at developing an end-to-end sentiment analysis system, including data preprocessing, exploratory analysis and feature extraction, to model building, evaluation and prediction. The methodological approach combines both the qualitative and quantitative analysis to produce sound and interpretable results. Every step of the workflow is developed to solve a problem in the real world like noisy data, informal language and uneven distribution of sentiment typical of social media data.

3.2. Dataset Description

The dataset in this project is the Sentiment140 dataset which is publicly available on Kaggle. It includes 1.6 million tweets with every one of them being labeled as either positive (4), neutral (2) or negative (0) (Dataset Link: https://www.kaggle.com/datasets/kazanova/sentiment140). The data are in six columns namely target, ids, date, flag, user and text. In this research, the target and text field were used as these are the two most relevant fields as far as sentiment classification is concerned. The target column was reformatted to categorical labels, Negative, Neutral and Positive which could be easily interpreted.

To ease analysis and eliminate imbalance in classes, the neutral tweets were eliminated, hence leaving only a binary classification problem, namely positive and negative sentiments. Following such filtering, the dataset still had a large number of samples (more than 1 million tweets) and this was sufficient to train the model. The data is made of real Twitter posts thus giving a natural noise of spelling errors, abbreviations, hashtags, mentions and URLs. Hence, giving a realistic scenario in text mining research.

3.3. Research Approach

The strategy was based on supervised machine learning with various consecutive steps: data purification, text processing, feature identification, training and testing of a model. Python was used in carrying out this project because of its strong support of NLP and data analysis with libraries that include NLTK, Pandas, Scikit-learn as well as Matplotlib.

Data Cleaning and Preprocessing: The raw tweets were subjected to a lot of preprocessing in order to make the text clean and consistent. This included several procedures such as text to lowercase conversion, URL elimination, references to other users (i.e. @users), hashtags and numbers, punctuation marks and additional white space. Following such cleaning, tokenization was carried out with NLTK word_tokenize and lemmatization was conducted with WordNetLemmatizer in order to reduce the number of words to their base form. English language stopwords were eliminated to get only meaningful tokens. Another feature that had been developed was a text_length that calculated the character length and word_count of each tweet, which offers an understanding of the complexity of text.

This was a necessary step since tweets usually include informal language, emoticons and abbreviations and they have the potential to bring noise to the model and influence its performance. Cleaning and normalization procedures were to minimize this variation to increase the uniformity of input text data to extract features and train models.

3.4. Feature Extraction

After the preprocessing stage, the second task was to change text data into numerical features that can be used in machine learning models. TF-IDF (Term Frequency-Inverse Document Frequency) was used as a technique to represent each of the tweets in the form of a numeric

vector reflecting the significance of words in comparison to the overall corpus. The TF-IDF vectorizer was set to use the top 5000 features and bi-grams (1,2) to retrieve the contextual relationship between words.

TF-IDF was chosen instead of a simpler method such as Count Vectorization because it minimizes the effect of words that occur frequently but provide less information, which increases discrimination of sentiments. The method is very popular in the context of text classification tasks and it guarantees that the models are able to get useful trends in the use of words which are characteristic of either positive or negative emotions.

TF-IDF feature matrix obtained was used to train models. The dataset was then categorized into training (80%) and testing (20%) subsets with train_test_split() with a stratification that ensured representation of balanced classes.

3.5. Experimental Setup

The purpose of the experimental design was to compare the performance of several machine learning models using the same dataset and characteristics representation. Four supervised learning algorithms were used:

- **Multinomial Naive Bayes (MNB) -** this method is useful when dealing with text classification because it assumes words are independent and it is effective with high dimensional sparse data.
- **Logistic Regression (LR) -** a powerful linear model which is used to guess binary data in a probabilistic manner.
- **Linear Support Vector Machine (SVM) -** useful when it is possible to linearly separate text data, which has high accuracy and generalization.
- **Random Forest (RF) -** an ensemble and predictive model in which multiple decision trees are utilized to enhance predictive accuracy and decrease overfitting.

All the models were trained on the TF-IDF-transformed training data. Hyperparameters were also selected wisely: Logistic Regression and SVM operated with a maximum 1000 iterations, whereas Random Forest operated with 50 estimators with depth and feature control to ensure the maximum speed and accuracy.

Evaluation of the models was done through the standard metrics such as accuracy, precision, recall and F1-score, which are computed based on test set predictions. Other measurement instruments such as confusion matrices, classification reports and ROC curves were employed to graphically represent performance and model behaviour. These classical models were accompanied by the lexicon-based tools (VADER, TextBlob, SentiWordNet) and deep learning models (LSTM and BERT) to allow a full comparison.

3.6. Research Workflow

The entire research investigative process was systematic:

- **Data Collection:** Importing Sentiment140 data on Kaggle.
- **Preprocessing:** Tweets text cleaning, tokenizing and lemmatizing.
- **Feature Engineering:** TF-IDF vectors conversion of text.
- **Model Training:** Using a combination of machine learning classifiers.
- **Evaluation:** Comparison of models based on the accuracy, F1-scores and ROC-AUC measures.
- **Hypothesis:** To check the applicability of the results in real life, testing using new and unfamiliar tweet examples is necessary.

The methodology was in such a way that it was reproducible, scalable and interpretable. The analysis provided also included data distribution, most used words and sentiment pattern by using Matplotlib and Seaborn to visualize the data.

3.7. Summary

Overall, the methodology implemented in the study comprises a full and reproducible pipeline for carrying out sentiment analysis on data obtained from Twitter. Each step including the dataset, the preprocessing method, the feature extraction method and the choice of the model was grounded in both theoretical and empirical findings of NLP research. The variety of models used gives a comparative insight into the different ways machine learning algorithms can be applied for text classification tasks. Such a detailed and thorough procedure is a guarantee of not only high accuracy of the models in question but also their interpretability and transferability to different domains. The later parts of this report account for the design and the implementation details, followed by the evaluation and the discussion of results.

