ISSN 2723-6471

Sentimental Analysis of Legal Aid Services: A Machine Learning Approach

Joe Khosa[1,*], Daniel Mashao[2], Ayorinde Olanipekun[3]

[1]*Department of Engineering Management, Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa*

[2]*Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa*

[3]*Data Science Across Disciplines Research Group, Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa*

(Received: October 30, 2024; Revised: November 22, 2024; Accepted: December 18, 2024; Available online: February 21, 2025)

Abstract

Legal Aid services in South Africa, administered by Legal Aid South Africa (SA), aim to provide essential legal representation to vulnerable individuals lacking financial resources. Despite its significant role, there is a pervasive perception among the public that the quality of these state-funded services is substandard, often leading to negative attitudes towards the organization. This research employs sentiment analysis to evaluate client perceptions of Legal Aid SA's services, using a dataset of 5,246 entries from Twitter and the Internal client feedback system between 2019 and 2024. The study utilizes various machine learning algorithms, including Naive Bayes, Stochastic Gradient Descent (SGD), Random Forest, Support Vector Classification (SVC), Logistic Regression, and Extreme Gradient Boosting (XGBoost), to analyze sentiment polarity and classify feedback into positive, neutral, and negative sentiments. The accuracy, precision, recall, and F1 scores assessed model performance. The SVC and XGBoost models demonstrated superior performance, achieving testing accuracies of 90.10% and 90.00%, respectively. In contrast, Naive Bayes and Logistic Regression lagged, with test accuracies of 82.00% and 85.00%, respectively. The findings reveal that most responses are either neutral or positive, suggesting a predominantly favorable impression of Legal Aid services. This research not only aims to enhance Legal Aid SA's service offerings but may also provide valuable insights for similar organizations globally.

Keywords: Legal Proceedings, Legal Outcomes, Artificial Intelligence, Machine Learning Algorithms, Legal Judgments, Classification Performance, Legal Aid SA, Legal Aid Services

1. Introduction

Legal Aid services refer to providing legal aid or counsel by governments, their agencies, or non-government organizations to individuals who lack the financial means to hire a lawyer and require court representation or legal expertise to address legal aid matters. In South Africa, Legal Aid South Africa (SA) is a National Public Entity with Constitutional and statutory authority to offer legal assistance to individuals who are in need and vulnerable. The institution embodies operational principles of dedication to professionalism and the pursuit of exceptional service [1]. Legal Aid SA is classified as Schedule 3A under the Public Finance Management Act, 1999 (Act No. 1 of 1999) of South Africa [2]. The primary objective of Legal Aid SA, as stated in the preamble of the Act, is to guarantee the availability of legal representation and to fulfil individuals' right to access justice as outlined in the South African constitution, including providing or facilitating access to legal assistance and legal advice.

State subsidies for Legal Aid services make the public perceive these services as free. Consequently, clients tend to link free services with substandard quality. Legal Aid SA is globally recognized as having the most significant annual Candidate Attorneys (CA) intake. CA's are recently graduated attorneys still establishing themselves in the legal profession. The clients often exhibit a pessimistic attitude or unfavorable opinion of the services. Clients believe that Legal Aid SA attorneys would recommend that clients plead guilty to avoid legal proceedings. Negative opinions of Legal Aid services may significantly impact clients' trust and readiness to engage with the organization. Refrain from

*Corresponding author: Joe Khosa (joe.khosa@gmail.com)
DOI: https://doi.org/10.47738/jads.v6i2.521

satisfying service quality or delays may deter clients from pursuing additional aid or recommending the services, thus compromising the organization's aim to ensure equitable access to justice. It is essential to keep clients informed about the status of their cases to mitigate against this problem. Given the lengthy duration of court proceedings, clients must be provided with regular information on the development of their cases. Conversely, customers should assist the organization in enhancing its service offerings by sharing feedback on the quality of service received, whether positive or negative.

In this research, data obtained from an internal client feedback system for Legal Aid SA and Twitter is analyzed using Sentimental analysis to evaluate clients' views of Legal Aid Services. We analyze a dataset covering five years, from 2019 to 2024. This system records both criminal and civil cases. The research will enable the organization to enhance its customer service by either reengineering business processes or using automation in some regions of the business processes. Furthermore, this research may also be advantageous for other Legal Aid services around the globe.

Section 2 discusses the basic concept of aspect-based Sentiment Analysis. Section 3 states the relevant material and methodology used in this study. Section 4 explains our machine learning results and discussion. Section 5 covers the conclusion of the study.

2. Related Study

Sentiment analysis is analyzing natural language to identify emotions associated with a text. Sentiment analysis monitors consumer opinion on social media and brand campaign monitoring [1]. It is also referred to as emotional sentiment, which [2] defines the analysis of the application of text analysis techniques in conjunction with natural language processing technologies to examine emotions. Sentiment analysis incorporates data from several sources to ascertain the user's attitude across several dimensions. It is extensively used to extract views and identify sentiments, enabling business organizations to comprehend user requirements [3].

[4] implemented aspect-based Sentiment Analysis on argument-based legal documents concerning Indian domestic violence cases to support the increasing number of cases and alleviate the burden on legal professionals handling these matters. They successfully applied the model to the data, predicting the offender's outcome during the COVID-19 pandemic. In another study, Bola [5] applied the sentimental analysis to improve the effectiveness of the Canadian court system; they introduced the machine learning module that was developed using LSTM and CNN algorithms; in their conclusion, they emphasized the use of Machine Learning Sentimental Analysis as a valuable tool for the future of court systems.

Sentimental analysis has also been applied in the film industry to recommend the best films. Einblick [6] found that users' teams of comparable interest were, through the tendency, submitted actively by users, so it projected a sentiment-aware cooperative filtering technique. When Facebook rebranded to Meta, a study by Hasgül [7] sought to determine the Indonesian people's response to the Metaverse. The study collected data from Twitter tweets with keywords or Metaverse queries and used a Support Vector Machine Algorithm with TF-IDF word weighting; the tests were carried out with the results of the class division of positive sentiment 70.69%, neutral 15.85%, negative 13.46%.

Much research has been done on Sentiment Analysis to increase the accuracy of sentiment analysis systems, ranging from basic linear models to complicated neural network models. Other models or algorithms were checked previously, but these took longer to operate and needed higher overall accuracy scores. Mohammed's study [8] uses the BERT model, which is pre-trained on a vast corpus. The model is offered to address concerns with sentiment analysis systems.

In this paper, we use sentiment analysis to assess the services provided by Lega Aid South Africa and address the perception that South African state-funded lawyers provide poor services. The study's results will help to improve the quality of service within the judiciary services in South Africa.

3. Material and methods

3.1. Data Collection

Crawling or data collection refers to the collection of datasets [6]. The data for this study was obtained from Twitter queries using the Legal Aid query. Using Twitter to connect to the Twitter Application Programming Interface (API)

to access required keys and tokens. The crawling method has been used to obtain tweet data from Legal Aid SA for five years, from 2019 to 2024; there has been limited data as only an average of 20 records were recorded annually, resulting in about 100 data entries for the entire 5-year period, deemed minimal for the required analysis. The limited data from social media could be attributed to the nature of Legal Aid SA's clients: indigent clients with minimal access to internet services and smart devices. Although this is the case, it is essential to note that the Twitter platform does not represent most Legal Aid SA clients. Specific demographics predominantly use social media, and its user base may not reflect the broader population. An internal system records clients' feedback for the organization. We collected 5246 data entries from 2019 through 2024. Most clients prefer to call in, as illustrated in figure 1 below. A call Centre agent records these clients' feedback and routes them to relevant parties for attendance.

