International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 16 (2024) pp. 162–183

Article

Twitter Sentiment Analysis of COVID-19 Vaccination Integrating SenticNet-7 and SentiWordNet-Adjusted VADER Models

Perumal Chockalingam Sridevi * and Thambusamy Velmurugan

PG and Research Department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, Tamil Nadu 600106, India; velmurugan_dgvc@yahoo.co.in
* Correspondence: sridevipc@gmail.com

Received date: 5 March 2024; Accepted date: 11 May 2024; Published online: 24 May 2024

Abstract: Social media platforms in the modern era are enormous informational databases that continually generate massive volumes of data that provide deep insights into human thought, behaviour, and trends. Nonetheless, it is now crucial and difficult to extract relevant data from this vast and unstructured social media data pool. The objective of this study is to improve the accuracy of sentiment analysis by combining various techniques designed for social media sentiment analysis, with a focus on Twitter data. People communicate and share ideas and opinions in a completely new way thanks to social media platforms like Twitter, which have also produced an abundance of data that is ready for analysis. But there are many obstacles to overcome when trying to manually extract pertinent information from this massive amount of unstructured data. This problem is addressed by data mining methodologies, which involve using a variety of statistical methods and algorithms to extract patterns, connections, and insights from huge databases. Text mining is a subfield of data mining that focuses on retrieving knowledge and information from unstructured textual data, especially content that users have created on social networking sites, such as posts, comments, reviews, and tweets. Text mining techniques allow sentiment analysis, topic extraction, and other important information to be extracted from social media text by utilising machine learning, linguistic analysis, and natural language processing. The goal of this project is to improve the accuracy of sentiment classification in social media mining, with a concentration on Twitter data. This is accomplished by combining a number of methods, such as SenticNet-7, a sentiment dictionary with a medical focus, and SentiWordNet-Adjusted VADER Sentiment Analysis (SAVSA-SN7). SentiWordNet is used by SAVSA-SN7 to provide sentiment ratings to individual words in tweets. Then, the Valence Aware Dictionary and sEntiment Reasoner (VADER) sentiment analyzer are used to refine the sentiment scores. SenticNet-7, which is customised for the medical sector, is also included to take into consideration sentiment peculiarities unique to this industry. The outcomes of the experiments show how effective this combination method is, especially when dealing with the brief text data that is common on Twitter, where sentiment can vary greatly depending on the context. The proposed methodology's evaluation highlights its accuracy and performance in capturing sentiment and generating insightful recommendations for decision-making processes. Through the integration of data mining, text mining, and social media mining techniques, this research contributes to advancing sentiment analysis, particularly in the context of Twitter data.

Keywords: natural language processing; twitter sentiment analysis; SentiWordNet; medical domain sentiment analysis; VADER

1. Introduction

Social networking platforms have become invaluable resources for analyzing public opinion and conversation in the digital age. Twitter is a well-known website where users may participate in real-time conversations and express their opinions on a variety of subjects. Twitter has emerged as a crucial forum for conversations about vaccination campaigns in the context of the continuing COVID-19 epidemic [1,2], offering researchers a wealth of data to investigate public attitudes and opinions. This study aims to provide greater understanding of public sentiment by examining Twitter discourse about the COVID-19 immunization. It is essential to comprehend people's attitudes, opinions, and concerns on social media in order to guide public health actions and plans. By employing sophisticated sentiment analysis methods, we want to glean insightful information from the massive amount of Twitter data, illuminating the range of viewpoints and emotions present in online community.

SenticNet-7 and SentiWordNet-Adjusted VADER are two examples of the cutting-edge sentiment analysis models that are integrated into the methods used in this work to improve sentiment classification accuracy. Sentiment subtleties unique to healthcare-related conversations are captured by SenticNet-7, a sentiment lexicon designed specifically for the medical field. Furthermore, SentiWordNet-Adjusted VADER offers a strong framework for sentiment analysis in social media data by fusing the advantages of SentiWordNet and VADER. Utilizing these cutting-edge methods, this study aims to provide a more thorough insight of public opinion by capturing the subtleties of sentiment conveyed in tweets about the COVID-19 immunization. In addition, the sentiment data is analyzed by machine learning algorithms, which makes it possible to spot trends and patterns in the Twitter conversation. The analysis's findings provide insightful information on the dynamics of public opinion on the COVID-19 vaccination on Twitter. Through an analysis of user sentiment, we are able to pinpoint recurring themes, feelings, and issues, giving legislators, public health authorities, and researchers useful information. This study adds to the body of knowledge regarding public attitudes and views regarding vaccination campaigns in the context of the COVID-19 pandemic by carefully analysing Twitter data. The literature review is offered in Section 2 of the paper, which is followed by a Section 3 description of the methodology, a Section 4 presentation of the results and discussions, and a Section 5 concluding remarks.

2. Literature Survey

In-depth academic studies and contemporary scholarly papers pertinent to the data analysis, text mining, and sentiment analysis domains are examined in the literature review conducted for this project. This part offers important insights that are essential for the study and provides a basic awareness of the difficulties, approaches, and developments in the field of sentiment analysis.

During a crucial period from June to July 2020, Kausar et al. [3] carried out a study that focused on sentiments within the current COVID-19 epidemic. Their findings showed a generally optimistic attitude across nations, suggesting adaptation and better recovery over time. Common phrases like "pandemic","COVID," and "health" were found through word cloud analysis, showing complex emotional undertones and providing a global view on COVID-19 that transcends national boundaries.

In order to obtain insights into pandemic-related conversations on social media and highlight public worries and awareness, Boon-itt et al. [4] placed a strong emphasis on sentiment analysis and topic modelling. Comparably, Pano et al. [5] carried out an extensive investigation of the digital domain, utilising special text pre-processing techniques to associate sentiment ratings from language on Twitter with Bitcoin values throughout the pandemic. Using Andalusia as a case study, Flores-Ruiz et al. [6] evaluated pandemic-induced changes in the tourist sector in the region by contrasting data from the Andalusia tourist Situation Survey (ECTA) with sentiment analysis from Twitter. Their research showed a relationship between the results of the ECTA survey and sentiment analysis using Twitter data, suggesting that traveler's priorities safety and seeks out less congested locations. In response, Zainuddin et al. [7] added to the conversation by putting up a brand-new hybrid approach that combines PCA, LSA, and random projection feature selection with a hybrid classification technique for data validation. They also presented evaluation methods for figuring out the outcome. This survey of the literature provides a thorough summary of current developments in text mining and sentiment analysis. Advances in machine learning and natural language processing have brought about a substantial transformation in sentiment analysis techniques. Various tactics, such as VADER, SentiWordNet, Support Vector Machines, and Logistic Regression approaches, are explored to accurately categories emotions.

3. Material and Methods

Four main components make up the research methodology: gathering data from Twitter, preparing the data, classifying the results using the SenticNet-7 with SentiWordNet-Adjusted VADER Sentiment Analysis (SAVSA-SN7) model, and evaluating the results. This strategic framework depends on thorough data collection, thorough preprocessing processes, sophisticated classification systems, and stringent evaluation protocols.

3.1. Data Set

In constructing our dataset, the research work employed the Twitter API [8], a potent instrument that provides access to Twitter's enormous collection of openly accessible data. Through this API, this work accessed and retrieved relevant data points from the Twitter platform, capturing a diverse range of tweets related to this research topic. These efforts yielded approximately 3000 data points, each representing a unique piece of information extracted from Twitter. Sample instance of the data point are showcased in Table 1. This extensive dataset serves as the foundation upon which our study is built. It offers a thorough understanding of the topic under inquiry and serves as a rich supply of information for the analyses and insights. By gathering a diverse array of tweets from various users and contexts, this research aim to capture the breadth and depth of perspectives surrounding our research topic.

Table 1. Sample Data.

tweet_id	Sentiment	Sentiment.1	Text
13456666666	Negative	−0.05271	4,000 a day dying from the so called Covid-19 "vaccine" @DailyBeast reports. #vaccine #PfizerVaccine #Moderna https://t.co/p1nQWWZpk4 (access on 5 March 2024)
13334444555	Neutral	0.050462	Pranam message for today manifested in Dhyan by @meenapranam #truth #love #karm #light #nature #consciousness #FridayThoughts #fridaymorning #CoronavirusIndia #COVID19India #?????_???????? #navratri #Thane #AmbedkarJayanti2021 #ModiJi #NarendraModi #SecondCOVIDWave #Covaxin https://t.co/bQNoMVowJg (access on 5 March 2024)

3.2. Text Pre-Processing

To make the textual data more relevant and of higher quality, a sequence of pre-processing steps was meticulously implemented on the raw content obtained from Twitter. These pre-processing measures [9] were designed to refine the text, remove extraneous noise, and ready it for subsequent analysis. The preprocessing pipeline begins with data cleaning, ensuring that the text is free from extraneous characters and artifacts that may hinder analysis. This includes removing newline characters, non-ASCII characters, and repeated characters while preserving certain words like "vaccine."