4. Design Specification

4.1. System Overview

The sentiment analysis system is built as an entire pipeline which takes in raw Twitter text and predicts whether the sentiment being expressed is positive or negative. The pipeline incorporates a few steps such as data preparation, text cleaning, preprocessing, feature mining, model training, evaluation and prediction. It aims to transform raw tweets into ordered numerical information that can be successfully utilized by machine learning algorithms to train on patterns of sentiment. This system is executed in Python due to its high data science and natural language processing capabilities. Libraries like Pandas, Numpy, NLTK, Scikit-learn, Seaborn and Matplot library are extensively used in text processing, model development and graphics.

The general design is such that it is modular where every individual component is expected to carry out a specific role and the output is expected to be fed into the next step. This renders the process efficient, debug friendly and reproducible. The implementation aims at creating an interpretable and modular and extendable solution, which can process large amounts of textual information in social media and also be accurate and computationally efficient.

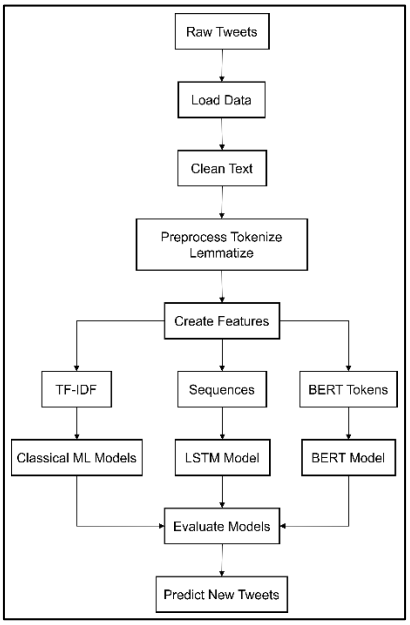

Fig 1: System Architecture

4.2. Dataset Design and Preparation

The Sentiment140 data is chosen due to the number of labeled tweets, which is 1.6 million and it was obtained through real users. Polarity value of 0, 2 or 4 in each tweet, which signifies negative, neutral and positive sentiment respectively. The data set contains target, id, date, flag, user and text. The other fields are not to be analysed since they are not directly related to the purpose of defining the sentiment. The information is imported to a Pandas DataFrame using the Latin-1 encoding to support special characters that are frequent in tweets.

To narrow down the model to a set of definite positive and negative sentiments, neutral twitters are eliminated. The other data is divided into two classes of sentiment. The new binary column, which is named sentiment, is created with 0 or 1 as the negative or positive sentiment respectively. This simplifies the work and makes it possible to use binary classification algorithms. Other derived data like length of the text and the number of words are added so as to know the complexity levels and the density of the tweets and this can subsequently be used to facilitate the exploratory analysis.

4.3. Data Cleaning and Text Normalization

Any Twitter text is usually full of noise, with hyperlinks, hashtags, mentions and numbers, as well as punctuation. This noise should be cleaned to make the models acquire meaningful patterns as opposed to meaningless tokens. The cleaning process commences with all the texts being converted to lowercase so that there is uniformity. Then, regular expressions are used to delete the URLs since they will not provide any semantic information. Mentions and hashtags are removed because they mostly are usernames or topics but not sentiment. It also removes numbers and punctuations in order to subject the analysis to pure textual content.

Subsequent cleaning is performed to remove redundant whitespaces into single spaces. The text is cleaned making it smoother and easier to read, but now it is ready to be tokenized and lemmatized. This is essential in reducing the diversity of vocabulary due to variations of the word of the same meaning. E.g. "running", "runs", "ran" etc. are all translated to "run". The wordnet lemmatizer of the NTLK library is used to perform lemmatization. Stopwords like "is", "the" and "of" are eliminated because they are common words, yet they do not say much about sentiment.

The text of the tweet is cleaned and normalized and each processed version of the text is stored in the DataFrame. This will guarantee the availability of both the raw and processed ones to compare or re-use. The cleaning and normalization step converts very informal, noisy tweets into the text that can be considered as standardized and can be utilized in the process of the subsequent vectorization and modeling.

4.4. Feature Engineering and Text Representation

Once the tweets have been preprocessed, they must be converted into numerical formats, which can be interpreted by machine learning models. This is done through the use of the Term Frequency-Inverse Document Frequency (TF-IDF) method. TF-IDF places more weight on words that are common in each tweet and uncommon in the corpus and this technique enables the model to put attention on unique words.

The vectorizer is set to sample single words and pairs of consecutive words (bigrams) and the maximum number of features is 5000. This balance ensures that the context is well represented in terms of richness without being over-dimensional. The process of vectorization converts every tweet into a sparse matrix in which each item is the significance of a word in that specific tweet as compared to the dataset. This is the basic set of core features that are used to train the model.

This dataset is further split into training and testing subsets in a 80:20 proportion. Stratification also makes sure that positive and negative samples are equally distributed in the two subsets. This would avoid bias in learning models and would make evaluation measures reflect the generalization performance. The resulting split data is then represented as vectors through the trained TF-IDF model which generates the numeric matrices that are to undergo the machine learning stage.

4.5. Model Design and Implementation

Four machine learning algorithms are used in the design of the model. Multinomial Naive Bayes, Logistic Regression, Linear Support Vector Machine (SVM) and the Random Forest. Both models possess their own advantages and balance each other in regards to the data.