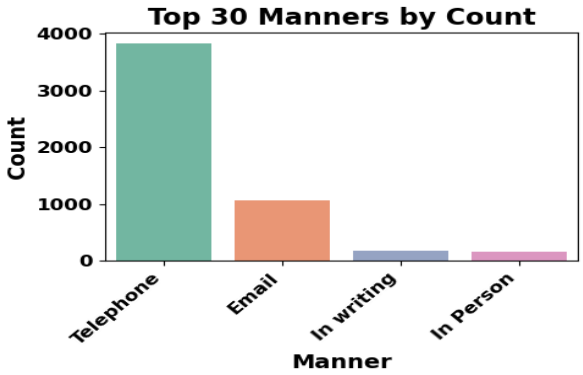

Figure 1. Client feedback category

3.2. Text Pre-Processing

Text Pre-processing is a process to improve text quality or selection of text data to eliminate noise [10]. Data becomes more structured and uniform through the following stages of the process. The process flow below in figure 2 illustrates the steps we follow to process our data.

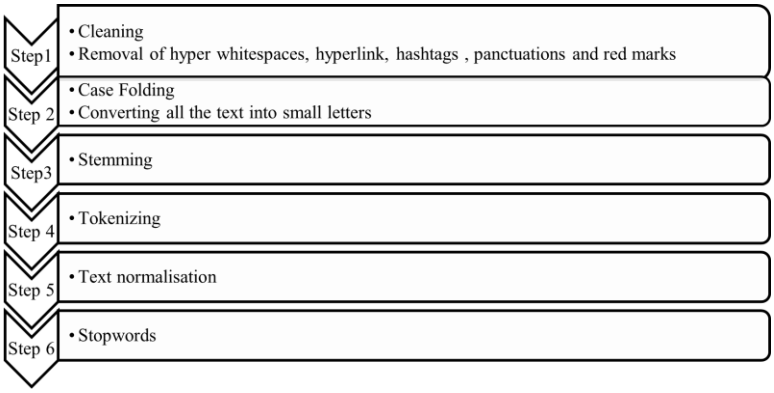

Figure 2. Data cleaning process

3.3. Stemming and Lemmatization

Stemming and lemmatization are language modelling techniques that improve document retrieval results [9]. Stemming can improve recall, but it can also hurt precision as words with distinct meanings may be fused to the same form (such as "army" and "arm"), and these mistakes are costly when performing sentence retrieval. In simple terms, Stemming refers to the process of removing prefixes and suffixes from words. Applying stemming algorithms reduces words to their root, allowing documents to be represented by the stems of words instead of original words. Lemmatization is an essential pre-processing step for many text-mining applications and is also used in natural language processing.

Lemmatization is similar to stemming as both reduce a word variant to its "stem" and its "lemma" in lemmatizing. It uses vocabulary and morphological analysis to return words to their dictionary form. In the English language, lemmatization and stemming often produce the same results. Sometimes the normalized/basic form of the word may be different than the stem e.g. "computes", "computing", "computed " is stemmed to "comput", but the lemma of that

words is "compute" [10], [11]. Stemming and lemmatization have an important role in increasing recall capabilities. Using Lemmatization and stemming in our study allowed us to compare their effectiveness. It provided insights into whether stemming, which focuses on computational efficiency, or lemmatization, which focuses on linguistic accuracy, significantly impacts the results.

3.4. Tokenization

According to Khurana [12], tokenization within natural language processing (NLP) is commonly understood as dividing a continuous sequence of characters into individual units of meaning, typically words. Frequently, it is correlated with processes at either the lower or higher level. Despite the common tendency to categorize both tasks as "pre-processing," it is essential to note that tokenization is distinct from initial "cleaning procedures," such as eliminating extraneous tags, removing non-textual elements, and excluding elements that do not pertain to natural languages, such as mathematical or chemical formulas and programs.

3.5. Stopwords

Stopwords are words that are frequently used but do not carry any significant meaning, such as "the," "a," "an," "in," and so on. The terms were removed from the data entries as they do not provide meaningful value to the analysis [13].

3.6. Word cloud

A word cloud is a computerized representation of the text in a document collection. The frequency of a keyword in the analyzed material determines the size of the word displayed in the image. The word cloud will be generated once the pre-processing phase is finished [14].

3.7. Labeling Sentiment

Labelling sentiment is the process of giving to a class based on the characteristics or characteristics contained in the document or sentence. The research on sentiment labelling involves categorizing the class into three distinct sentiment classes: Positive, Neutral, and Negative. A sentence with a value more than 0 is defined as positive. When a sentence has a value equal to 0, it is classified as neutral, and finally, when a phrase has a value less than 0, it is classified as negative [15].

In this study, we have chosen TextBlob as a package. This package is acknowledged and extensively employed in the domain of sentiment analysis. Our selection criteria centred on packages and libraries that offer a continuous value to indicate sentiment intensity. TextBlob is supplied with a feature for calculating sentiment ratings or classifying polarity. The TextBlob library employs the TextBlob() function, which accepts text as input and generates a blob object as output. This object can be utilized to segment text into words and phrases. The sentiment intensity can be obtained through the sentiment polarity attribute of this object. The resultant values lie within the interval of [−1, 1]. Figure 3 below is a snippet of the Python code used to label sentiments [16].

```
from textblob import TextBlob
def get_sentiment(text):
    blob = TextBlob(text)
    sentiment_polarity = blob.sentiment.polarity
    sentiment_subjectivity = blob.sentiment.subjectivity
    if sentiment_polarity > 0:
        sentiment_label = 'Positive'
    elif sentiment_polarity < 0:
        sentiment_label = 'Negative'
    else:
        sentiment_label = 'Neutral'
    result = {'polarity':sentiment_polarity,
              'subjectivity':sentiment_subjectivity,
              'sentiment':sentiment_label}
return result
```

Figure 3. Python code used to label sentiments

The get_sentiment function evaluates the sentiment of a specified text utilizing the TextBlob library. The process is initiated by generating a TextBlob object from the input text, facilitating access to TextBlob's sentiment analysis capabilities. The function identifies two essential sentiment attributes: polarity, which quantifies the text's tone on a

 ISSN 2723-6471

scale from -1 (very negative) to 1 (extremely positive), and subjectivity, which evaluates the degree of opinion in the text, ranging from 0 (entirely objective) to 1 (entirely subjective). The function assigns a sentiment label based on the polarity score: "Positive" if the score exceeds 0, "Negative" if it is below 0, or "Neutral" if it equals 0. Subsequently, it compiles these outcomes into a dictionary that includes polarity, subjectivity, and sentiment labels, which is then returned to the user. For instance, the analysis of the text "I love this product, it's amazing!" would yield a polarity of 0.85, a subjectivity of 0.75, and a sentiment classification of "Positive," signifying a very positive and subjective assertion.