Code Snippet for Pre-processing

```
# Read the CSV file
df = pd.read_csv(file_path, encoding='utf-8')
def remove_repeated_chars(text, preserve_words=[]):
def remove_repeats(match):
  char = match.group(1)
  return char * 2  # Change to char * 2 to preserve repeated characters
preserved_text = text
for word in preserve_words:
preserved_text = preserved_text.replace(word, '' + word + '')
pattern = re.compile(r'(\w)\1+')  # Modify the pattern to preserve only letters
text_without_repeats = pattern.sub(remove_repeats, preserved_text)
return text_without_repeats
def correct_spelling(text):
blob = TextBlob(text)
corrected_text = blob.correct()
return str(corrected_text)
def preprocess_text(text):
text = text.replace('\n', '')
text = re.sub(r'[^\x00-\x7F]+', '', text)
remove_repeated_chars_result = remove_repeated_chars(text, preserve_words=['vaccine'])
corrected_text = correct_spelling(remove_repeated_chars_result)
lowercase = corrected_text.lower()
remove_html = BeautifulSoup(lowercase, 'html.parser').get_text()
```

```
remove_urls = re.sub(r'http\S+', '', remove_html)
remove_usernames = re.sub(r'@\w+', '', remove_urls)
remove_hashtags = re.sub(r'#\w+', '', remove_usernames)
remove_numbers = re.sub(r'\d+', '', remove_hashtags)
remove_special_chars = re.sub(r'[^\w\s]', '', remove_numbers)
remove_repeated_chars_result = remove_repeated_chars(remove_special_chars,
    preserve_words=['vaccine'])
convert_reserved_words = re.sub(r'\bRT\b', 'retweet', remove_repeated_chars_result)
replace_smileys = emoji.demojize(convert_reserved_words)
replace_smileys = re.sub(r':[a-z_]+:', '', replace_smileys)
replace_smileys = re.sub(r'\([^)]*\)', '', replace_smileys)
replace_smileys = re.sub(r'(:\)|:-\)|:\(|:-\()', lambda m: 'smiley' if m.group() in [':)', ':-)'] else 'sad' if
    m.group() in [':(', ':-('] else '', replace_smileys)
tokens = word_tokenize(replace_smileys)
stop_words = set(stopwords.words('english'))
remove_stopwords = [word for word in tokens if word not in stop_words]
remove_short_words = [word for word in remove_stopwords if len(word) > 2]
tokens_pos = pos_tag(remove_short_words)
lemmatizer = WordNetLemmatizer()
no_lemmatize = ['vaccine', 'vaccines']
lemmatize = [lemmatizer.lemmatize(word, tag[0].lower() if tag[0].lower() in ['a', 'r', 'n', 'v'] else 'n')
    if word not in no_lemmatize else word for word, tag in tokens_pos]
preprocessed_text = ''.join(lemmatize)
…………………………….
…………………………….
print('Preprocessing completed')
```

The preprocessing applied on the data set are discussed below

Text Cleaning: The text is initially cleaned to remove any newline characters (\n) and non-ASCII characters using regular expressions.

Repeated Character Removal: A custom function remove_repeated_chars() is applied to remove repeated characters, preserving certain words like "vaccine" from being modified.

Spell Correction: Another custom function correct_spelling() is used to correct spelling errors in the text using the TextBlob library.

Lowercasing: The text is converted to lowercase to ensure consistency in text representation.

HTMLTagRemoval: We use the BeautifulSoup package to extract any HTML tags from the text.

URLRemoval: URLs are removed from the text using regular expressions.

UsernameRemoval: Usernames (starting with '@') are removed from the text using regular expressions.

HashtagRemoval: Hashtags (starting with '#') are removed from the text using regular expressions.

NumberRemoval: Numerical digits are removed from the text using regular expressions.

SpecialCharacterRemoval: Special characters (excluding word characters and spaces) are removed from the text using regular expressions.

ReservedWordConversion: The reserved word "RT" (indicating a retweet) is converted to "retweet" for consistency.

EmojiHandling: Emojis in the text are replaced with textual representations using the emoji library.

SmileyReplacement: Smileys and emoticons are replaced with generic terms like "smiley" or "sad".

Tokenization: The word tokenize() method from the NLTK package is used to tokenize the text into individual words.

StopwordRemoval: Stop words (commonly occurring words like "the", "and", "is") are removed from the tokenized text.

ShortWordRemoval: Words with a length less than or equal to two characters are filtered out from the tokenized text.

Part-of-Speech Tagging: Each token in the preprocessed text is tagged with its part-of-speech (POS) using NLTK's pos_tag() function.

Lemmatization: Words in the preprocessed text are lemmatized (reduced to their base or dictionary form) using NLTK's WordNet lemmatizer.

Final Text: Using the lemmatized tokens, the preprocessed text is rebuilt and saved for additional examination. The preprocessed text is saved in a new CSV file called "preprocessed_tweets.csv" once these preprocessing processes have been applied to each tweet in the dataset. The processed data is now prepared for additional modelling and analysis.

3.3. Text Vectorization Using TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is utilized for text vectorization, a crucial step in natural language processing [10] tasks such as sentiment analysis. TF-IDF assigns weights to terms based on their frequency in a document and their rarity across all documents in the corpus. Here's how TF-IDF is employed in the research work.

3.4. TF-IDF Vectorization

TF-IDF feature vectors are created from text data using the scikit-learn [11] {TfidfVectorizer` class.

- Grid search with cross-validation ({Grid-SearchCV}) is used to tune the hyperparameters of the `TfidfVectorizer} in an effort to maximise model performance by finding the ideal set of hyperparameters.
- "max features" (the maximum number of features to consider) and "ngram range" (the range of n-gram features to extract) are two hyperparameters taken into account during grid search.
- The TF-IDF vectorizer ({tfidf_vectorizer}) is then instantiated using the best-performing hyperparameters found via grid search.

3.5. Vectorizing Text Data

To convert raw text data into TF-IDF feature vectors, the instantiated TF-IDF vectorizer ({tfidf vectorizer}) is applied to the training, validation, and testing sets.

- Extra engineering characteristics, such average word length, word count, and the presence of particular keywords (like {has_keyword}), are concatenated with the converted TF-IDF feature vectors.

3.6. Feature Engineering Integration

- After combining the engineered features and TF-IDF-transformed features, a final feature matrix ({X_train_final}, {X_val_final}, {X_test_final}) is produced that may be used to train machine learning models.

3.7. Model Training and Evaluation

- The TF-IDF-transformed and engineered features are used to train machine learning classifiers, such as SVM [12], LinearSVC, and Logistic Regression, using the resampled training data ({X_train_resampled}, `y_train_resampled}).

The trained models are assessed using accuracy, precision, recall, F1-score, and confusion matrix metrics on the testing set ({X_test_final}, {y_test}).

3.8. Model Comparison and Interpretation

- To evaluate the efficacy of TF-IDF vectorization in sentiment analysis tasks, results from several models—including Random Forest, LinearSVC, and Logistic Regression—are compared based on their performance criteria.
- We examine and describe how TF-IDF vectorization affects model performance and how it contributes to increased robustness and accuracy in sentiment analysis.

Code for TF- IDF and ROS

```
# Feature Engineering
df['average_word_length'] = df['Text'].apply(average_word_length)
df['num_words'] = df['Text'].apply(num_words)
df['has_keyword'] = df['Text'].apply(has_keyword)
# Split the data into training, validation, and testing sets
X_train, X_temp, y_train, y_temp = train_test_split(df[['Text', 'average_word_length',
    'num_words', 'has_keyword']], df['combined_sentiment'], test_size=0.6, random_state=48)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.6, random_state=48)
X_train = X_train.fillna('')
X_val = X_val.fillna('')
X_test = X_test.fillna('')
# TF-IDF Vectorizer Hyperparameter Tuning
param_grid_tfidf = {
'max_features': [100, 500, 1000],
'ngram_range': [(1, 1), (1, 2)],
```

```
}
grid_search_tfidf = GridSearchCV(TfidfVectorizer(), param_grid_tfidf, cv=5, n_jobs=-1,
    scoring='accuracy')
grid_search_tfidf.fit(X_train['Text'], y_train)
best_params_tfidf = grid_search_tfidf.best_params_
# Update the TF-IDF vectorizer with the best hyperparameters
tfidf_vectorizer = grid_search_tfidf.best_estimator_
X_train_tfidf = tfidf_vectorizer.transform(X_train['Text'])
X_val_tfidf = tfidf_vectorizer.transform(X_val['Text'])
X_test_tfidf = tfidf_vectorizer.transform(X_test['Text'])
# Include additional features in the final X_train_final, X_val_final, and X_test_final
X_train_final = np.hstack((X_train_tfidf.toarray(), X_train[['average_word_length', 'num_words',
    'has_keyword']].values))
X_val_final = np.hstack((X_val_tfidf.toarray(), X_val[['average_word_length', 'num_words',
    'has_keyword']].values))
X_test_final = np.hstack((X_test_tfidf.toarray(), X_test[['average_word_length', 'num_words',
    'has_keyword']].values))
# Create a RandomOverSampler with the best hyperparameters
ros = RandomOverSampler(sampling_strategy={'neutral': 3175, 'positive': 3175, 'negative': 3175},
    random_state=22)
# Resample the training data
X_train_resampled, y_train_resampled = ros.fit_resample(X_train_final, y_train)
# Train a classifier on the resampled data with the best hyperparameters
clf = RandomForestClassifier(random_state=22)
clf.fit(X_train_resampled, y_train_resampled)
# Make predictions on the test set
y_test_pred = clf.predict(X_test_final)
```

3.9. Handling Class Imbalance With ROS

In order to overcome the problem of class imbalance, Random Over-Sampling [13] is used to make sure that each class obtains a enough amount of samples for training. In ROS, minority class samples are duplicated at random [14] until the distribution of classes is balanced. Over-sampling is done in the code by using the RandomOverSampler from the imbalanced-learn package (imblearn). To guarantee that, following oversampling, each class (negative, neutral, and positive) has an equal number of samples, the sampling_strategy parameter is supplied. In order to reduce bias towards the majority class and enhance model performance on unbalanced datasets, ROS is used to augment the training set with more instances from the minority classes.