Multinomial Naive Bayes classifier is used first due to simplicity and its usage when dealing with text classification problems where independence of features can be assumed. The model approximates chances of affective condition based on occurrence of words and is effective when using TF-IDF characteristics. A linear model of prediction is then done using the Logistic Regression, which is the use of logistic function to predict probabilities. It also offers interpretability following the coefficients which point out the intensity and direction of word impact on sentiment.

Linear SVM model is formed to determine a hyperplane that can be used to discriminate best between positive and negative examples. It works well with large dimensional data like text vectors. A convergence limit is set to 1000 iterations to be used in the implementation. Random Forest is presented as an ensemble technique, which makes use of multiple decision trees and averages their output. The model is set up to have 50 estimators and a maximum depth of 20 to strike balance between performance and computational efficiency. The features of the training set include TF-IDF and each model is trained with them and validated with the testing set.

4.6. Lexicon-Based Model Implementation

Three lexicon-based systems are also included in the system: VADER, TextBlob and SentiWordNet. Such approaches estimate sentiment by using pre-existing vocabulary directly rather than supervised learning. VADER is social media specific and gives a compound score, which is a value that reflects the intensity and the polarity of the tweet [13]. TextBlob gives a negative or a positive polarity. SentiWordNet needed further processing that included tokenisation, part-of-speech tagging and mapping tokens to WordNet synsets. In each synset, the mean of positive and negative sentiment scores was obtained to develop a general sentence-level polarity. The low-quality text versions that were cleansed were used by these lexicon methods as baseline comparisons between the machine-learning and deep-learning models.

4.7. LSTM Model Implementation

A Long Short-Term Memory (LSTM) model was applied to learn sequential dependencies in the text of tweets [14]. The sample size of 120,000 tweets was taken to train to ensure efficiency in computation. A tokenizer of 40,000 vocabulary size was used to convert text into a sequence of integers and zero-padded to an even length of 40 tokens. This model was made up of an embedding layer, an LSTM (128) layer, dropout and classification dense layers. 10 epochs of binary cross-entropy loss training were done. The LSTM model was subsequently trained and tested on a withheld section of the sampled information and subsequently applied to categorize new example tweets.

4.8. BERT Model Implementation

In order to conduct an analysis on transformer-based sentiment analysis, the system contains a fine-tuned BERT model (bert-base-uncased) [15]. A reduced group of 20,000 tweets was also

chosen to be trained because of the large computational cost associated with BERT. The Tweets were tokenized by the BertTokenizerFast with the maximum sequence length of 64. A PyTorch Dataset and DataLoader were developed to process batching. The model was trained on 10 epochs with the optimizer AdamW. After training, the model was able to achieve high performance on the test split due to the ability of BERT to learn contextual associations between text. Prediction of new tweets was also done using this model.

4.9. Evaluation Framework

Standard classification measures: accuracy, precision, recall and F1-score are used to evaluate the trained models. These measures offer the holistic approach of model performance with both the correctness and reliability. Confusion matrices are calculated so as to visualize the number of positive and negative tweets that were correctly or incorrectly classified. This assists in knowing the common misclassification tendencies and model weakness.

The results are tabulated and compared to find the algorithm that provides optimal trade-off between the accuracy and interpretability of each particular model. The highest accuracy is usually achieved by the use of Logistic Regression and Linear SVM, which depict the appropriateness of the linear models to classify the texts. The ability of each model to differentiate between positive and negative sentiments at threshold values is analyzed by means of ROC curve and the corresponding area under the curve (AUC) scores.

Another significant element of evaluation is visualization. The findings are presented in bar charts, heatmaps and ROC plots with the help of Matplotlib and Seaborn. The graphical outputs ensure that model behavior and performance trends can be easily interpreted.

4.10. Exploratory and Statistical Analysis

The design contains an exploratory analysis part, which investigates different aspects of the data before and after preprocessing. To ensure that the balance of classes is correct, pie charts and count plots are used to visualize sentiment distribution. The distribution of the length of tweets and the number of words used in them are analyzed with the help of histograms and boxplots which provide us an idea of the tweet composition. Horizontal bar charts are used to present the most common words in positive and negative tweets providing an intuitive insight into sentiment vocabulary.

These exploration tasks are useful to ensure that data quality is correct, as well as inform subsequent preprocessing or feature engineering. As an example, when there seems to be a longer length of positive tweets or when there are phrases, they can be used to shape a tailored set of token filters or weighting models.

4.11. Prediction and Real-Time Testing

The system was tested on a small number of new tweets, which were not included in the Sentiment140 dataset after the model training and evaluation. The examples have been chosen to account for both positive and negative expressions. The entire training process was repeated on every tweet and all the implemented models were compared to analyze each tweet. These are the four classical machine-learning models (Naive Bayes, Logistic Regression, Linear SVM, Random Forest), the three lexicon-based models (VADER, TextBlob, SentiWordNet), the LSTM model and the fine-tuned BERT model.

The classical models were based on prediction by means of TF-IDF vectors, whereas lexicon methods were applied directly to the cleaned text. The LSTM and BERT models were based on their respective sequence representations. In all the models, the forecasts were in line with the sentiment expected of each tweet. The most consistent results were given by the Logistic Regression, Linear SVM, LSTM and BERT, with the BERT giving the highest scores on confidence. Another way in which the system can be used is to visualise the classical model

predictions in a horizontal bar chart with positive examples represented in green and negative ones represented in red. This enables the user to grasp the polarity of any given tweet promptly. This step in multi-model prediction illustrates the way the various methods of sentiment analysis perceive the same input and points out differences between lexicon and learning-based methods.