3.8. Weighing TF-IDF

The fifth stage of frequency-inverse document frequency (TF-IDF) combines two TF and IDF processes. The IDF measures how important a word is in a document to determine the number of words that often appear in a sentence or language. Here is the TF-IDF weighting formula [17].

$$idf(t, D) = \log \frac{N}{|\{d : t_i \in d\}|} \quad (1)$$

3.9. Support Vector Machine Classification

The SVM classification algorithm is used to classify linear and non-linear data, specifically to tackle non-linear problems by employing kernel concepts to convert the data into higher dimensional spaces. Classification is utilized to classify entities based on new data. Supervised models involve the procedure of categorization. Regarding categorizing data samples and forecasting several pre-existing classes using pre-existing samples. The linear kernel is commonly employed for classifying data that does not have a linear classification. The reason for this is that its kernel operations are simple, and it may be used for text classification. The Confusion Matrix assessment is a testing phase that computes and generates evaluation matrices, including accuracy, precision, recall, and F1-score. The matrices are produced when labelling and categorizing the data testing method using the Support Vector Machine [18].

3.10. Adopted Machine Learning Algorithms

After vectorizing the data, the next step will be to develop deep learning machine learning algorithms to analyze the classification performance of the datasets. Since the sentiment analysis task is usually modelled as a classification problem, machine learning algorithms that can solve classification problems were used. Different machine learning algorithms were used to measure the performance of sentimental analysis, viz: Extreme Gradient Boosting, Support Vector Machine, Naive Bayes, Stochastic Gradient Descent, Random Forest, and Logistic Regression. The data is first split into 80% training and 20% test set followed by machine learning analysis, involving model evaluation and statistical analysis.

3.10.1.Extreme gradient boosting (XGBoost)

An open-source implementation of gradient-boosted trees is called XGBoost. It falls under the supervised algorithm, which accurately predicts by integrating the estimate of simpler and weaker models [19]. Wu et al. define XGBoost as a typical decision tree ensemble-based model. It is optimized from GBDT, which introduced second-order derivatives into the optimization process. Wu et al. [20] suggested that the XGBoost algorithm, based on the GBDT structure, is known for its outstanding results in Kaggle's ML competitions. Unlike GBDT, the XGBoost goal function includes a regularization term to avoid overfitting. The main objective function is described as follows:

$$O = \sum_{i=1}^{n} L(y_i, F(x_i)) + \sum_{k=1}^{t} R(f_k) + C \quad (2)$$

R(fk) represents the regularization term at iteration k, and C is a constant that can be removed selectively. Regularization term R(fk) written as,

$$R(f_k) = \alpha H + \frac{1}{2}\eta \sum_{j=1}^{H} w_j \quad (3)$$

where α is the complexity of leaves, H denotes the number of leaves, η signifies the penalty variable, and ωj represents output results in each leaf node. Leaves denote the expected categories based on classification criteria, whereas the leaf node denotes the tree node, which cannot be divided.

3.10.2. Random Forest

The Random Forest algorithm is a well-known supervised machine learning technique widely utilized for addressing classification and regression tasks within machine learning. It is widely understood that a forest is composed of many trees and that its resilience is positively correlated with the number of trees present [21]. A random forest algorithm is a classification technique that uses data presented to it to produce multiple decision trees. Its accuracy and problem-solving ability increase proportionally with the number of trees it contains. Averaging techniques increase the model accuracy based on the predicted results [22]. The figure below is a graphical depiction of the random forest algorithm from the sample dataset. After applying the grid search for the hyperparameter tunning optimization using max_depth: [8,10,12,14], max_features: [60,70,80,90,100], min_samples_leaf: [2, 3, 4], and n_estimators: [100, 200, 300], the best hyperparameters for Random forest Classifier are 0.881, and grid_search:{'CV': 3, 'n_jobs': '-1', 'verbose': '2'}.

3.10.3. Stochastic Gradient Descent (SGD)

The stochastic gradient descent (SGD) falls under the supervised machine learning algorithm. It is known for its robustness in building a predictive model [23]. The SDG algorithm reduces the cost of computation while also having a faster convergence rate. Meanwhile, as the amount of data increases, the timing for the weight update increases. Steps taken for SDG computation are as follows: individual weight and gradient computation and weight update. The gradient of an instance i can be calculated as $\nabla E(W_t, x_i, y_i)$ selected randomly at iteration t. While x_i represents a given data instance, having y_i, with weight vector W_t [24]. Unlike Batch gradient descent (BDG), it fluctuates continuously to converge. Assisting W_t to accelerate towards a better non-convex error function local minimal [25]. The algorithm begins to learn again whenever the termination criteria are reached after reaching a maximum number of iterations. The SDG classifier used the following parameter for the hyperparameter tuning: Loss='hinge', penalty='12', random_state=0. Scikit-learn(Sklearn) machine learning library in Python was used to import the SDG model.

3.10.4. Logistic Regression

Logistic regression techniques are usually used to explain the connection between output and input variables. They can be used to make discrete predictions with true or false results and predictions for continuous values. The input and output variables are usually independent and dependent [26]. However, the probability is calculated by the logistic sigmoid function or logistic function. Using an S-shaped curve, the logistic function transforms the input into a number between 0 and 1 [27].

$$\mathrm{l} = \log_{\mathrm{b}} \frac{\mathrm{p}}{1-\mathrm{p}} = \beta_0 + \beta_1\chi_1 + \beta_2\chi_2 + \cdots \beta_\mathrm{n}\chi_\mathrm{n} \quad (4)$$

β_0 represents the y-intercept, p denotes the estimated probability, the coefficient of χ_1 is β_1, and the coefficient of β_1 is χ_1. The grid search method was used for the hyperparameter tuning. Working with different machine learning algorithms or classifiers made it impossible to know the best parameters to yield the best prediction. It might be time-wasting to search for the best parameter combination manually. Hence, using grid search becomes necessary. Therefore, Grid search allows for the specification of ranges of values for the tuning parameters, while the hard work is left for the classifier to execute different permutations automatically. The permutations will figure out the best parameter combinations. This applies to other classifiers, which will be discussed later.

The models and Grid search parameters used for the logistic Regression in Python programming implementation are defined below in figure 4:

```
Model=LogsiticRegresion()
Parameters:
solvers = ['netwon − cg', 'lbfgs', 'liblinear']
penalty = ['l2']
C_values =[100, 10, 1.0, 0.1, 0.01, 0.001]
```

Figure 4. Models and Grid search parameters

3.10.5. Naive Bayes

Naïve Bayes uses theorem and Bayes to build classifiers. The Theorem describes the tendency of an event to happen due to different situations associated with that event. Naïve Bayes build classification model based on Baye's Theorem.

The Naïve Bayes classifier is built by allotting class labels problem instances, represented by vectors' features. The assumption is that any feature value is independent of other feature values. This assumption can be referred to as an independence assumption. In Naïve Bayes, when presented with a class of observation named Y, the probability of X belonging to Y is given by the equation below [28].

$$P(Y\backslash X) = \frac{P(Y)P(X\backslash Y)}{P(X)} \quad (5)$$

In this classification, the Multinomial Naïve Bayes algorithm approach was used. Multinomial is popular in NLP and is also known to be the most improved version of the Naive Bayes classifier. It assumes the tag of a text using Bayes theorem [29], [30]. Scikit-learn(Sklearn) machine learning library in Python was used to import the Multinomial Naïve Bayes model.