4. Sentiwordnet-Adjusted Vader Sentiment Analysis (SAVSA-SN7)—Proposed Method

Determining the emotional context or attitude expressed in a given text is the main goal of sentiment analysis. Many hybrid approaches have been explored by many researchers [15–17] similar to that this work uses a combination of approaches to classify tweets according to their textual content into different sentiment classes. Several attempts have been made to incorporate approaches that polarity-assign textual data in order to guarantee accuracy. Accordingly, our study aims to combine VADER and SentiWordNet (SWN), offering a hybrid method explained as follows:

4.1. Obtaining Sentiment Scores Using SentiWordNet

A lexical repository of words is used to give some sentiment scores. The sentiment score of each word in a text is calculated by comparing its positive and negative values. An overall sentiment score for the text is then calculated by adding together these scores.

4.2. Sentiment Classification Utilizing SentiWordNet

The text is divided into words, and a preset function is used to calculate the sentiment score of each word. The text's sentiment is then evaluated to see if it is favourable, negative, or neutral based [18] on the cumulative sentiment score.

4.3. Refining VADER Scores with SentiWordNet

A specific function is designed to use the sentiment score obtained from SentiWordNet to improve the sentiment score produced by the VADER (Valence Aware Dictionary and Sentiment Reasoner) tool.

4.4. Standardizing the Adjusted Scores

The adjusted sentiment scores are normalised within the range of −1 to 1 to ensure consistency and comparability. Subtracting the lowest score, dividing by the score range, and finally mapping the normalised score to the assigned range are the steps in this normalisation process.

4.5. Sentiment Classification Employing the Adjusted VADER Scores

Different sentiment classifications are created for each tweet based on the normalised sentiment scores. To distinguish between good, negative, and neutral sentiments, a preset threshold value is used. Normalised scores that are higher than the threshold arecategorised as positive, scores that are lower than the threshold are classified as negative, and scores that are in the threshold range are labelled as neutral.

5. Classification Algorithm

A multimodal approach comprising multiple sentiment analysis approaches is used in the evaluation of the SentiWordNet-Adjusted VADER Sentiment Analysis (SAVSA-SN7) methodology. The test attempts to determine how well SAVSA-SN7 classifies textual data based on sentiment, utilisingSentiWordNet's semantic insights in conjunction with VADER, a potent rule-based sentiment analysis tool. Furthermore, the efficacy of SAVSA-SN7 [19] is thoroughly examined by comparing and evaluating its performance against recognised sentiment analysis approaches using methods like Support Vector Machine (SVM), Linear SVC, and Logistic Regression.

5.1. VADER (Valence Aware Dictionary and Sentiment Reasoner

A vocabulary and rule-based sentiment analysis tool created especially for text on social networking platforms is called VADER. It gives a sentiment intensity score for every word in the text, taking into account the sentiment's intensity as well as its polarity (positive or negative) [20]. The various word ratings are added together to determine the text's overall sentiment score. Using a pre-defined vocabulary, each word is awarded a sentiment value, with polarity ratings ranging from −4 to +4. The final sentiment score for the text is calculated by taking into account the intensity of the sentiment in addition to capitalization and punctuation.

5.1.1. Working of VADER

VADER lexicon Makes use of a pre-compiled list of terms together with sentiment scores. Based on the intensity of the sentiment conveyed by each word in the text, polarity scores are assigned. Sentence-level sentiment adds up each word's score to determine the text's total sentiment score.

5.1.2. Algorithm

1. Break apart the text into its constituent words.
2. From the VADER vocabulary, extract the sentiment intensity scores for every word.
3. Add up the ratings for each word to determine the text's overall sentiment score.
4. Depending on the overall sentiment score, you can optionally apply a threshold to divide the text into positive, negative, or neutral sentiment groups.

5.2. SentiWordNet

SentiWordNet is a lexical resource that uses the semantic relationships between words in WordNet to provide sentiment scores to individual words. It gives each word positive, negative, and objective scores that represent how positive, negative, or neutral the word's sentiment is. SentiWordNet's positive score is subtracted from its negative score to determine each word's sentiment score in the text. The final score reflects the word's overall sentiment polarity.

5.2.1. Working of SentiWordNet

Finding the synsets for every word in the text is known as synset retrieval.

emotion scoring: Based on the emotion scores connected to each word's synsets, each word is assigned a sentiment score (positive, negative, or neutral).

Score aggregation: Computes the text's overall sentiment score by combining the ratings of each individual word.

5.2.2. Algorithm

1. Break apart the text into its constituent words.
2. Get each word's SentiWordNet positive and negative sentiment scores.
3. Subtract the negative score from the positive score to determine the sentiment score for each word.
4. To determine the text's overall sentiment score, add up each word's emotion scores.

5.3. Support Vector Machine (SVM)

For classification problems, supervised learning algorithms like SVM are employed. It operates by locating the ideal hyper-plane in a high-dimensional space that divides the data points into various classes. SVM seeks to minimise [21,22] classification errors while maximising the margin between the classes. By identifying the decision boundary that maximises the margin between the support vectors of various classes, SVM aims to solve the optimisation problem. The hyperplane equation, which divides the classes in the feature space, represents the decision boundary.

Working of SVM

Finding the hyperplane that maximises the margin between classes is known as hyperplane optimisation.

When the data is not linearly separable in its original feature space, a kernel technique maps the input data onto a higher-dimensional space to allow for linear separation.

Classification: Algorithm determines which side of the hyperplane new data points land on to assign class labels to them.

1. Using labelled training data, where each data point is represented as a feature vector, train the SVM model.
2. By resolving the optimisation problem, determine the ideal hyperplane that divides the data points into distinct classes.
3. Use feature vectors to classify new data points by identifying which side of the hyperplane they belong to.

5.4. Linear Support Vector Classification (LinearSVC)

A linear kernel function is used by LinearSVC, an SVM version, to determine the ideal hyperplane for classification tasks. Large-scale datasets and binary classification issues are two areas in which it excels [23]. A linear decision function is used by LinearSVC to categorise data points into various classifications. The dot product of the weight and feature vectors plus an extra bias factor represents the decision function.

5.4.1. Working of Linear SVC

Finding the hyperplane in a linear feature space that maximises the margin between classes is known as margin maximisation.

Classification: Determines a new data point's distance from the hyperplane in order to assign class labels to it.

5.4.2. Algorithm

1. Use labelled training data, where each data point is represented as a feature vector, to train the LinearSVC model.
2. By using a linear kernel function to solve the optimisation problem, determine the ideal hyperplane that divides the data points into distinct classes.
3. Assign new data points to the class with the closest distance by calculating their distances from the hyperplane.

5.5. Logistic Regression (LR)

A statistical model used for binary classification tasks is called logistic regression. Based on its attributes, it calculates the likelihood that a given data point belongs to a specific class. The link between the categorical dependent variable [19] (class label) and the independent variables (features) is modelled. The log-odds of the chance that a data point belongs to a specific class are modelled by logistic regression as a linear function of the independent variables. The likelihood that the data point belongs to the positive class is then calculated by applying the logistic function, also known as the sigmoid function, to the log-odds.

5.5.1. Working of LR

Probability estimation: Makes use of the logistic function to determine the likelihood that an input piece belongs to each class.

Using the probabilities to categorise [23] the input into the most likely class, a decision boundary is determined.

Training: Modifies model parameters to optimise the observed data's probability given the model.

5.5.2. Algorithm

1. Using labelled training data, where each data point is represented as a feature vector, train the logistic regression model.
2. Use optimisation methods like gradient descent to minimise the logistic loss function in order to estimate the logistic function's parameters.
3. Use the learnt logistic function to calculate the chance that a new data point will belong to the positive class, then apply a threshold to get the class label.

5.6. Evaluation Metrics

Metrics such as precision, recall, accuracy, F1 score, and Cohen's Kappa are employed in the assessment of classification techniques. Together, they offer a thorough assessment of sentiment analysis models' performance, which facilitates the interpretation of their dependability and efficacy.

N = Number of Accurate Forecasts
Total Number of Predictions, or TNP
Tp stands for True Positives.
False Positives, or FP
False Negatives (FN)
TN stands for True Negatives.