4.12. System Implementation Environment

It is implemented on a Jupyter Notebook with Python 3. Data manipulation is done through the libraries Pandas and Numpy with Matplotlib and Seaborn taking care of visualization. Text preprocess is done using NLTK and model building using Scikit-learn. The whole workflow is also a module and it can be easily replicated and adjusted to other datasets. The architecture is scalable and adaptive.

4.13. Summary

The design embeds a complete sentiment-analysis pipeline, which takes raw Twitter text through stages of cleaning, preprocessing and TF-IDF feature extraction and then uses the text for training machine-learning, lexicon-based and deep-learning models. The components of each are modular, permitting efficient experimentation and comparison across algorithms. Traditional models employ TF-IDF vectors, whereas LSTM and BERT use sequence representations for deeper contextual learning. Visualization, evaluation metrics and real-time predictions, thus, constitute the system, which is interpretable, scalable and reproducible for large-scale Twitter sentiment classification.

5. Implementation

The implementation was done in a well-organized pipeline that took unprocessed Twitter text and processed it up to full-fledged sentiment-classification models. The functioning started with loading the Sentiment140 dataset of 1.6 million tweets and the subsequent preparation of the dataset (in terms of deleting the neutral elements and transforming the labels into a binary system). This was done by a series of cleaning operations that standardised the text by removing URLs, mentions, hashtags, punctuations and extra whitespace, followed by tokenisation, stopword removal and lemmatisation of the text to clean the noise and extract the underlying meaning of each tweet. The resulting processed texts were then converted into TF-IDF vectors based on a 5000-feature vocabulary of unigrams and bigrams to create the numerical representation needed by the classical machine-learning models. Naive Bayes, Logistic Regression, Linear SVM and Random Forest were four classifiers that were trained and tested on these features in an 80/20 split to evaluate performance in terms of accuracy, precision, recall, F1-score, confusion matrices and ROC curves. In addition to these models, three systems, based on lexicon (VADER, TextBlob and SentiWordNet), were applied as rule-based baselines. Two additional deep learning models were included to further extend the comparison: an LSTM network, which was trained on the tokenised sequences and another model, a fine-tuned BERT model trained with the help of PyTorch. Lastly, each of the models was trained on test example tweets not seen and the behaviour of the models could be compared directly, which showed the generalisation capability of the overall system.

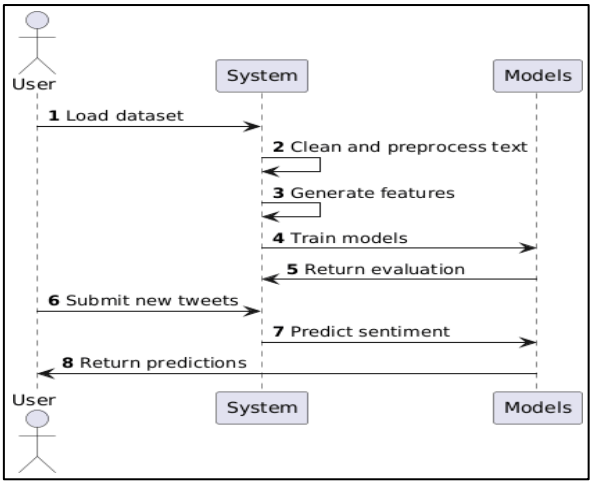

Fig 2: Workflow Diagram

6. Evaluation

6.1. Overview of the Evaluation Process

The sentiment analysis system evaluation is dedicated to the idea of classifying the tweets based on the positive and negative sentiment models. The evaluation includes quantitative evaluation of accuracy, precision, recall and F1-score and visual evaluations with the help of confusion matrices and ROC curves. All these measures give a comprehensive outlook of performance, interpretability and robustness. The testing stage is done on a large set of 318,126 tweets, which is 20 per cent of the total Sentiment140 dataset. The evaluation plan will be used to test not only the raw predictive power, but also the reliability and balance of each classifier in dealing with the real-world text data.

6.2. Dataset Integrity and Preprocessing Validation

The dataset was checked beforehand to determine completeness and consistency. The number of tweets in the dataset was 1.6 million and no values were missing. The preprocessing and cleaning process left behind 1,590,627 tweets, which proves that there was a small loss of data in the transformation. There was an equal representation in both classes of sentiment, with 800,000 positives and 800,000 negatives to form the sample of the perfectly balanced dataset. The mean length of tweets was about 74 characters and the mean number of words per tweet was 13 words, which is in line with the short text characteristic of Twitter. These findings proved that the data set was a good foundation to train and test machine learning models without bias and imbalance. Preprocessing, such as cleaning, tokenization, stopword elimination and lemmatization, also helped a lot in enhancing the uniformity of the text. The clean sample outputs were meaningful and concise, with sentiment cues effectively represented. This guaranteed that the TF-IDF vectorization was going to concentrate on significant linguistic patterns as opposed to noise.

6.3. Exploratory Data Analysis (EDA)

The exploration stage was used to understand the patterns of sentiment and word distributions prior to the evaluation of the model. The distribution of both positive and negative feelings was balanced, which proved the balance of the dataset. The histogram charts indicated that the majority of tweets were between 10 and 20 words in length, which indicates the brevity of Twitter. The box plots with text length and word count as a measure of sentiment showed comparable distributions with minor differences and no major bias to either of the classes.

The top word frequency analysis demonstrated the differences between the emotional tones of the classes. Negative tweets had common words such as sad, miss, bored and tired,

whereas the positive tweets had the words good, love, great and happy. These results were indicative of intense word polarity tendencies, which supported the idea that the labeling of sentiments in the dataset was correlated with actual semantic differences.

