TEXT-CLASSIFY: A COMPREHENSIVE COMPARATIVE STUDY OF LOGISTIC REGRESSION, RANDOM FOREST, AND KNN MODELS FOR ENHANCED TEXT CLASSIFICATION PERFORMANCE

Ravikant Kholwal
PDPM IIITDMJ, Jabalpur, India

ABSTRACT

In an era inundated with text documents, the essence of text classification technology is paramount, serving as a linchpin for the categorization and delineation of diverse content types and facilitating streamlined information retrieval. This research delineates the development of an intricate text classification model, specifically tailored for BBC news articles, utilizing pivotal machine learning algorithms such as logistic regression, random forest, and K-nearest neighbour algorithms. The model is meticulously structured into distinct segments including text preprocessing, representation, classifier implementation, and classification, each playing a crucial role in the overall classification process. The evaluation phase of this research was marked by rigorous testing and analytical scrutiny of three distinct classifiers on the BBC news dataset, focusing on deriving outputs characterized by parameters like accuracy, precision, F1-score, support matrix, and confusion matrix. These parameters were instrumental in providing insights into the features exhibiting the highest value across various classes in the dataset, thereby assessing the reliability and performance of the classification models in categorizing text data effectively. The findings of this research underscore the superior efficacy of the logistic regression classifier, integrated with the TF-IDF Vectorizer feature, achieving an impressive accuracy of 97% on the dataset, proving its reliability especially with smaller datasets. The random forest and K-nearest neighbour classification algorithms also demonstrated commendable accuracy, with rates of 93% and 92% respectively, contributing to the advancements in the field of text classification using machine learning methodologies. The insights derived from the extensive evaluations and comparisons conducted have not only contributed to the advancement of text classification methodologies but also have enhanced the capability to organize and retrieve information efficiently in news articles. This refined classification system optimizes information retrieval in news content and lays down foundational innovations in text classification, extending its applicability to diverse domains and content types, and paving the way for more intuitive and intelligent information management systems. This document serves as a comprehensive guide, elucidating the selection rationale for these specific algorithms and aiding in discerning the most apt algorithm amongst the evaluated ones, based on meticulous analysis conducted, keeping in view the advancements and nuances in the field. The detailed exploration and results of this study are aimed at providing accessible and comprehensible solutions, advancing the field of text classification, and offering insights into models' decision-making processes, thereby fostering a deeper understanding of the models' decisions made through them.

KEYWORDS: *Natural Language processing, Logistic regression, Machine learning, Random Forest, K-Nearest Neighbour (KNN)*

I. INTRODUCTION

Artificial Intelligence (AI) and human intellect have united to usher in an exciting era of digital advancements that are revolutionizing various aspects of life. Artificial Intelligence, comprising systems capable of mimicking human tasks like object distinction and voice recognition, has seen widespread adoption across a range of fields. Machine learning, a pillar of AI, has emerged as an essential field, employing past learning experiences to predict future outcomes via algorithmic analyses of datasets. Machine learning's recent comeback can be attributed to the increased transparency of its underlying algorithms, making once-difficult tasks more accessible. Machine learning's origins lie in

mathematics and statistics, yet its applications span numerous fields, such as object differentiation, speech and text classification, weather forecasting, face recognition, medical diagnostics, etc. Machine learning's strength lies in its use of data exclusively as its basis; big data's role is essential to its evolution.

Natural Language Processing (NLP) has revolutionized text classification. At an age when vast quantities of unstructured data are created daily, approximately 80% remains unstructured, requiring its transformation into structured formats for meaningful analysis. NLP, in conjunction with text mining, offers a solution enabling computers to extract valuable insights from various languages like Persian, Turkish, Chinese and English. NLP algorithms utilizing machine learning techniques to deduce rules from the text have found multiple applications across many fields: text classification, information extraction and retrieval, speech tagging and opinion mining, among others. NLP has become an indispensable part of modern computing as it empowers computers with intelligence in interpreting textual data intelligently.