4. Results and Discussion

4.1. Data Collection

Twitter data as a resource for sentiment analysis offers a novel viewpoint on public perceptions of Legal Aid South Africa's services. Nonetheless, the constraints of this dataset require careful evaluation to provide a balanced and precise interpretation of the results. The Twitter dataset comprised 100 data points gathered over five years, in contrast to the 5,246 internal client feedback system records. This discrepancy put into question the representativeness of the Twitter data. Clients of Legal Aid South Africa are primarily underprivileged and vulnerable, lacking access to social media platforms such as Twitter. Twitter users are mainly from urbanized areas, an economically stable segment of the South African population. This results in a demographic bias, omitting clients' perspectives from rural or economically disadvantaged areas of South Africa, representing a substantial segment of Legal Aid SA's clientele. Legal Aid SA's annual performance reports indicate that they represent over 500,000 clients annually. In this study, 5246 records were analyzed from an internal client feedback system. Figure 5 below provides a snapshot of the data.

Date	Details	RemedialSteps	Evaluation	DateCreated	Status	DateFinalised	RegOffJC
2019-07-31 00:00:00	client complained that he had had no feedback ...	We are correspondents for this matter from Que...	Investigated and Resolved	2019-08-05 09:40:45.763	Finalised	NaN	Grahamstown
2020-10-09 00:00:00	Client indicated that he instructed Adv Rakosa...	none	Investigated and Resolved	2020-10-09 10:30:41.780	Finalised	2020-10-12 00:00:00	Lichtenburg
2020-10-19 00:00:00	Client complaint about a divorce matter that i...	Pension money was frozen & client updated re h...	Investigated and Resolved	2020-10-19 10:05:23.120	Finalised	2021-03-02 00:00:00	Polokwane
2020-06-22 00:00:00	Head of the Family Advocate's Office feels tha...	Meeting held by HOO with Ms Cader-Begg and a r...	Investigated and Resolved	2020-07-17 12:13:36.623	Finalised	2020-07-01 00:00:00	Uitenhage
2021-02-08 00:00:00	client is saying legal aid delayed his case re...	The file unde ref number X689095017 was opened...	Unfounded Complaint	2021-02-08 08:42:28.370	Finalised	2021-02-18 00:00:00	Advice Line

Figure 5. Data representation.

4.2. Text Pre-Processing

The results of data obtained from the complaint system are shown in figure 6.

ISSN 2723-6471

	Before	After
0	client complained that he had had no feedback ...	client complained that he had had no feedback ...
1	Client indicated that he instructed Adv Rakosa...	Client indicated that he instructed Adv Rakosa...
2	Client complaint about a divorce matter that i...	Client complaint about a divorce matter that i...
3	Head of the Family Advocate's Office feels tha...	Head of the Family Advocate's Office feels tha...
4	client is saying legal aid delayed his case re...	client is saying legal aid delayed his case re...
...	...	...
5241	Client alleges that she is struggling to get h...	Client alleges that she is struggling to get h...
5242	The legal representative of the other party wa...	The legal representative of the other party wa...
5243	Progress in his matter.	Progress in his matter.
5244	client complain that he attorney does not appe...	client complain that he attorney does not appe...
5245	Client advises that he opened up a file in 201...	Client advises that he opened up a file in 201...

5246 rows × 2 columns

Figure 6. Data Processing

The result of Text pre-processing is shown in figure 6 above. Data sets that have already passed the pre-processing phase are uniform and structured, and noise in the text is lost, so the classification phase is more optimum for calculation at the time of entry.

4.3. Word Cloud

Word Cloud Visualization visualizes data sets to determine what often appears in documents. WordCloud processes use Python as a programming language with the help of the matplotlib library. The following Word Cloud visualization can be seen in figure 7.

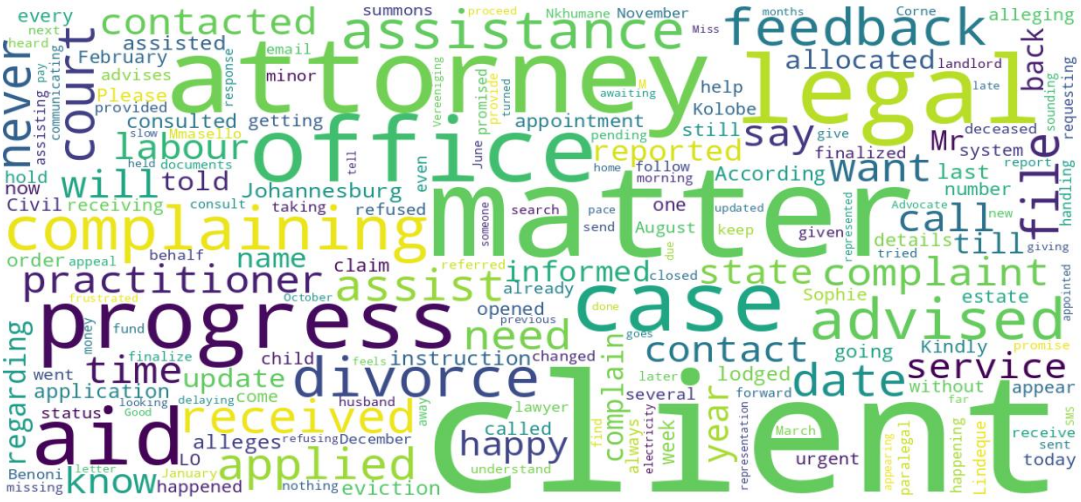

Figure 7. Word cloud

The Word Cloud result in figure 7 shows a big word, meaning a word that often appears.

4.4. Word Scores

Figure 8 below illustrates the distribution of scores for various words in the neg_df DataFrame. Each bar represents a word from the words column, and its height corresponds to the associated value in the scores column, providing a visual comparison of their scores. The x-axis displays the words, while the y-axis shows their respective scores.