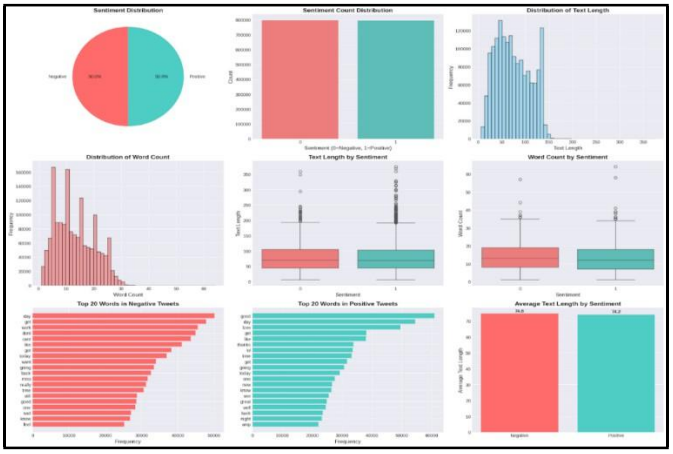

Fig 3: EDA Plots

6.4. Model Evaluation Summary

The TF-IDF features were tested using four supervised machine learning algorithms: Multinomial Naive Bayes, Logistic Regression, Linear Support Vector Machine (SVM) and Random Forest. The summary of the results is presented below:

There was an overall trend of performance indicating that Logistic Regression and Linear SVM obtained the highest accuracy and F1-scores of about 0.78 each. Naive Bayes was next with low vehicle pursuit, with random forest being the last with less accuracy, though with high recall. The findings affirmed that the text is best represented by TF-IDF features, which are high-dimensional, sparse data and best fitted by linear models.

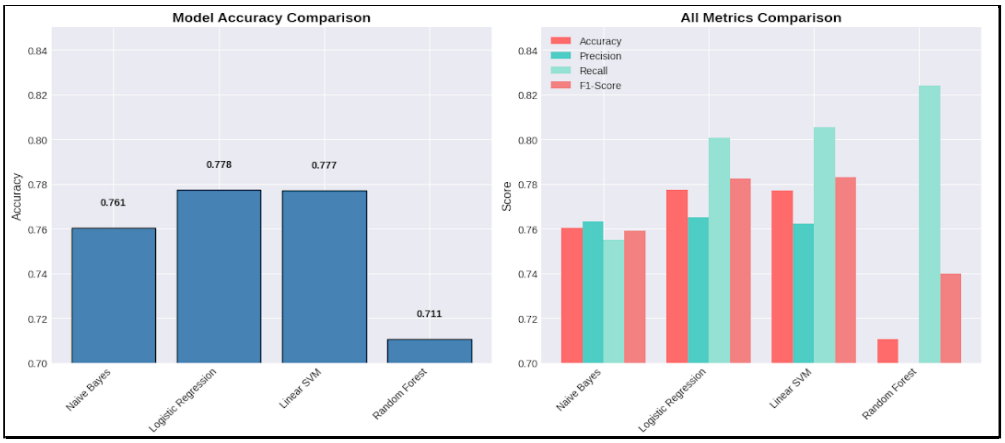

Fig 4: Model Comparison Bar Chart

6.5. Confusion Matrix Analysis

Confusion matrices offered a finer level of knowledge about prediction errors. Logistic Regression and SVM had equal true positives and true negatives, which showed equal classification accuracy of both types of sentiments. Naive Bayes resulted in a few more false positives, which implies that it is more likely to make a false positive mistake. Random Forest had the highest misclassification rate, especially between negative and positive tweets.

In the case of Logistic Regression, 120,012 negative and 127,333 positive tweets were correctly identified, which is a total of 77.8 percent. The Linear SVM model had an equal balance, given that 119,147 negatives and 128,064 positives were correctly classified. Naive

Bayes somewhat missed the correct predictions, but it was stable. The lower specificity of Random Forest meant overfitting or less generalization to scanty textualized features.

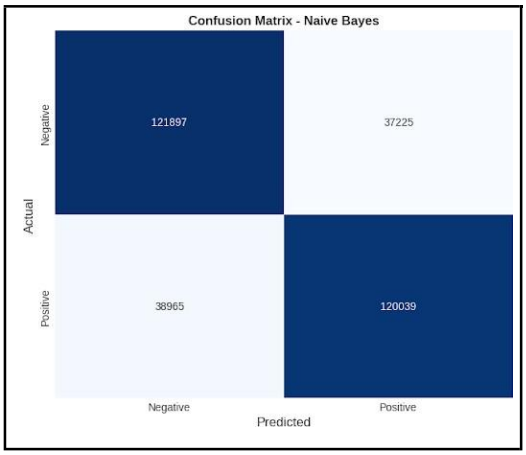

Fig 5: Confusion Matrix - Naive Bayes

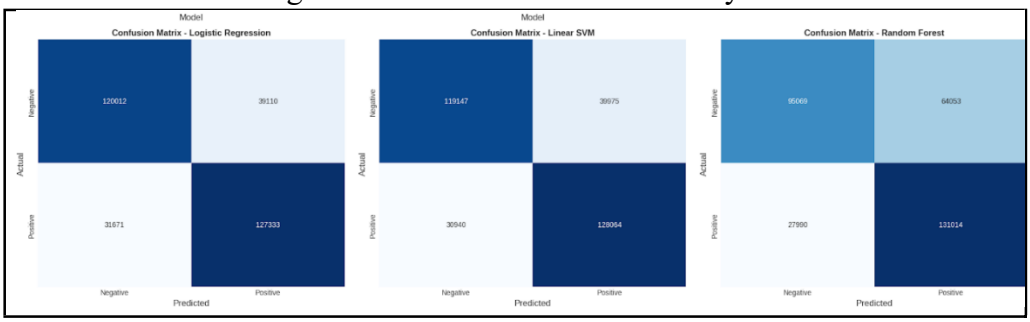

Fig 6: Confusion Matrices Visualization Placeholder

6.6. ROC Curve and AUC Analysis

All models were used to plot Receiver Operating Characteristic (ROC) curves to measure the discriminative capability of the model at varying thresholds. The Area Under the Curve (AUC) offers one area to measure performance. Logistic Regression and Linear SVM had an AUC value of 0.858 and Naive Bayes was next in the run with 0.844. Random Forest also had a smaller AUC of 0.791, which validates its rather diminished ability to separate positive and negative sentiment.