5.6.1. Accuracy

The percentage of correctly classified instances relative to all instances is known as accuracy. It offers a general evaluation of the model's accuracy [22].

$$Accuracy = \frac{\mathrm{NP}}{\mathrm{TNP}} \tag{1}$$

5.6.2. Precision

The precision of the model is determined by dividing all of its positive predictions by the percentage of true positive forecasts. It demonstrates the model's capacity to steer clear of erroneous positive predictions [22].

$$\text{Precision} = \frac{Tp}{Tp + FP} \tag{2}$$

5.6.3. Recall (Sensitivity)

The percentage of accurate positive predictions among all real positive events in the dataset is measured by recall. It evaluates how well the model captures every positive example.

$$\mathrm{Re}\,call = \frac{TP}{TP+FN} \tag{3}$$

5.6.4. F1 Score

The harmonic mean of recall and precision is the F1 score. It offers a harmony between recall and precision, particularly in the case of unbalanced datasets [22].

$$F1 = 2 * \frac{\Pr ecission * \operatorname{Re} call}{\Pr ecission + \operatorname{Re} call} \tag{4}$$

5.6.5. Confusion Matrix

A table that compares expected and actual labels to provide a summary of the model's performance is called a confusion matrix. It offers information about the kinds of mistakes the model makes [22]. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are its four terms.

5.6.6. Cohen's Kappa

A metric called Cohen's Kappa is used to quantify the degree of agreement between actual and anticipated classifications while taking random variation into consideration [23]. It takes into account how much better the agreement is than would be predicted by chance.

$$k = \frac{P_O - P_C}{1 - P_C} \tag{5}$$

where Pc is the fictitious probability of chance agreement and Po is the proportionate observed agreement among raters. When combined, these measures offer a thorough assessment of sentiment analysis models' performance, which facilitates the interpretation of their dependability and efficacy.

6. Result and Discussions

Comparing and analysing the results obtained from applying the VADER, SentiWordNet, Support Vector Machine (SVM), Linear Support Vector Classification (LinearSVC), and Logistic Regression models is the main focus of the evaluation. We analyse the recall, accuracy, precision, F1 score, and confusion matrices in order to shed light on the subtleties and effectiveness of each strategy. This work also discusses the consequences and possible improvements in sentiment classification accuracy of integrating SentiWordNet-adjusted VADER Sentiment Analysis (SAVSA-SN7). This workseek to clarify the benefits, drawbacks, and potential areas for further development of sentiment analysis methods in the context of textual data processing by carefully analysing the findings.

VADER

Precision, recall, and F1-score metrics are given for each sentiment class—Negative, Neutral, and Positive—in Table 2 and Figure 1. These metrics show how well the algorithm can categorise examples within each sentiment category. It is noteworthy that the algorithm accurately identifies the Positive (87%) and Negative (85%) sentiment classes with high precision. But neutral mood has a far lower precision (32%), which points to a higher likelihood of false positives. Recall values differ between classes; in this case, Neutral sentiment has the highest recall (84%), suggesting that the model can successfully catch genuine positive events in this category. With F1-scores ranging from 47% to 78% across emotion classes, they offer a fair assessment of both recall and precision. Furthermore, the model's total accuracy—which accounts for its performance in all classes—is reported as 69%.

Table 2. Class Wise Performance Matrices of VADER.

	Precision	Recall	F1-score	Support
Negative	85	72	78	16,794
Neutral	32	84	47	3,504
Positive	87	60	71	9,702
Accuracy	69			30,000
Macro avg	68	72	65	30,000
Weighted avg	80	69	72	30,000

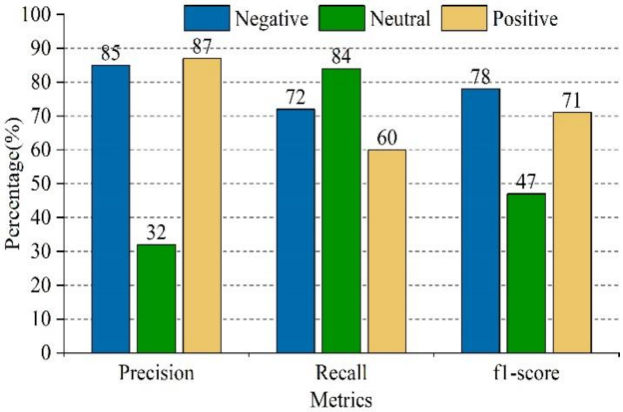

Figure 1. Class Wise VADER- Performance Matrices.

The VADER sentiment analysis model's predictions are broken down into three sentiment classes (Negative, Neutral, and Positive) and compared to the actual sentiment labels in a thorough manner in the confusion matrix shown in Figure 2. The matrix shows that, for the first sentiment class, 12,066 instances of negative sentiment were appropriately categorized as negative. Nevertheless, 738 cases of negative sentiment and 3,990 cases of negative sentiment were mistakenly classified as positive and neutral, respectively. This implies that even while the model does a good job of recognizing negative emotion, there are a good amount of misclassifications, especially into the Positive and Neutral categories. Regarding the class of Neutral sentiment, the matrix indicates that 2,958 occurrences of this attitude were accurately categorized. On the other hand, 96 cases of neutral sentiment were mistakenly labeled as positive, and 450 cases of neutral sentiment were mistakenly classified as negative. This suggests that it can be difficult to discern between neutral, negative, and positive thoughts with accuracy.

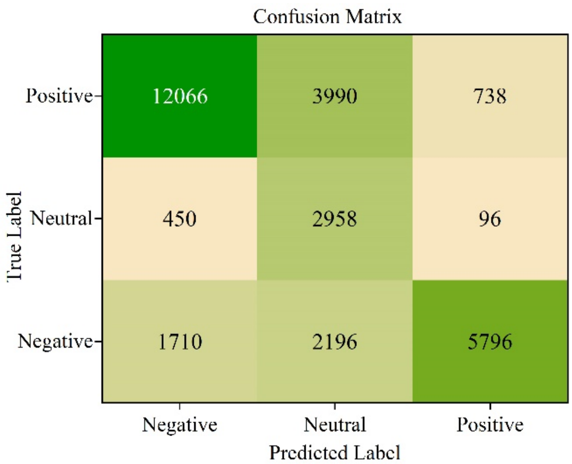

Figure 2. ConfusionMatrics VADER.

The confusion matrix shows that 5,796 occurrences of the Positive emotion class were correctly classified. On the other hand, 2,196 cases of Positive sentiment and 1,710 instances of Negative sentiment were misclassified as Neutral. This implies that even if the model is good at detecting positive sentiment, there are still situations in which it incorrectly labels positive emotion as neutral or negative. All things considered, the confusion matrix offers insightful information about how well the model performs across various sentiment classes. It draws attention to both areas of strength, like correctly detecting positive sentiment instances, and areas for growth, like correctly differentiating between negative, neutral, and positive attitudes. In order to improve the accuracy and efficacy of the model, these insights can direct additional analysis and sentiment analysis technique development. A breakdown of the cases that the VADER sentiment analysis model properly and mistakenly predicted for each sentiment class—Negative, Neutral, and Positive—is shown in Table 3 and Figure 3.

Table 3. VADER—Model Prediction: Correct vs. Incorrect.

	Correctly Predicted	**Wrongly Predicted**
Negative	12,066	4,728
Neutral	2,958	546
Positive	5,796	3,906

First, focusing on negative sentiment, the Table 3 and Figure 3 reveals the model correctly predicted 12,066 instances out of a total of 16,794 as negative. Nevertheless, 4,728 cases were incorrectly classified as negative when they truly fell into the categories of neutral or positive emotion. This shows that there are still a significant number of incorrect classifications even though the algorithm does a fair job of accurately identifying negative sentiment. Regarding Neutral emotion, the table shows that the model successfully predicted 2,958 out of 3,504 cases as Neutral. Nevertheless, 546 cases were incorrectly classified as Neutral, indicating that it can be difficult to discern Neutral attitude from other groups.

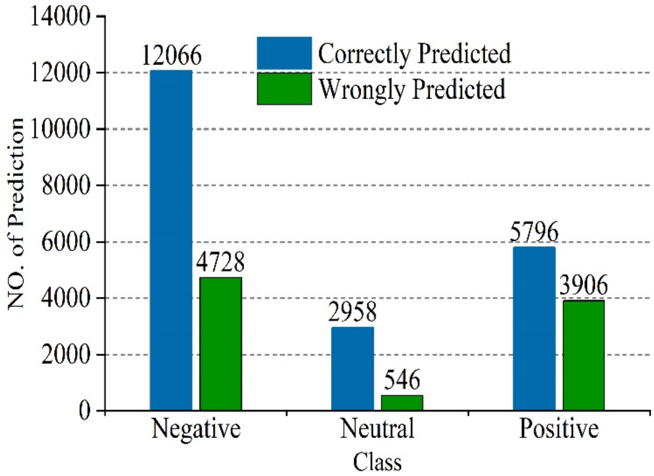

Figure 3. VADER- Model Predictions - Correct vs. Incorrect.