Text classification can be understood as a process analogous to mathematical mapping and can be represented as:

$$f: X \to Y \qquad (1)$$

In this representation, X symbolizes the varied sets of text that are slated for classification, and Y represents the distinct categories to which the texts are assigned. Here, f is the function that executes the classification, associating each piece of text in set X to a designated category in set Y. This mathematical representation offers a clear and organized framework, elucidating the principles of text classification. It ensures that every piece of text is methodically allocated to a particular category, enabling streamlined and effective management and retrieval of information. This analogy with mathematical mapping underscores the organized and deliberate methodology inherent in the process of classifying and categorizing textual information.

This research paper describes a method for assigning predefined categories to texts within a dataset for efficient information retrieval. Classifiers use machine learning models to assess groups of texts and assign appropriate tags. Logistic regression, random forest, and K-nearest neighbour classifiers were employed to classify textual data. A comparative analysis was then performed in order to select the most efficient algorithm. Results have exceeded initial expectations with higher accuracy and precision than was anticipated while also identifying areas for further optimization of these algorithms, potentially leading to increased text classification performance.

II. RELATED WORKS

In a study by Yen et al. [1], logistic regression was employed for Chinese text categorization with an innovative approach that used N-gram-based language models instead of tokenization methods. This model, which takes into account word relationships in categorizing Chinese text categorization processes, eliminates the need for Chinese word tokenizers. In addressing the complications arising from out-of-vocabulary words, a ground breaking smoothing method has been introduced, utilizing logistic regression to amplify accuracy. This method is applied to fine-tune the probability associated with N-grams, while concurrently crafting a feature selection strategy, specifically designed for models based on N-grams. The findings from this research have shown that the adoption of this innovative method has led to considerable enhancements in F-measure performance in a variety of instances.

Aseervatham and colleagues [2] have explored the application of logistic regression in resolving the challenges related to text categorization. They have argued that the performance of ridge logistic regression is closely aligned with the capabilities of support vector machine (SVM) algorithms. However, logistic regression stands out due to its ability to derive probability values, offering a deeper insight compared to the scores generated by other models. To augment the capabilities of logistic regression, the researchers have introduced a unique method of selection. This method, by initially calculating and subsequently selecting features, achieves sparse solutions that are comparable to ridge solutions, creating a balance between these and LASSO solutions and leading to optimal results.

Further, Elghazel and his team [3] have investigated ensemble multi-label text categorization, integrating rotation forest and latent semantic indexing (LSI) to improve categorization accuracy. The methodology they introduced is built on four key concepts: 1) the application of latent semantic indexing to explore a lower-dimensional concept space, 2) the randomized segmentation of vocabulary words to avoid any bias, 3) the bootstrapping of documents to ensure a representative sample, and 4) the incorporation of BoosTexter as a robust multi-label base learner. This comprehensive approach, by leveraging the intrinsic semantic structure of the text, aimed to enhance accuracy levels. The integration of latent semantic indexing and rotation forest resulted in remarkable improvements in precision, ranking loss, and error, surpassing five other prominent approaches when evaluated on a range of textual datasets. The exploration of these methodologies highlights the ongoing advancements and refinements in the domain of text categorization, with each study offering nuanced insights and pioneering solutions to the broader dialogue.

Nadi and Moradi [4] highlighted the power of random forest as an ensemble method, especially for handling high-dimensional data. Their research provided an innovative technique to enhance its performance: by adjusting both tree number and depth for each individual tree. With this bounding technique, they aimed at increasing multiple perspectives or local views of a problem by restricting tree depth; their results demonstrated this approach can significantly improve classification accuracy on high-dimensional problems.

N. Kumar and A. Kumar [5] conducted a research study with the aim of improving the efficiency of Australia's forest fire detection system. Australian bushfires in 2019-20 provided an important reminder of the urgent need to take measures that ensure effective prevention, protection, and preservation of the environment for its various species that depend on it. This study used data mining and machine learning techniques to detect forest fires quickly so that timely action could be taken to limit damage and lessen their effects. Researchers employed multiple classification algorithms, such as K-Nearest Neighbors (K-NN) and Artificial Neural Networks (ANN), in order to address limitations associated with existing fire detection systems. By analyzing a Kaggle dataset and employing Multilayer Perceptron (MLP) algorithm within an artificial neural network (ANN), this study demonstrated improved detection rate accuracy as evaluated through confusion matrix calculations. LANCE FIRMS data provided by NASA Earth Science Data and Information System (ESDIS), while model training/testing occurred using the University of Maryland dataset and implemented into Python, were utilized during research.