 ISSN 2723-6471

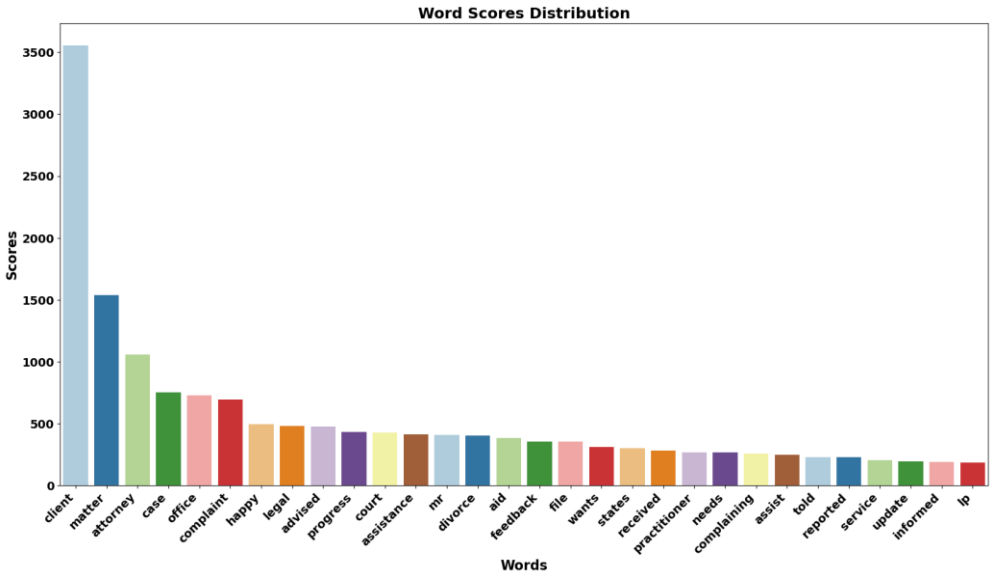

Figure 8. Word Scores

4.5. Polarity Analysis

Sentiment labelling uses Python programming to create negative and positive data dictionaries and a system. The labelling results are shown in figure 9. Labelling divides the class into negative, neutral, and positive categories. Each sentence will contain the value of each word containing the sentiment class, as in figure 9. Classifying positive, negative, and neutral sentences is necessary to calculate the text's polarity [31]. Textblob has been used to analyze the text polarity and sentiments. Figure 9 shows the counts for positive, negative, and neutral polarity. Negative sentiments have the highest counts with about 35,06%, followed by Positive with 34.20% and neutral sentiments with 30.75%. This shows more negative complaints about Legal Aid SA's service offerings to its clients. The organization must focus on this to improve its service delivery. If the polarity is greater than 0, the text will be classified as positive tweets; 0 polarity will go for neutral tweets, while if the polarity is less than 0, the text will be classified as negative. The sentiments have a subjectivity value of 0.2 and a polarity value of -0.3, which is less than 0, representing negative sentiments.

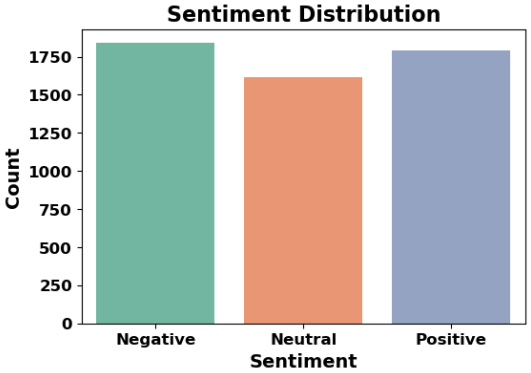

Figure 9. Labelling data

According to [17], sentiment analysis determines the writer's emotion, i.e., whether the sentence's emotions are inclined towards negative, neutral or positive directions. TextBlob methods of sentiment method return two properties, polarity and subjectivity [32]. Subjectivity is an objective sentence expressing factual information about the world, while a subjective sentence expresses personal feelings or beliefs. The subjectivity score is a float value between [0 and 1]. When the subjectivity value is closer to 0, this is a more factual reflection as the subjectivity becomes more of an opinion. In this case, the subjectivity score of 0.2 indicates that the text is predominantly objective. This suggests that our dataset contains more factual information than personal opinions.

Polarity refers to identifying sentiment orientation, such as negative, neutral, and positive. Polarity is a float value in the [-1, 1] range. The polarity 1 means a positive statement, -1 means a negative statement, and 0 means neutral [17]. The polarity score of -0.3, which is between -1 and 1, is notable in our data and reflects a slight leaning towards negativity in the sentiment of the dataset. Although not strongly negative, reflecting a trend of unfavourable sentiment.

4.6. Machine Learning and Statistical Analysis

This section evaluates the model's usefulness for Naive Bayes, Stochastic Gradient Descent (SGD), Random Forest, Support Vector Classification (SVC), Logistic Regression, and Extreme Gradient Boosting (XGBoost). These models facilitated efficient experimentation with available resources while yielding valuable insights. We acknowledge that other methods and models, such as deep learning models, could have produced similar or better results. It is to be noted, however, that Deep learning models necessitate substantial computer resources and extensive labelled data for effective training. Deep learning models generally surpass regular machine learning models when the dataset is extensive and varied. Nevertheless, conventional models such as SVC, Random Forest, and XGBoost for low to medium-sized datasets can still yield competitive outcomes without requiring extensive data or processing resources. Numerous conventional machine learning models, such as Logistic Regression, Naive Bayes, and Random Forest, offer a level of interpretability frequently absent in deep learning models. This study aimed to compare a selection of machine learning algorithms frequently employed in sentiment analysis applications. Incorporating deep learning models would have markedly heightened the experiment's complexity without enough rationale or comparative analysis relative to the study objectives. Table 1, table 2, table 3, table 4, table 5 depict the F1 Score, Precision, and Recall, while figure 10, figure 11, figure 12, figure 13, figure 14 provide a confusion matrix.

Table 1. Naïve Bayes classification

Sentiments	Precision	Recall	F1-Score	Support	Training Accuracy	Testing Accuracy
Negative	0.75	0.72	0.73	384	0.87	0.73
Neutral	0.69	0.69	0.76	265		
Positive	0.75	0.68	0.71	401		
Accuracy			0.73	1050		
Macro Avg	0.73	0.74	0.73	1050		
Weighted Avg	0.74	0.73	0.73	1050		

From table 1 above, we observed that Naïve Bayes Training accuracy was 87%, indicating that the model excels on the training dataset and assimilates patterns proficiently. The disparity between training accuracy (87%) and testing accuracy (73%) suggests potential overfitting. Testing Accuracy of 73% indicates a moderate performance, with room for improvement. The model achieves a balanced performance across all sentiment categories but struggles with generalization compared to training. Figure 10 depicts the Confusion Matrix for the Naïve Bayes model, providing a detailed representation of the model's performance.

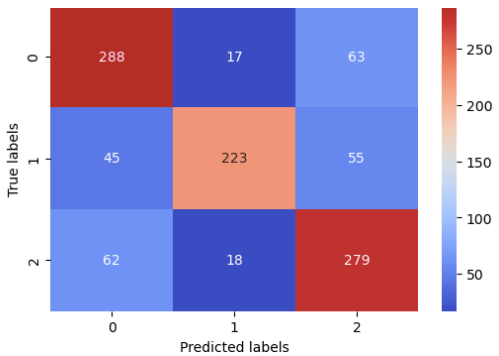

Figure 10. Confusion Matrix Naïve Bayes

ISSN 2723-6471

From table 2, the SGBD classifier's Training Accuracy of 98% illustrates high training accuracy, indicating that the model performs exceptionally well on the training data, capturing patterns effectively. A testing Accuracy of 85% illustrates a vital testing accuracy, suggesting that the model generalizes well to unseen data.

Table 2. SGBD Classifier

Sentiments	Precision	Recall	F1-Score	Support	Training Accuracy	Testing Accuracy
Negative	0.79	0.90	0.85	322	0.98	0.85
Neutral	0.91	0.86	0.89	342		
Positive	0.88	0.82	0.85	386		
Accuracy			0.86	1050		
Macro Avg	0.86	0.86	0.86	1050		
Weighted Avg	0.86	0.86	0.86	1050		

Figure 11 depicts the Confusion Matrix for the SGD Classifier model, providing a detailed representation of the model's performance.