The table for positive sentiment indicates that the model successfully predicted 5,796 out of 9,702 cases as positive. 3,906 cases, on the other hand, were incorrectly forecasted as Positive, indicating that the model misclassified Positive sentiment in certain cases. Overall, Table 3 offers a thorough analysis of the model's predictions, showing both the areas in which the model correctly and incorrectly identified feelings. These observations are essential for assessing the model's efficacy and pinpointing areas in need of development, as they will direct future iterations of the sentiment analysis methodology to maximize precision and efficiency. High precision for negative and positive sentiments is shown in VADER and similar to that low precision for neutral sentiments, as demonstrated by the VADER model. Misclassifications between Negative and Neutral sentiments are particularly noticeable, despite the fact that the model catches instances of Neutral sentiments with high recall successfully. Insights on the model's performance, such as the confusion matrix and prediction breakdown, are vital for improving the sentiment analysis approach's accuracy and efficacy. These insights show the model's strengths and weaknesses.

6.1. SENTIWORDNET

SentiWordNet's class-wise performance matrices (Table 4 & Figure 4) provide the F1-score, precision, and recall metrics for the negative, neutral, and positive sentiment classes. SentiWordNet is notable for achieving excellent precision in all sentiment classes, with remarkable results in the categorization of positive sentiment. Although recall and F1-score measurements show variability among classes, precision is good, indicating possible areas for improvement. With an overall accuracy of 87%, the model is effective in accurately classifying sentiments.

Table 4. Class Wise Performance Matrices of SentiWordNet.

	Precision	**Recall**	**f1-score**	**Support**
Negative	98	83	90	16,794
Neutral	52	90	66	3,504
Positive	92	92	92	9,702
Accuracy	87			30,000
Macro avg	81	89	83	30,000
Weighted avg	91	87	88	30,000

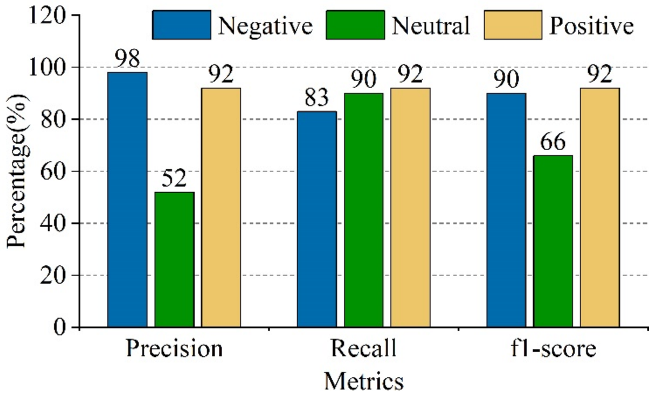

Figure 4. Class Wise SentiWordNet- Performance Matrices.

SentiWordNet's Confusion Matrix (Figure 5) offers a comprehensive analysis of how the model's predictions compare to the actual sentiment labels for every sentiment class. It draws attention to instances of both accurate and inaccurate classifications, highlighting both the model's strong points and possible misclassification areas. SentiWordNet's split of successfully and wrongly predicted instances for each sentiment class is shown in Table 5 and Figure 6. The model's accuracy in classifying feelings and the occurrence of misclassifications are displayed in the table. Interestingly, the model shows a high percentage of accurate predictions in all sentiment classifications, indicating its overall good performance.

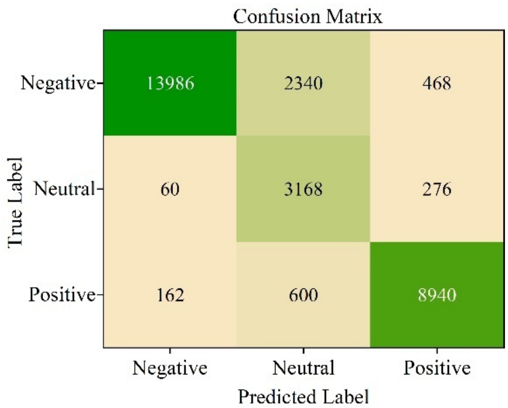

Figure 5. Confusion Matrix SentiWordNet.

SentiWordNet's Model Predictions are shown in Figure 6 and in Table 5, with accurate and inaccurate predictions for each sentiment class. This graphic gives a concise summary of the model's advantages and disadvantages for sentiment categorization, as well as insights into how well it performs. All things considered, the combination of these metrics and visualisations provides a thorough evaluation of SentiWordNet's sentiment analysis ability. They offer insightful information about the model's advantages—like its high precision—as well as its shortcomings—like its inconsistent recall and misclassification rates. These realisations are essential for improving the sentiment analysis methodology and raising the model's classification accuracy and efficacy.

Table 5. Class Wise Performance Matrices of SentiWordNet.

	Correctly Predicted	Wrongly Predicted
Negative	13,986	2,808
Neutral	3,168	336
Positive	8,940	792

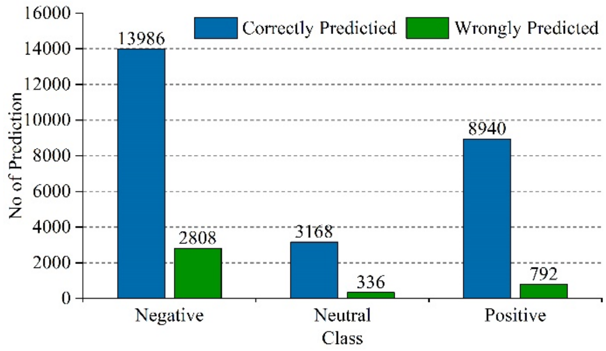

Figure 6. SentiWordNet—Model Predictions: Correct vs. Incorrect.

6.2. LinearSVC

The precision, recall, and F1-score metrics for each sentiment class (Negative, Neutral, and Positive) in the LinearSVC model are broken down in Table 6. Notably, the model does exceptionally well at classifying Negative feelings, achieving high precision in all sentiment classes. Recall and F1-score measurements, however, show considerable variation, suggesting possible areas for development. The model's total accuracy of 88% is given, indicating that it is effective at accurately classifying sentiments.

Table 6. Class Wise Performance Matrices of LinearSVC.

	Precision	Recall	F1-score	Support
Negative	97	84	90	5,646
Neutral	85	89	87	4,195
Positive	77	92	84	2,759
Accuracy	88			12,600
86	86	88	87	12,600
Weighted avg	89	88	88	12,600

A comparison analysis of the model's performance across several sentiment categories is provided by the Class Wise LinearSVC Performance Matrices in Figure 7, which graphically depict the precision, recall, and F1-score for each sentiment class. These matrices shed light on the sentiment categorization model's advantages and disadvantages. A thorough analysis of the discrepancies between the LinearSVC model's predictions and the actual sentiment labels for each sentiment class is provided by the Confusion Matrix, which is shown in Figure 8. It draws attention to instances of both accurate and inaccurate classifications, highlighting both the model's strong points and possible misclassification areas.

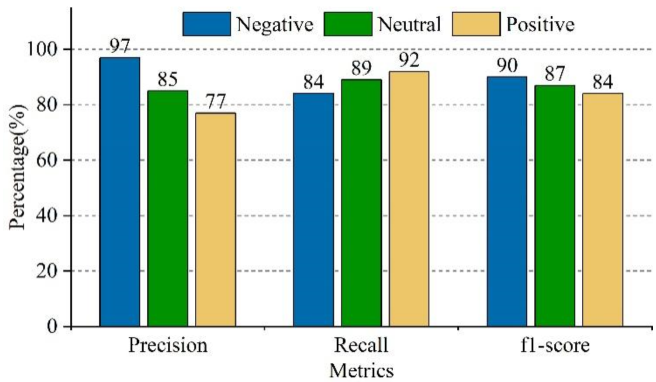

Figure 7. Class Wise LinearSVC - Performance Matrices.

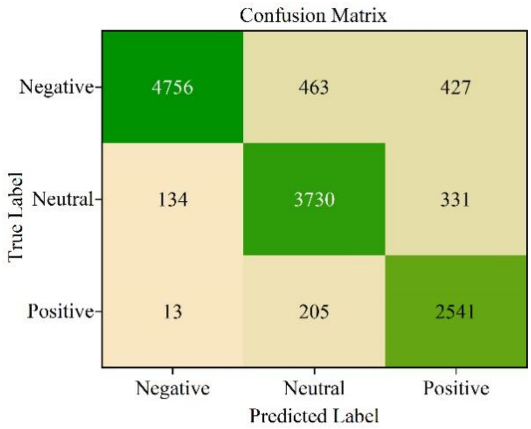

***Figure* 8.** ConfusionMatricsLinearSVC.

Table 7 provides more details on the model's predictions by breaking down the cases that the LinearSVC model predicted correctly and erroneously for each sentiment class. And Figure 8 provides the confusion matrix of SVC. This table provides information on the sentiment classification accuracy of the model by displaying examples of both accurate and inaccurate predictions.

Table 7. LinearSVC - Model Predictions: Correct vs. Incorrect.

	Correctly Predicted	**Wrongly Predicted**
Negative	4,756	890
Neutral	3,730	465
Positive	2,541	218

Table 7 breaks out the instances that the LinearSVC model properly and incorrectly predicted for each sentiment class to provide additional information on the model's predictions. This Table 7 and Figure 9 shows instances of both accurate and inaccurate predictions to give an idea of the model's accuracy in sentiment classification.