Such curves show that the linear models (Logistic Regression and SVM) have the same sensitivity and specificity, even with different classification thresholds. The observation is further supported by the fact that linear classifiers are more efficient than the ensemble approaches in this situation, given their larger AUC values.

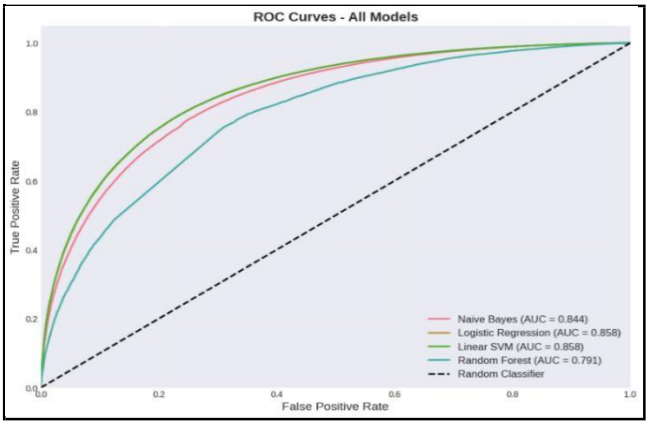

Fig 7: ROC Curve

6.7. Lexicon-Based Model Evaluation

The three methods based on lexicon (VADER, TextBlob as well as SentiWordNet) were assessed to offer a rule-based baseline. VADER had the best performance in this category with an accuracy of 0.6514 with the assistance of punctuations, degree modifiers and emojis. TextBlob was more polarizing and unable to deal with informal expressions, whereas SentiWordNet gave more neutral values since it relied on proper POS-tagging. They were quick and understandable and their general performance was far lower than the machine learning and deep learning models.

6.8. LSTM Model Evaluation

The LSTM was trained on a smaller set of 120,000 samples because of the computational limitations, but it still performed well. It obtained an accuracy of 0.7348 and an F1-score of 0.7403. This model was used to capture sequential patterns and deal with longer expressions in a better way than the TF-IDF-based models. Although it did not outperform Logistic Regression or Linear SVM, it outperformed all lexicon-based models and was generally stable in generalisation.

6.9. BERT Model Evaluation

The overall performance of BERT was the best. It achieved 0.8137 and an F1-score of 0.8089 even when trained with a limited population of 20,000 samples. BERT was much more efficient with subtle phrasing, multi-word phrases and contextual indicators than any other model. Its forecast of any new tweet was also very high. This points to the benefit of transformer-based sentiment analysis.

6.10. Interpretation of Results

The comparison shows how efficient the conventional linear algorithms are in the processing of short and informal text information like tweets. Both Logistic Regression and Linear SVM were able to learn subtle word association using the TF-IDF representation. The fact that they were very close in terms of their performance metrics implies that the two models arrived at similar decision boundaries, though there were slight differences in terms of regularization or optimization procedures. Naive Bayes was not as complex, but was an effective method, as it proved to be appropriate in large-scale text classification, where interpretability and speed are important. Random Forest, however, was not so strong, mainly because it utilizes hierarchical splits in decision making, which is not effective when dealing with sparse vectors. The model had a higher recall (i.e., it was more relaxed in labeling tweets as positive, but more precise this way), but at the expense of a lower precision. One can see this trade-off in the confusion of the table and the precision-recall balance. Comprehensively, the analysis proved that linear models, particularly the Logistic Regression, were the best trade-offs in terms of interpretability, performance and time consumption.

6.11. Evaluation on New Data

Unseen tweets were predicted to ensure that the model was tested on an independent set other than the test set. All implemented models were used to predict the sentiment of six unseen tweets, including lexicon-based methods, classical machine learning models, LSTM and BERT. The forecasts were close to the anticipated sentiments and this exhibited good generalization. Positive expressions such as “I love this product! It’s amazing and works perfectly!” were rightly attributed to positive, whereas negative ones such as “Terrible customer support. Waste of time and money.” were identified as negative. The predictions showed that the classical models were consistent, lexicon-based methods were more sensitive to phrasing, the LSTM handled longer expressions well and BERT produced the highest

confidence and most accurate outputs. Polarity separation was well represented visually in the predictions, which made them easier to interpret for non-technical people.

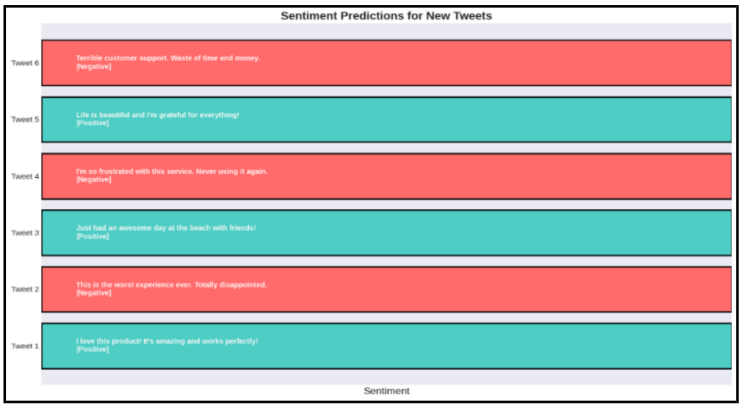

Fig 8: Prediction Visualization

6.12. Comparative Discussion

Under the comparative perspective, the Logistic regression and SVM models had a great balance in terms of precision and recall. They both attained an F1-score of around 0.78 which shows strong classification of classes of sentiments. The high-ranking consistency of the results of the two models shows that the linear classifiers are stable under the sparse representation of texts. Naive Bayes was still a sensible lightweight choice in fast sentiment screening applications, whereas Random Forest needed additional optimization or feature optimization to achieve higher accuracy.