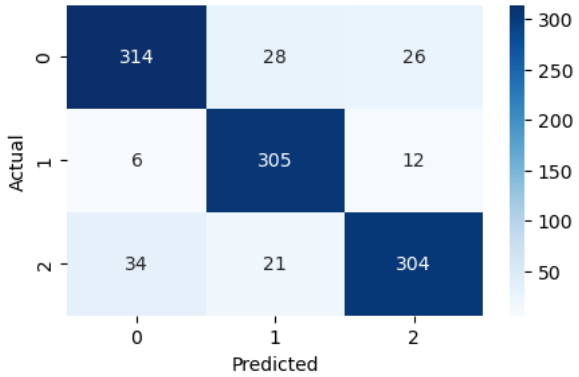

Figure 11. Confusion Matrix: SGD Classifier

From table 3, the Training Accuracy of 100% indicates that the model achieves perfect accuracy on the training dataset and fully captures patterns in the training data. Testing Accuracy of 86% indicates that the testing accuracy is slightly lower but still demonstrates the model's ability to generalize well to unseen data. The minimal gap between training and testing accuracy suggests effective regularization or optimization.

Table 3. Random Forest

Sentiments	Precision	Recall	F1-Score	Support	Training Accuracy	Testing Accuracy
Negative	0.84	0.85	0.84	364	1.0	0.86
Neutral	0.96	0.89	0.92	350		
Positive	0.82	0.88	0.85	336		
Accuracy			0.87	1050		
Macro Avg	0.87	0.87	0.87	1050		
Weighted Avg	0.87	0.87	0.87	1050		

Figure 12 depicts the Confusion Matrix for the Random Forest model, providing a detailed representation of the model's performance.

ISSN 2723-6471

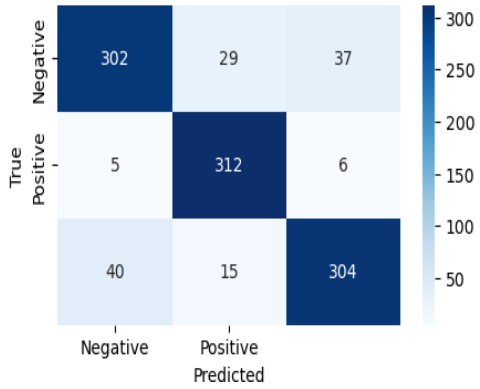

Figure 12. Confusion Matrix Random Forest

From table 4, The SVC Training Accuracy of 93% and the testing accuracy of 86% indicate that the model generalizes well to unseen data without significant overfitting. The alignment of macro and weighted averages at 86% shows that the model treats all classes fairly, with no significant bias toward one sentiment category.

Table 4. SVC

Sentiments	Precision	Recall	F1-Score	Support	Training Accuracy	Testing Accuracy
Negative	0.80	0,85	0,82	346	0.93	0.86
Neutral	0.96	0.85	0.90	365		
Positive	0.83	0.88	0.85	339		
Accuracy			0.86	1050		
Macro Avg	0.86	0.86	0.86	1050		
Weighted Avg	0.86	0.86	0.86	1050		

Figure 13 depicts the Confusion Matrix for the SVC model, providing a detailed representation of the model's performance.

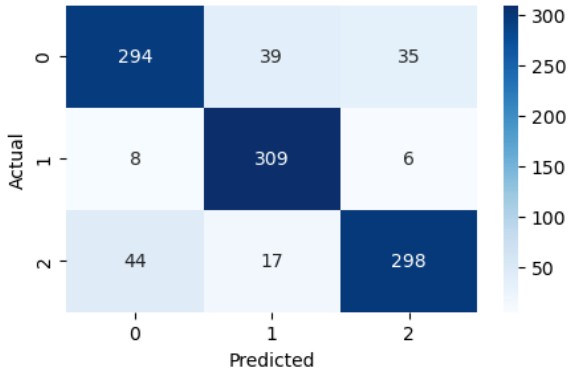

Figure 13. Confusion Matrix SVC

From table 5, the Logistic Regression model maintains a robust testing accuracy of 87%, which aligns well with its training accuracy of 98%, suggesting minimal overfitting.

Table 5. Logistic Regression

Sentiments	Precision	Recall	F1-Score	Support	Training Accuracy	Testing Accuracy
Negative	0.83	0.88	0.85	349	0.98	0.87
Neutral	0.95	0.87	0.91	352		
Positive	0.85	0.87	0.86	349		

Accuracy			0.87	1050
Macro Avg	0.88	0.87	0.87	1050
Weighted Avg	0.88	0.87	0.87	1050

Figure 14 depicts the Confusion Matrix for the Logistic Regression model, providing a detailed representation of the model's performance.

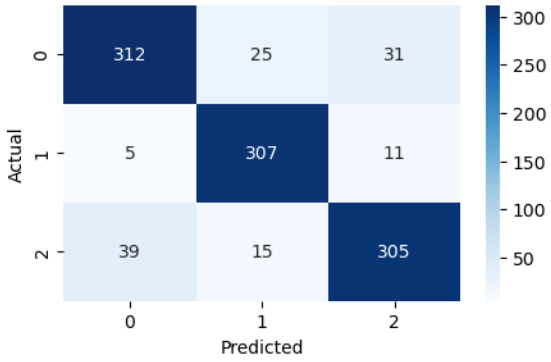

Figure 14. Confusion Matrix Logistic Regression

Table 6 below provides a summative view of all model performances for comparative purposes.

Table 6. Model and Test Accuracy

Model	Test Accuracy
XGBOOST	0.90
SVC	0.90
Random Forest	0.88
SGD	0.87
Logistic Regression	0.85
Naive Bayes	0.82

4.7. Weighing TF-IDF

Word weighing implements TF-IDF. Counting how many words appear in the document and the stages of weighin' on the word are performed after pre-processing. TF-IDF to value the term and then the term value for the input on the SVM classification process. The process of grinding the Python TF - IDF with the programming language assisted by the Scikit learn library, TfidfVectorizer, is below in figure 14.