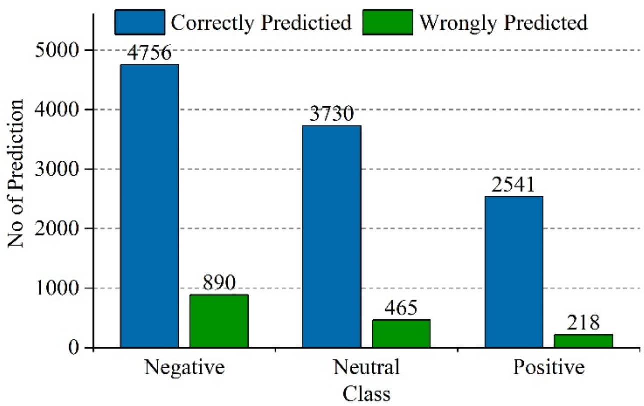

***Figure* 9.** LinearSVC - Model Predictions: Correct vs. Incorrect.

All together, these metrics and visualisations provide a thorough evaluation of the sentiment analysis performance of the LinearSVC model. They offer insightful information on the model's efficacy and accuracy in sentiment classification, directing future research and development to improve its performance.

6.3. Logistic Regression

The Logistic Regression model's precision, recall, and F1-score metrics are shown in Table 8 for each sentiment class (Negative, Neutral, and Positive). The model performs exceptionally well in classifying Negative feelings, with high accuracy across all sentiment classes. Recall and F1-score measurements, however, show considerable variation, suggesting possible areas for development. According to reports, the model's overall accuracy is 84%, demonstrating how well it can classify attitudes. The Class Wise Logistic Regression Performance Matrices are shown in Figure 10, which also provides a comparison of the precision, recall, and F1-score for each sentiment class. These matrices shed light on how well the model performs in various sentiment categories.

Figure 9's Confusion Matrix offers a thorough analysis of how each sentiment class's actual sentiment labels relate to the predictions made by the Logistic Regression model. It facilitates comprehension of the model's performance by highlighting examples of both accurate and inaccurate classifications.

Table 8.Class Wise Performance Matrices of Logistic Regression

	Precision	**Recall**	**F1-Score**	**Support**
Negative	92	83	87	5,646
Neutral	82	86	84	4,195
Positive	77	86	81	2,759
Accuracy	84			12,600
86	83	85	84	12,600
Weighted avg	85	84	85	12,600

A breakdown of the cases that the Logistic Regression model properly and incorrectly predicted for each sentiment class is shown in Figure 10, which expands on the model's predictions even more. This Table 8 provides information on the sentiment classification accuracy of the model by displaying examples of both accurate and inaccurate predictions. Lastly, the predictions of the Logistic Regression model are visualised in Figure 10, which contrasts accurate and inaccurate predictions for each sentiment class. Understanding the model's performance and pinpointing areas for development is made easier with the help of this visualisation. The Figure 11 is the confusion matrix for the model.

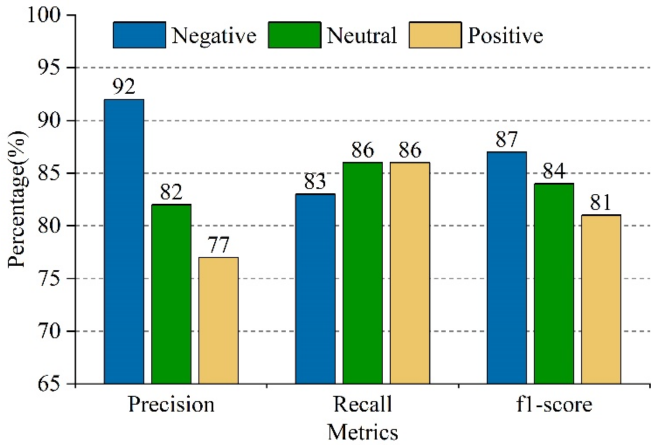

Figure 10. Class Wise Logistic Regression - Performance Matrices.

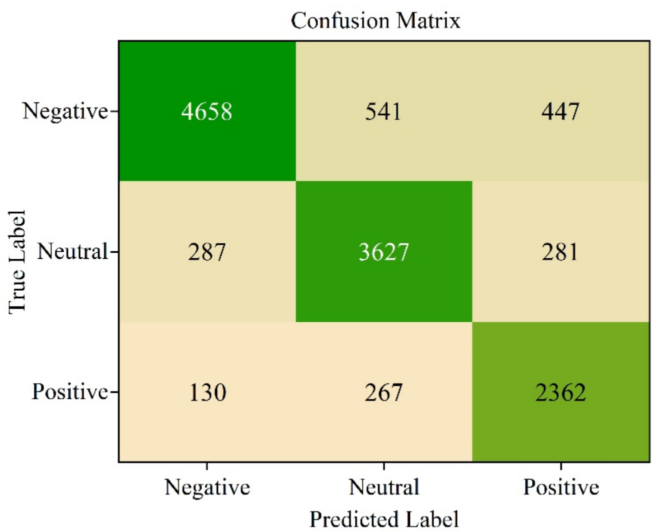

Figure 11. Confusion Matrix Logistic Regression.

In general, these metrics and visualisations provide a thorough evaluation of the Logistic Regression model's performance in sentiment analysis, including insightful information about how well it can identify sentiments. The correct prediction mad by the model is presented in Tables 9 and 10 with the pictorial representation in Figure 12. This model has 988, 568, 397 wrong prediction in negative, neutral and positive respectively. The wrong prediction made by this model is high compares to SAVA model.

Table 9. Logistic Regression Class wise correct Performance.

Class	Correctly Predicted
Negative	4,658
Neutral	3,627
Positive	2,362

Table 10. Logistic Regression - Model Predictions: Correct vs. Incorrect

Class	Correctly Predicted	Wrongly Predicted
Negative	4,658	988
Neutral	3,627	568
Positive	2,362	397

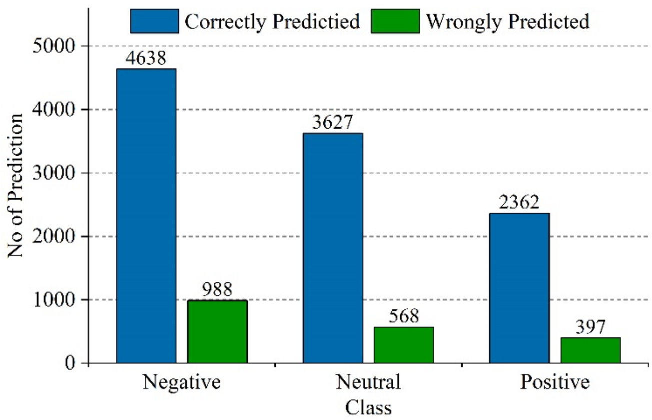

Figure 12. Logistic Regression - Model Predictions: Correct vs. Incorrect.

6.4. SVM

The precision, recall, and F1-score metrics for the Support Vector Machine (SVM) model's three sentiment classes—Negative, Neutral, and Positive—are shown in Table 11. In both negative and positive feelings, the model shows especially good precision across all sentiment classes. Nonetheless, there appears to be a notable disparity in the recollection of Neutral sentiment, indicating possible difficulties in accurately recognising examples of this category. The model's total accuracy—which has been claimed to be 90%—reflects its remarkable efficacy in sentiment classification.

Table 11. Class Wise Performance Matrices of SVM.

	Precision	Recall	f1-score	Support
Negative	96	92	94	16,440
Neutral	1	71.7	83.5	5,520
Positive	96.5	97	85.5	8,040
Accuracy	900			30,000
Macro avg	90	87	87.8	30,000
Weighted avg	91.6	90	90	30,000

The Class Wise SVM Performance Matrices are shown graphically in Figure 13, which also offers details on the F1-score, recall, and precision metrics for each sentiment class. These matrices make it easier to compare how well the model performs in various sentiment categories. For each sentiment class, the SVM model's predictions are compared to the actual sentiment labels in the Confusion Matrix, which is shown in Figure 14. It draws attention to instances of both accurate and inaccurate classifications, offering insightful information on the model's functionality.

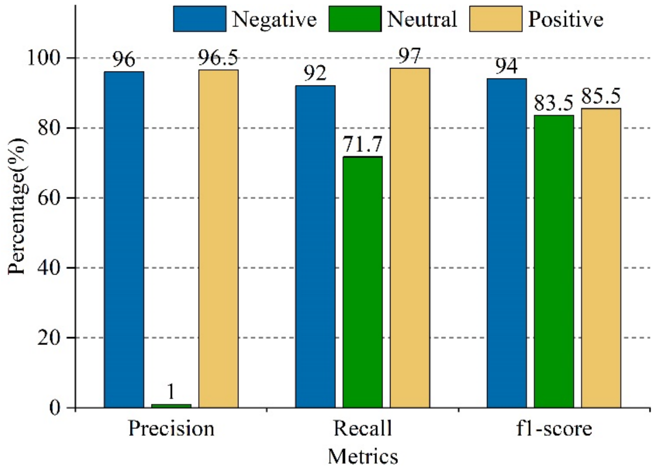

Figure 13. Class Wise SVM - Performance Matrices.