The TF-IDF representation was useful in obtaining these results. It weighed the rare but informative words and enabled the models to concentrate on important sentiment words like love, hate, good and terrible and reduced the contribution of high-frequency neutral tokens. This was found to be a valid choice of vectorization in this data.

This was concluded using ROC-AUC comparison and confusion matrices that indicated that, in addition to being more accurate, Logistic Regression and Linear SVM were also reliable in sentiment differentiation when compared at different confidence levels. Such models can be safely applied to cases where the user is to classify text sentiment in social media in a fast and correct manner.

Beyond the classical models, evaluations also included lexicon-based tools and deep learning methods. VADER, TextBlob and SentiWordNet offered fast and interpretable results but performed noticeably lower on informal or ambiguous tweets. The LSTM model improved performance by modelling word order and sequential dependencies. BERT achieved the highest overall accuracy and confidence, demonstrating stronger contextual understanding than any other approach.

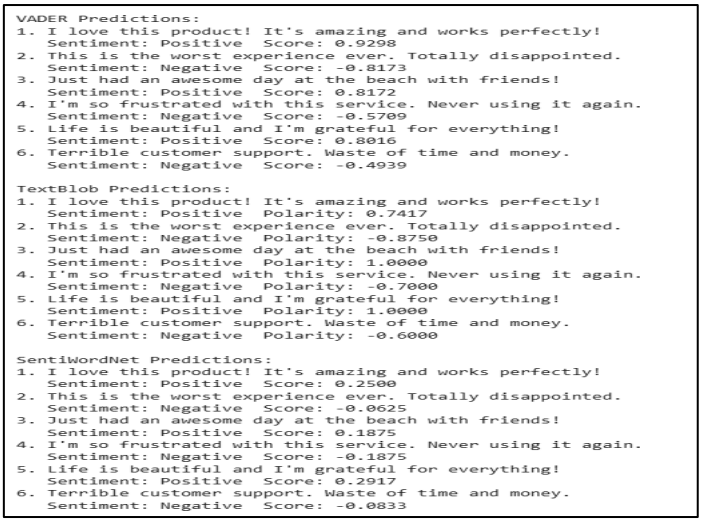

Fig 9: Prediction using VADER, TextBlob and SentiWordNet

6.13. Results and Critical Analysis

The analysis indicated that the sentiment analysis pipeline was efficient in processing and categorizing 1.59 million of the cleaned tweets, reaching a balance in the distribution of sentiment and a constant quality of the preprocessing process. The highest accuracy of 0.78 and AUC of 0.858 made Logistic Regression and Linear SVM the most suitable models. However, BERT achieved the highest accuracy and strongest generalisation, even when trained on a smaller subset of the data. The strong performance of Logistic Regression and Linear SVM is still consistent with previous research, where TF-IDF–based linear models are widely reported as effective baselines [6,9]. Naive Bayes was next in line with a lower precision and the random forest came next, as it had insufficient data. The reliable patterns of classification with low bias were demonstrated through visual inspection, such as confusion matrices and ROC curves. The system achieved good prediction of new unseen tweets, which proves good generalization and applicability. In general, the findings can be used to corroborate the literature that linear models are still useful in sentiment analysis on a large scale. The inclusion of both LSTM and BERT in the evaluation showed that deep learning models are already integrated into the system and contribute significantly to contextual understanding and improved performance.

6.14. Conclusion of Evaluation

The analysis of the evaluation process proves that the sentiment analysis pipeline works effectively throughout the stages of preprocessing to prediction. The findings show that linear models, especially Logistic Regression, are best used when the feature is TF-IDF and the dataset is Sentiment140. The results indicate that deep learning strategies have an additional potential to improve the performance, yet in the case of standard machine learning strategies, the existing system has a good trade-off between scalability and accuracy.

The methodological design, preprocessing strategy and feature selection are justified by the success of the evaluation. It sets a basis for future experimentation, including the inclusion of lexicon-based or transformer models to interpret sentiments more subtly. All in all, the system can be considered strong, interpretable and useful on addressing large-scale sentiment classification on Twitter data.

6.15. Discussion

The results were able to meet the main objective of the project, that is, to develop a proper, reliable and interpretable sentiment analysis pipeline, which is dedicated to Twitter data. The constructed models had been expected to do well and they actually recorded good performances

with worthy rates of accuracy and F1-scores of around 0.78. These statistics suggest clearly that the models have a strong capability of successfully managing and handling the sparse and high-dimensional TF-IDF variables that are typical of text-based datasets. These results are in line with the earlier research studies in the area, which have all pointed to and proved the efficacy of linear classifiers when tasked with sentiment analysis of text. The model was tested and verified in many ways and proved to be highly robust, scalable and generalizable in the case of application in real-life situations with previously unseen tweets. The model was found to be practical as its performance was not affected by the type of data sample. Overall, the workflow was able to address the three main issues that were clearly defined and incorporated into the project requirements: the noise of data in the contents of social media, the massive variability and diversity of text expressions and the problem of class balance in the data set. These achievements support the efficiency of the offered solution to address the real-world sentiment analysis issues on social media. The broader comparison of all model families reflected evident performance differences. Lexicon-based approaches were quick and failed to deal with colloquial language and meaning. The LSTM model was able to capture sequential structure in a better way and it performed better than all lexicon-based tools. BERT demonstrated the best accuracy and generalisation and was able to deal with finer phrasing and contextual information more than classical approaches to machine learning. The findings highlight the strengths of transformer-based solutions for short, informal social media text.