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vect = TfidfVectorizer()
tfidf_vect.fit(df['Tweet'])
train_X_tfidf = tfidf_vect.transform(df_train['Tweet'])
train_X_tfidf = tfidf_vect.transform(df_test['Tweet'])
tfidf_vect
```

Figure 14. TfidfVectorizer

4.8. Support Vector Machine Classification

The research involved categorizing data received from tweets and complaint systems regarding Legal Aid SA services. A total of 5246 data points were collected over 5 years. The SVM classification method consists of several phases, including pre-processing, sentiment tagging, and word weighing using TF-IDF. The data is then divided into training

and test data for the classification stage. The utilized approach is a SVM, which employs the linear kernel technique. The sentiment labelling value yields the same level of accuracy for each label, and this accuracy is represented by the weight value produced. Next, we analyze the confusion matrix for several classes to determine the accuracy, precision, recall, and F1-score values. Classification testing employs split data, with 90% allocated to training data and 10% to test data, encompassing a wide range of diverse data.

5. Discussion

This study evaluated the effectiveness of several classification algorithms on a dataset, including Naive Bayes, SGDClassifier, Random Forest, SVC, and Logistic Regression. Performance was assessed using accuracy, precision, recall, and F1 scores. Naive Bayes had a training accuracy of 87.13% but a lower validation accuracy of 73.24%, indicating limited generalization. Although potential overfitting was noted, SGDClassifier showed strong results with training and validation accuracies of 98.81% and 85.71%. Random Forest achieved perfect training accuracy and a validation accuracy of 86.95%. SVC had accuracies of 93.42% and 83.71%, performing well but slightly underperforming compared to XGBoost. Logistic Regression showed solid results with 98.95% and 87.33% accuracies but fell short in test performance compared to top models. XGBoost (90.10%) and SVC (90.00%) were the best performers in the testing phase, followed by Random Forest (88.00%) and SGDClassifier (87.00%). Logistic Regression and Naive Bayes lagged with 85.00% and 82.00% test accuracies, respectively.

Future research may include hyperparameter optimization, ensemble methodologies, and feature engineering. Furthermore, cross-validation and evaluating a broader range of datasets may improve model generalization. This study has shown substantial enhancements relative to previous research, especially with our SVC model, which attained precision, recall, and F1 scores of 86%, indicating exceptional performance. Abimbola et al. investigated the improvement of legal sentiment analysis with a Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) document-level model. Their findings indicated that conventional machine learning techniques were ineffective, with SVM attaining an accuracy of 52.57%, Naïve Bayes at 57.44%, and Logistic Regression at 61.86%, highlighting the necessity for more robust methodologies [5].

Sumayah et al. utilized SVM to assess the sentiment of the Indonesian populace regarding the Metaverse. Their model attained an accuracy of 81%, with an average precision of 79%, a recall of 63%, and an F1 score of 57%, derived from 2,504 data points. This work emphasized the potential of SVM for sentiment analysis, yet its performance was lower than that of our SVC model [15].

Jefriyanto et al. examined the Naïve Bayes classification to assess sentiment performance both with and without stemming and stopwords. The optimal outcomes, attained through stemming, indicated an F1-score of 65%. This result, although justifiable, illustrates the inherent constraints of Naïve Bayes, particularly in the context of complex linguistic input. Our Naïve Bayes implementation attained an improved F1 score of 73%, presumably because of a more comprehensive and detailed dataset [13].

Ahmad et al. investigated auto-labeling to enhance aspect-based sentiment analysis via the K-Nearest Neighbours (KNN) technique [31]. Their methodology categorized comments from Twitter users with an accuracy of 79.43% over 1,409 data points. Although their approach could improve accuracy, it still needed to meet the testing accuracies and F1 scores attained by our study's sophisticated machine learning models, such as SVC, Logistic Regression, and Random Forest.

In contrast to previous studies, this research utilizes a more extensive dataset (5,246 items) and implements more thorough assessments of machine learning models. In our research, the SVC and XGBoost models exhibited testing accuracies of 90.10% and 90.00%, respectively, with F1 scores exceeding 85%. These findings highlight the benefits of employing sophisticated machine learning methodologies and comprehensive datasets for sentiment analysis, establishing a standard for subsequent research.

6. Conclusion

Using sentiment analysis tools, this study analyzed client feedback regarding Legal Aid services in South Africa. Using machine learning models such as Logistic Regression, Naive Bayes, and SVM we categorized client sentiments into positive (35.06%), negative (34.20%), or neutral (30.75%). The results suggest that most feedback is either neutral or positive, indicating a generally favourable perception of Legal Aid services. However, significant negative feedback related to service delays and communication inefficiencies highlights areas for improvement. While this study primarily utilized data from an internal client feedback system, there is potential value in social media platforms, such as Twitter, for assessing public sentiment and identifying specific concerns.

Beyond Legal Aid South Africa, these findings emphasize the transformative potential of sentiment analysis in the legal and public service sectors. Other organizations, such as law companies, legal aid programs facilitated by governments, and other judicial systems, may leverage sentiment analysis to understand client experiences better, uncover systemic inefficiencies, and identify priority areas for service improvement. Law companies could analyze client feedback to improve services and address common challenging areas, while government legal aid programs could monitor feedback to ensure that access to justice is promoted.

Logistic Regression surpassed all other machine learning models, demonstrating its superiority as a technique for sentiment analysis in comparable data sets. Organizations with more extensive and varied datasets could benefit from advanced NLP models, such as transformer-based architectures, to uncover significant insights. Organizations can transition from reactive problem-solving to proactive, data-driven decision-making by adopting these approaches.

Overall, this study demonstrates that sentiment analysis can be a strategic tool for improving legal services. By continuously monitoring sentiment trends, legal aid organizations across the globe, including private legal firms, can enhance client satisfaction, optimize resource allocation, and foster greater trust and transparency. Future studies could expand on this work by applying the model to more extensive and diverse datasets and using advanced NLP models such as RNN, transformers (BERT), GPT, or LSTM.

7. Declarations

7.1. Author Contributions

Conceptualization: J.K., D.M., and A.O.; Methodology: A.O.; Software: J.K.; Validation: J.K., A.O., and D.M.; Formal Analysis: J.K., A.O., and D.M.; Investigation: J.K.; Resources: A.O.; Data Curation: A.O.; Writing Original Draft Preparation: J.K., A.O., and D.M.; Writing Review and Editing: A.O., J.K., and D.M.; Visualization: J.K. All authors have read and agreed to the published version of the manuscript.