Table 12 provides a breakdown of the occurrences that the SVM model successfully and incorrectly predicted for each sentiment class, offering more insight into the model's predictions. This table provides information on the sentiment classification accuracy of the model by displaying examples of both accurate and inaccurate predictions.

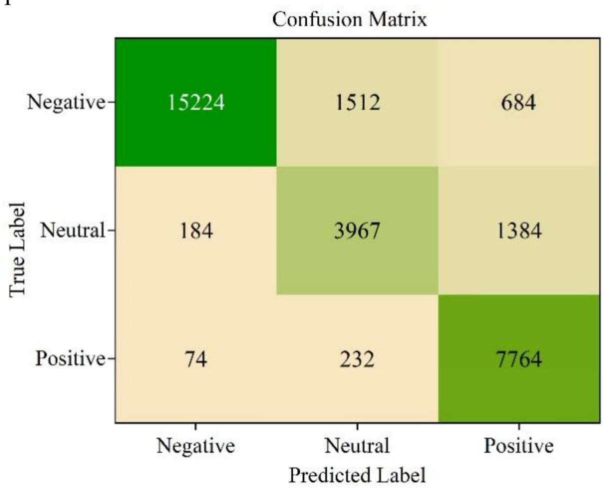

Figure 14. Confusion Matrix SVM.

Finally, the predictions made by the SVM model are shown in Figure 15 and confusion matrix in Figure 14, which contrasts accurate and inaccurate predictions for each sentiment class. Understanding the model's performance and pinpointing areas for development is made easier with the help of this visualisation. All things considered, these metrics and visualisations provide a thorough evaluation of the SVM model's performance in sentiment analysis, offering insightful information about how accurate and successful it is at categorising sentiments.

Table 12. SVM - Model Predictions: Correct vs. Incorrect.

	Correctly Predicted	**Wrongly Predicted**
Negative	15,244	2,196
Neutral	3,967	1,568
Positive	7,764	306

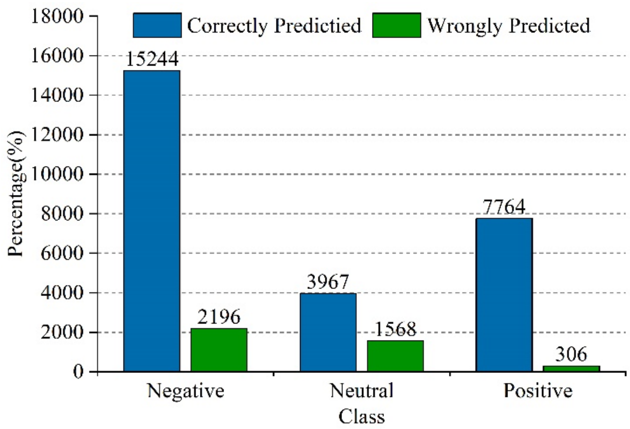

Figure *15.* SVM—Model Predictions: Correct vs. Incorrect.

6.5. SAVSA-SN7

The SAVSA-SN7 model's precision, recall, and F1-score metrics are shown in Table 13 for each sentiment class (Negative, Neutral, and Positive). The model performs remarkably well in sentiment classification, as evidenced by its astonishingly high precision, recall, and F1-score values across all sentiment classes. The model's claimed accuracy is at an astounding 99.7%, demonstrating its exceptional efficacy in accurately categorising emotions. The Class Wise SAVSA-SN7 Performance Matrices are shown graphically in Figure 16, which also offers details on the precision, recall, and F1-score metrics for each sentiment class. These matrices make it easier to compare how well the model performs in various sentiment categories. A thorough analysis of the discrepancies between the SAVSA-SN7 model's predictions and the actual sentiment labels for each sentiment class is provided by the Confusion Matrix, which is shown in Figure 17. It draws attention to instances of both accurate and inaccurate classifications, offering insightful information on the model's functionality. Table 13 provides an in-depth analysis of the predictions made by the SAVSA-SN7 model, breaking down the occurrences that it predicted correctly and erroneously for each sentiment class. This table provides information on the sentiment classification accuracy of the model by displaying examples of both accurate and inaccurate predictions.

Table 13. Class Wise Performance Matrices of SAVSA-SN7

	Precision	**Recall**	**F1-score**	**Support**
Negative	0.997	0.999	0.998	16,794
Neutral	0.996	0.994	0.995	3,504
Positive	0.997	0.994	0.996	9,502
Accuracy	99.7			29,800
Macro avg	0.997	0.996	0.996	29,800
Weighted avg	0.997	0.997	0.997	29,800

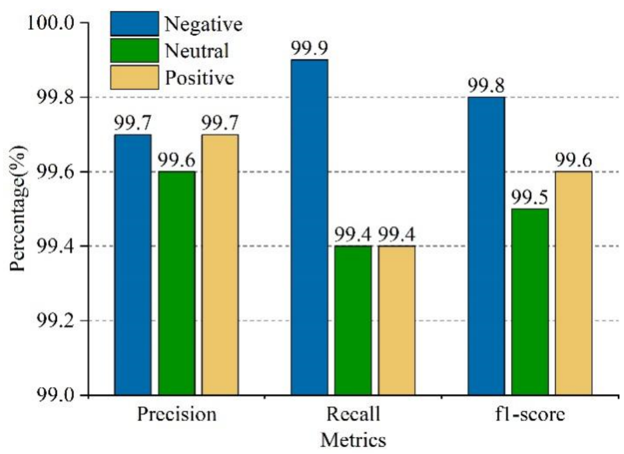

Figure 16. Class Wise SAVSA-SN7 - Performance Matrices.

Finally, the predictions made by the SAVSA-SN7 model are shown in Figure 16 and in Table 14 which contrasts accurate and inaccurate predictions for each sentiment class. Understanding the model's performance and pinpointing areas for development is made easier with the help of this visualisation. All things considered, these metrics and visualisations shown in how well the SAVSA-SN7 model performs in sentiment analysis and offer insightful information on how accurate and successful it is at identifying sentiments. The Figure 17 is the confusion matrix of the model.

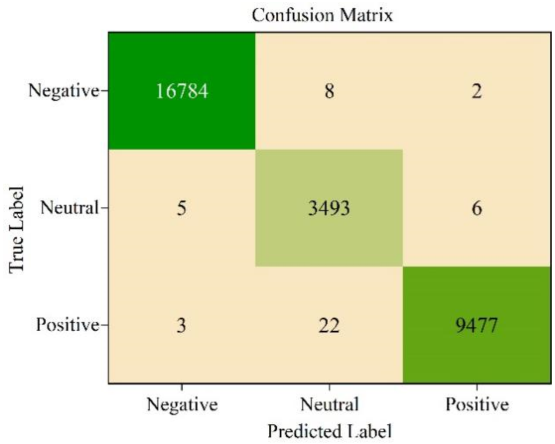

***Figure* 17.** Confusion Matrix SAVSA-SN7.

Table 14. *SAVSA*-SN7 - Model Predictions: Correct vs. Incorrect.

	Correctly Predicted	Wrongly Predicted
Negative	16,784	10
Neutral	3,493	11
Positive	9,477	25

Strong performance in sentiment classification was shown Figure 18 this is across a range of sentiment analysis models, such as SVM, Logistic Regression, LinearSVC, and SAVSA-SN7. Notable precision, recall, and F1-score metrics were also noted. With a nearly flawless accuracy of 99.7%, SAVSA-SN7 stands out and demonstrates how excellent it is at precisely classifying feelings. These findings highlight the developments in sentiment analysis methods and provide insightful information for practical uses in natural language processing jobs.

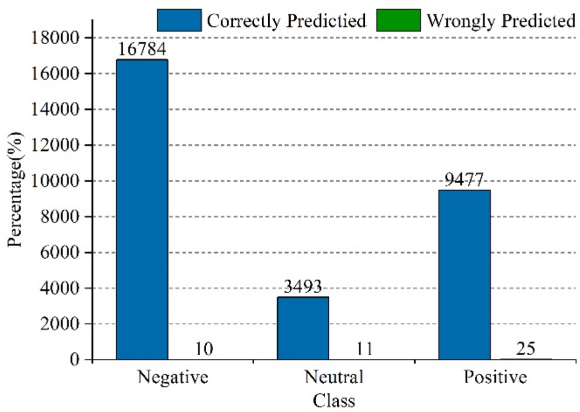

Figure 18. SAVSA-SN7 - Model Predictions: Correct vs. Incorrect.