7. Conclusion and Future Work

7.1. Conclusion

The paper was able to come up with a full-fledged, fully developed and repeatable sentiment analysis platform with the proven ability to correctly classify and categorize the overall opinion and sentiment of a large volume of Twitter data. High performance and effectiveness of the linear models, particularly the Logistic Regression algorithm, which has been proven to be highly effective, can be seen as the impressive success of the conventional and traditional machine learning algorithms in the actual application of text analytics in real-life contexts. This study will be very resourceful and will contribute much to the academic literature and the field of practice by confirming and verifying the fact that a light, computationally efficient and yet highly effective sentiment prediction system is indeed deployable using small resources.

7.2. Future Work

Future research directions and extensions may be to include more complex transformer-based architectures or a hybrid deep learning architecture to capture and extract finer semantic hues, contextual meanings and better sarcasm detection capacity, which are difficult for traditional methods. Furthermore, it would be completely viable and advantageous to broaden and extend this known pipeline to multilingual sentiment analysis or real-time streaming data analysis functionality, thus making it far more legitimate, diverse and useful in various fields, including social media analytics, brand attending, analysis of customer feedback and business intelligence software. These improvements would also make the system useful in many industries and applications.

8. REFERENCE

Hota, H. S., Sharma, D. K., & Verma, N. (2021). Lexicon-based sentiment analysis using Twitter data: a case of COVID-19 outbreak in India and abroad. In Data science for

COVID-19 (pp. 275-295). Academic Press. https://www.sciencedirect.com/science/article/pii/B9780128245361000150
Sudhir, P., & Suresh, V. D. (2021). Comparative study of various approaches, applications and classifiers for sentiment analysis. Global Transitions Proceedings, 2(2), 205-211. https://www.sciencedirect.com/science/article/pii/S2666285X21000327
Bhakuni, M., Kumar, K., Sonia, Iwendi, C., & Singh, A. (2022). Evolution and evaluation: Sarcasm analysis for twitter data using sentiment analysis. Journal of Sensors, 2022(1), 6287559. https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/6287559
Wang, Y., Guo, J., Yuan, C., & Li, B. (2022). Sentiment analysis of Twitter data. Applied Sciences, 12(22), 11775. https://www.mdpi.com/2076-3417/12/22/11775
Rodríguez-Ibánez, M., Casánez-Ventura, A., Castejón-Mateos, F., & Cuenca-Jiménez, P. M. (2023). A review on sentiment analysis from social media platforms. Expert Systems with Applications, 223, 119862. https://www.sciencedirect.com/science/article/pii/S0957417423003639
Qi, Y., & Shabrina, Z. (2023). Sentiment analysis using Twitter data: a comparative application of lexicon-and machine-learning-based approach. Social network analysis and mining, 13(1), 31. https://link.springer.com/article/10.1007/s13278-023-01030-x
AlBadani, B., Shi, R., & Dong, J. (2022). A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM. Applied System Innovation, 5(1), 13. https://www.mdpi.com/2571-5577/5/1/13
Aljedaani, W., Rustam, F., Mkaouer, M. W., Ghallab, A., Rupapara, V., Washington, P. B., ... & Ashraf, I. (2022). Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry. Knowledge-Based Systems, 255, 109780. https://www.sciencedirect.com/science/article/pii/S0950705122009017
Srivastava, R., Bharti, P. K., & Verma, P. (2022). Comparative analysis of lexicon and machine learning approach for sentiment analysis. International Journal of Advanced Computer Science and Applications, 13(3), 71-77. https://www.academia.edu/download/87358181/Paper_12-Comparative_Analysis_of_Lexicon_and_Machine_Learning_Approach.pdf
Catelli, R., Pelosi, S., & Esposito, M. (2022). Lexicon-based vs. Bert-based sentiment analysis: A comparative study in Italian. Electronics, 11(3), 374. https://www.mdpi.com/2079-9292/11/3/374
van der Veen, A. M., & Bleich, E. (2025). The advantages of lexicon-based sentiment analysis in an age of machine learning. PloS one, 20(1), e0313092. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0313092
Raees, M., & Fazilat, S. (2024). Lexicon-based sentiment analysis on text polarities with evaluation of classification models. arXiv preprint arXiv:2409.12840. https://arxiv.org/abs/2409.12840
Alam, K. N., Khan, M. S., Dhruba, A. R., Khan, M. M., Al-Amri, J. F., Masud, M., & Rawashdeh, M. (2021). [Retracted] Deep Learning-Based Sentiment Analysis of COVID-19 Vaccination Responses from Twitter Data. *Computational and Mathematical Methods in Medicine*, *2021*(1), 4321131. https://scholar.google.com/scholar?output=instlink&q=info:WUrlp4WHDPAJ:scholar.google.com/&hl=en&as_sdt=0,5&scillfp=9602880915915947 36&oi=lle
Mollah, M. P. (2022). An LSTM model for Twitter sentiment analysis. *arXiv preprint arXiv:2212.01791*. https://arxiv.org/pdf/2212.01791
Thakkar, G., Hakimov, S., & Tadić, M. (2024). M2sa: multimodal and multilingual model for sentiment analysis of tweets. *arXiv preprint arXiv:2404.01753*. https://arxiv.org/pdf/2404.01753