7.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

7.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

7.4. Institutional Review Board Statement

Not applicable.

7.5. Informed Consent Statement

Not applicable.

7.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Gupta and R. Sandhane, "Use of sentiment analysis in social media campaign design and analysis," *CARDIOMETRY*, vol. 1, no. 22, pp. 351–363, May 2022, doi: 10.18137/cardiometry.2022.22.351363.

[2] O. Dontcheva-Navratilova and R. Povolná, "Offensive language in media discussion forums: A pragmatic analysis," *Lodz Papers in Pragmatics*, vol. 19, no. 2023, pp. 223–238, Dec. 2023, doi: 10.1515/lpp-2023-0012.

[3] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artif Intell Rev*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.

[4] Jia Guo and Jia Guo, "Deep learning approach to text analysis for human emotion detection from big data," *Journal of intelligent systems*, vol. 31, no. 1, pp. 113–126, Jan. 2022, doi: 10.1515/jisys-2022-0001.

[5] Bola Abimbola, Enrique A. De La Cal Marín, and Qing Tan, "Enhancing Legal Sentiment Analysis: A Convolutional Neural Network-Long Short-Term Memory Document-Level Model," *Machine Learning and Knowledge Extraction*, vol. 2024, no. 6, pp. 877–897, 2024, doi: 10.3390/make6020041.

[6] M. Varghese, D. DCruz2, S. Aiswarya, M. James, and H. Renjini, "A Sentimental Analysis And Opinion Mining System For Mobile Networks," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 3, pp. 2165–2174, Jun. 2021, doi: 10.30534/ijatcse/2021/931032021.

[7] E. Hasgül, M. Karataş, M. D. Pak Güre, and V. Duyan, "A perspective from Turkey on construction of the new digital world: analysis of emotions and future expectations regarding Metaverse on Twitter," *Humanit Soc Sci Commun*, vol. 10, no. 1, pp. 484 -502, Aug. 2023, doi: 10.1057/s41599-023-01958-7.

[8] J. Mutinda, W. Mwangi, and G. Okeyo, "Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network," *Applied Sciences*, vol. 13, no. 3, pp. 1 -14, 3, Jan. 2023, doi: 10.3390/app13031445.

[9] I. Boban, A. Doko, and S. Gotovac, "Sentence Retrieval using Stemming and Lemmatization with Different Length of the Queries," *Adv. sci. technol. eng. syst. j.*, vol. 5, no. 3, pp. 349–354, 2020, doi: 10.25046/aj050345.

[10] J. Khosa, D. Mashao, A. Olanipekun, and C. Harley, "How Effective are Different Machine Learning Algorithms in Predicting Legal Outcomes in South Africa?," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1890–1900, Oct. 2024, doi: 10.47738/jads.v5i4.215.

[11] D. Khyani, B. S. Siddhartha, N. M. Niveditha, and B. M. Divya, "An interpretation of lemmatization and stemming in natural language processing," *Journal of University of Shanghai for Science and Technology*, vol. 22, no. 10, pp. 350–357, 2021.

[12] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed Tools Appl*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[13] J. Jefriyanto, N. Ainun, and M. A. A. Ardha, "Application of Naïve Bayes Classification to Analyze Performance Using Stopwords," *Journal of Information System, Technology and Engineering*, vol. 1, no. 2, pp. 49–53, Jun. 2023, doi: 10.61487/jiste.v1i2.15.

[14] A. I. Kabir, K. Ahmed, and R. Karim, "Word Cloud and Sentiment Analysis of Amazon Earphones Reviews with R Programming Language," *Informatica Economica*, vol. 24, no. 4, pp. 55–71, Dec. 2020, doi: 10.24818/issn14531305/24.4.2020.05.

[15] S Sumayah, Falentino Sembiring, and Wisuda Jatmiko, "ANALYSIS OF SENTIMENT OF INDONESIAN COMMUNITY ON METAVERSE USING SUPPORT VECTOR MACHINE ALGORITHM," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 1, pp. 143–150, Feb. 2023, doi: 10.52436/1.jutif.2023.4.1.417.

[16] A. Mahmoudi, D. Jemielniak, and L. Ciechanowski, "Assessing Accuracy: A Study of Lexicon and Rule-Based Packages in R and Python for Sentiment Analysis," *IEEE Access*, vol. 12, no. 1, pp. 20169–20180, 2024, doi: 10.1109/ACCESS.2024.3353692.

[17] R. K. Bania, "COVID-19 Public Tweets Sentiment Analysis using TF-IDF and Inductive Learning Models," *INFOCOMP Journal of Computer Science*, vol. 19, no. 2, pp. 23–41, Dec. 2020.

[18] S. N. Khan, S. U. Khan, H. Aznaoui, C. B. Şahin, and Ö. B. Dinler, "Generalization of linear and non-linear support vector machine in multiple fields: a review," *Computer Science and Information Technologies*, vol. 4, no. 3, pp. 226–239, Nov. 2023, doi: 10.11591/csit.v4i3.pp226-239.

[19] M. Niazkar *et al.*, "Applications of XGBoost in water resources engineering: A systematic literature review (Dec 2018-May 2023)," *Environmental Modelling & Software*, vol. 174, no. 2024, pp. 1–21, 2024, doi: 10.1016/j.envsoft.2024.105971.

[20] S. Wu, Q. Yuan, Z. Yan, and Q. Xu, "Analyzing Accident Injury Severity via an Extreme Gradient Boosting (XGBoost) Model," *Journal of Advanced Transportation*, vol. 2021, no. 1, pp. 1–11, 2021, doi: 10.1155/2021/3771640.

[21] B. Charbuty, Adnan Abdulazeez, Adnan Abdulazeez, and A. M. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 1, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[22] O. ElSahly and A. Abdelfatah, "An Incident Detection Model Using Random Forest Classifier," *Smart Cities*, vol. 6, no. 4, pp. 1786–1813, Aug. 2023, doi: 10.3390/smartcities6040083.

[23] M. Mahdikhani, "Predicting the popularity of tweets by analyzing public opinion and emotions in different stages of Covid-19 pandemic," *International Journal of Information Management Data Insights*, vol. 2, no. 1, p. 100053, 2022, doi: 10.1016/j.jjimei.2021.100053.

[24] A. Sharma, "Guided Stochastic Gradient Descent Algorithm for inconsistent datasets," *Applied Soft Computing Journal*, vol. 73, pp. 1068–1080, 2018, doi: 10.1016/j.asoc.2018.09.038.

[25] Y. Wang, D. Qiu, Y. Wang, M. Sun, and G. Strbac, "Graph Learning-Based Voltage Regulation in Distribution Networks With Multi-Microgrids," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 1881–1895, Jan. 2024, doi: 10.1109/TPWRS.2023.3242715.

[26] U. Gazder, A. Ahmed, and U. Shahid, "Predicting Severity of Accidents in Malaysia By Ordinal Logistic Regression Models," *JTTM*, vol. 03, no. 01, pp. 11–16, Mar. 2021, doi: 10.5383/JTTM.03.01.002.

[27] S. Agrawal, S. Jain, S. Sharma, A. K.-I. J. of, and undefined 2023, "COVID-19 Public Opinion: A Twitter Healthcare Data Processing Using Machine Learning Methodologies," *mdpi.com.*, vol. 20, no. 432, pp. 1-17, 2022, doi: 10.3390/ijerph20010432.

[28] D. E. Cahyani and I. Patasik, "Performance comparison of TF-IDF and Word2Vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2780–2788, Oct. 2021, doi: 10.11591/eei.v10i5.3157.

[29] Hubert, P. Phoenix, R. Sudaryono, and D. Suhartono, "Classifying Promotion Images Using Optical Character Recognition and Naïve Bayes Classifier," *Procedia Computer Science*, vol. 179, no. 2021, pp. 498–506, Jan. 2021, doi: 10.1016/j.procs.2021.01.033.

[30] V. Balakrishnan and W. Kaur, "String-based Multinomial Naïve Bayes for Emotion Detection among Facebook Diabetes Community," *Procedia Computer Science*, vol. 159, no. 2019, pp. 30–37, Jan. 2019, doi: 10.1016/J.PROCS.2019.09.157.

[31] I. Priyadarshini, P. Mohanty, R. Kumar, R. Sharma, V. Puri, and P. K. Singh, "A study on the sentiments and psychology of twitter users during COVID-19 lockdown period," *Multimedia Tools and Applications*, vol. 81, no. 19, pp. 27009–27031, 2022, doi: 10.1007/s11042-021-11004-w.

[32] Ankit, Ankit, Nabizath Saleena, and N. Saleena, "An Ensemble Classification System for Twitter Sentiment Analysis," *Procedia Computer Science*, vol. 132, no. 2018, pp. 937–946, Jan. 2018, doi: 10.1016/j.procs.2018.05.109.