7. Conclusions

SVM, Logistic Regression, LinearSVC, SAVSA-SN7, shown strong performance in analyzing sentiment. Notable metrics for F1-score, recall, and precision were also observed. SAVSA-SN7 stands out and proves how good it is at accurately classifying feelings with an almost perfect accuracy of 99.7%. These results demonstrate the advancements in sentiment analysis techniques and offer useful data for real-world applications in natural language processing. A detailed examination and comparison of several sentiment analysis models, such as SVM, Logistic Regression, Linear SVC, and SAVSA-SN7, sheds light on how well they work and how well they can distinguish between distinct classes of sentiments. The models exhibit discernible advantages and potential shortcomings, providing insight into the dynamic terrain of sentiment analysis techniques. SAVSA-SN7 performs exceptionally well among the models tested, displaying remarkable precision and recall scores that highlight its resilience in sentiment classification tasks. SAVSA-SN7 shows a remarkable capacity to reliably categories sentiments, with near-perfect precision levels and good recall rates across all sentiment classes. This has promising implications for real-world applications. These results indicate a move towards more sophisticated and trustworthy sentiment classification models in addition to highlighting the noteworthy advancements made in sentiment analysis methodologies. These developments have a lot of potential for a number of real-world uses, from customer sentiment tracking in corporate analytics to sentiment analysis in social media monitoring. Organizations may obtain more profound understanding of consumer attitudes, market trends, and brand impression by utilizing these state-of-the-art sentiment analysis models. This allows them to make data-driven decisions and more successfully customize their strategies. Furthermore, the conclusions drawn from this thorough assessment provide important direction for further sentiment analysis research projects. Researchers can concentrate on improving current procedures and creating novel approaches to solve the changing issues in sentiment categorization by evaluating the advantages and disadvantages of various models. The discipline of sentiment analysis has to be improved iteratively in order to continue to be relevant and useful in a variety of contexts. To sum up, the assessment of sentiment analysis models highlights the impressive advancements in this area, with SAVSA-SN7 standing out as a particularly strong candidate. These results not only confirm the efficacy of contemporary sentiment analysis methods but also stimulate additional research and development. Organizations can seize new chances to glean insightful information from textual data and facilitate well-informed decision-making in a world where data is driving decisions by utilizing sophisticated sentiment analysis algorithms.

Author Contributions

P.C.S.: Conceptualization, Data curation, Methodology, Software, Writing—original draft, Writing—review & editing. T.V.: Conceptualization, Methodology, Supervision, Writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding

The research received no external funding.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Data Availability Statement

The data used in this study is not currently available for external access as it is intended for further research by the authors.

References

1. Lyu, Joanne Chen, Eileen Le Han, and Garving K. Luli. "COVID-19 vaccine–related discussion on Twitter: topic modeling and sentiment analysis."*Journal of medical Internet research* Vol. 23, No. 6 pp. e24435, 2021, doi: 10.2196/24435
2. Bonnevie, Erika, Allison Gallegos-Jeffrey, Jaclyn Goldbarg, Brian Byrd, and Joseph Smyser. "Quantifying the rise of vaccine opposition on Twitter during the COVID-19 pandemic."*Journal of communication in healthcare,* Vol. 14, No. 1 pp. 12-19, 2021, https://doi.org/10.1080/17538068.2020.1858222
3. Kausar, Mohammad Abu, ArockiasamySoosaimanickam, and Mohammad Nasar. "Public sentiment analysis on Twitter data during COVID-19 outbreak." *International Journal of Advanced Computer Science and Applications*, Vol.12, No. 2 (2021).
4. Boon-Itt, Sakun, and YukolpatSkunkan. "Public perception of the COVID-19 pandemic on Twitter: sentiment analysis and topic modeling study."*JMIRPublic Health and Surveillance*, Vol 6, No. 4, pp. e21978, (2020).
5. Pano, Toni, and Rasha Kashef, "A complete VADER-based sentiment analysis of bitcoin (BTC) tweets during the era of COVID-19."*Big Data and Cognitive Computing* Vol.4, No. 4, pp. 33, (2023).
6. Flores-Ruiz, David, Adolfo Elizondo-Salto, and María de la O. Barroso-González. "Using social media in tourist sentiment analysis: A case study of Andalusia during the COVID-19 pandemic." *Sustainability*, Vol.13, No. 7, pp. 3836, (2021).
7. Zainuddin, Nurulhuda, Ali Selamat, and Roliana Ibrahim. "Hybrid sentiment classification on twitter aspect-based sentiment analysis." Applied *Intelligence,* Vol. 48 pp.1218-1232, (2018).
8. Trupthi, M., Suresh Pabboju, and G. Narasimha. "Sentiment analysis on twitter using streaming API." *IEEE 7th international advance computing conference (IACC)*, pp. 915-919. (2017).
9. Sridevi, P. C., and T. Velmurugan. "Impact of Preprocessing on Twitter Based Covid-19 Vaccination Text Data by Classification Techniques."*International Conference on Applied Artificial Intelligence and Computing (ICAAIC),* pp. 1126-1132. IEEE, (2022).

10. Wendland, André, Marco Zenere, and Jörg Niemann. "Introduction to text classification: impact of stemming and comparing TF-IDF and count vectorization as feature extraction technique."*Systems, Software and Services Process Improvement: 28th European Conference, EuroSPI*Volume 28, pp. 289-300. Springer International Publishing, 2021. https://doi.org/10.1007/978-3-030-85521-5_19
11. Hoque, Mohammad Mahmudul. "An analytical approach to analyze the popular word search from nineteen-year news dataset using Natural language processing technique." 2023, https://urn.fi/URN:NBN:fi:amk-2023062124248
12. Kaur, Ravneet, Ayush Majumdar, Priya Sharma, and Bhavana Tiple. "Analysis of Tweets with Emoticons for Sentiment Detection Using Classification Techniques." *International Conference on Distributed Computing* and Intelligent Technology, pp. 208-223. Cham: Springer Nature Switzerland, 2023. https://doi.org/10.1007/978-3-031-24848-1_15
13. Prusa, Joseph, Taghi M. Khoshgoftaar, David J. Dittman, and Amri Napolitano, "*Using random undersampling to alleviate class imbalance on tweet sentiment data*" IEEE international conference on information reuse and integration, pp. 197-202, 2015, 10.1109/IRI.2015.39.
14. Elamrani Abou Elassad, Zouhair, Mohammed Ameksa, DauhaElamrani Abou Elassad, and Hajar Mousannif. "*Efficient Fusion Decision System for Predicting Road Crash Events: A Comparative Simulator Study for Imbalance Class Handling*"**,** Transportation Research Record, 2023 03611981231192985.
15. P.C. SRIDEVI, T. Velmurugan, Enhancing Sentiment Analysis of User Response for COVID-19 Vaccinations tweets Using Senti WordNet-Adjusted VADER Sentiment Analysis (SAVSA): A Hybrid Approach, ISDA 2023
16. Boudali, Imen. "Data-Driven Prediction of Complications Risks in Cancer Patients: Machine Learning based approach." International Journal of Computer Information Systems & Industrial Management Applications15, Vol.12, pp 17-188, 2023.
17. Srihari, P., and J. Harikiran. "A Deep Learning Based Hybrid Approach for Human Physical Activity Recognition in Thermal Imaging." International Journal of Computer Information Systems and Industrial Management Applications Vol. 14, pp. 110-118.Isnan, Mahmud, Gregorius Natanael Elwirehardja, and Bens Pardamean. "Sentiment Analysis for TikTok Review Using VADER Sentiment and SVM Model." *Computer Science*, Vol. 227 2023, pp. 168-175, https://doi.org/10.1016/j.procs.2023.10.514.
18. Velmurugan, T. & Chokkalingam, S. P. (2024). Analysing Domain-Specific Sentiments in Healthcare: SentiWordNet Adjusted Vader Sentiment Analysis (SAVSA), *Journal of Computer Science*, Vol. 20, No. 3, pp. 239-253. https://doi.org/10.3844/jcssp.2024.239.253
19. Habbat, Nassera, Hicham Nouri, Houda Anoun, and Larbi Hassouni. "Sentiment analysis of imbalanced datasets using BERT and ensemble stacking for deep learning." *Engineering Applications of Artificial* Intelligence, Vol. 126, pp. 106999, 2023.
20. Chouhan, Kuldeep, Mukesh Yadav, Ranjeet Kumar Rout, Kshira Sagar Sahoo, N. Z. Jhanjhi, Mehedi Masud, and Sultan Aljahdali. "Sentiment Analysis with Tweets Behaviour in Twitter Streaming API." *Comput. Syst. Sci. Eng*, Vol. 4, No. 2, pp. 1113-1128, 2023, https://doi.org/10.32604/csse.2023.030842
21. Gopi, Arepalli Peda, R. Naga Sravana Jyothi, V. Lakshman Narayana, and K. Satya Sandeep. "Classification of tweets data based on polarity using improved RBF kernel of SVM."*International Journal of Information Technology*, Vol. 15, No. 2, pp. 965-980, 2023.
22. Huq, Mohammad Rezwanul, Ali Ahmad, and Anika Rahman, "*Sentiment analysis on Twitter data using KNN and SVM*", International Journal of Advanced Computer Science and Applications 8, no. 6 (2017).
23. Raheja, Supriya, and Anjani Asthana. "Sentiment Analysis of Tweets During the COVID-19 Pandemic Using Multinomial Logistic Regression." *International Journal of Software Innovation (IJSI)*, Vol.11, No. 1, pp. 1-16, 2023, DOI: 10.4018/IJSI.315740.