Tan [6] conducted a study to optimize the K-nearest neighbour (KNN) classifier by addressing misfitting model issues arising from its assumptions. Recognizing its simplicity and efficiency, he proposed an innovative refinement strategy called DragPushing; experimental results confirmed this strategy had improved the performance of the KNN classifier.

III. METHODOLOGY

Machine learning algorithms are employed for text classification, each offering differing levels of accuracy and precision. In order to evaluate and compare these algorithms, three distinct classifiers - logistic regression, random forest, and K-nearest neighbours - were employed on a specific dataset for testing purposes. These classification algorithms have proved vital in assessing machine learning strategies and providing accessible and comprehensible solutions. Each algorithm operates differently; logistic regression relies on a particular formula for classification and prediction, while random forest creates nodes and trees. Research efforts conducted utilizing each of these three algorithms for text classification solutions have been found that help advance this field.

In the methodology illustrated in Figure 1, a structured approach is delineated, outlining each phase in the procedure. Initially, the necessary libraries are imported for subsequent integration into the coding framework. Following this, the dataset intended for classification is introduced, with the BBC news dataset [7] being the dataset of choice in this instance. This dataset is structured with two columns: (i) the category, representing five distinct classes, and (ii) the text, containing the associated lines of text for each category.

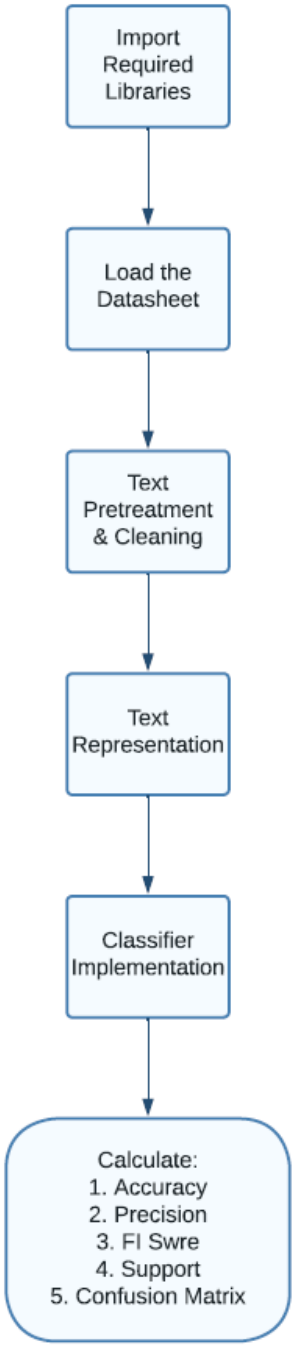

Fig. 1 Architecture of the implementation

Post the data loading phase, it is imperative to conduct any requisite cleaning operations to ensure the integrity of the data before progressing to the manual preprocessing of text lines. Once the preliminary focus is established, the emphasis shifts to text representation, a crucial step in refining the data for analysis. During the evaluation phase, three distinct classifiers are deployed on the dataset, yielding outputs characterized by five parameters. These parameters signify features that exhibit the highest value across the various classes in the dataset. The parameters include metrics like accuracy, precision, F1-score, support matrix, and confusion matrix, each providing a different perspective on the performance and reliability of the classification models in categorizing the text data effectively.

3.1 Data Preprocessing and Representation

An initial step in data preprocessing involved creating a bar graph to visualize the distribution of text across five classes in the dataset, thus providing a clear overview without manual inspection. The bar graph was produced by plotting each category against text lines on the y-axis: for instance, the business had over 500, entertainment nearly 400 and politics 400+ lines, respectively. Once this graph had been examined, it allowed data preprocessing steps to begin, as shown in Figure 2.

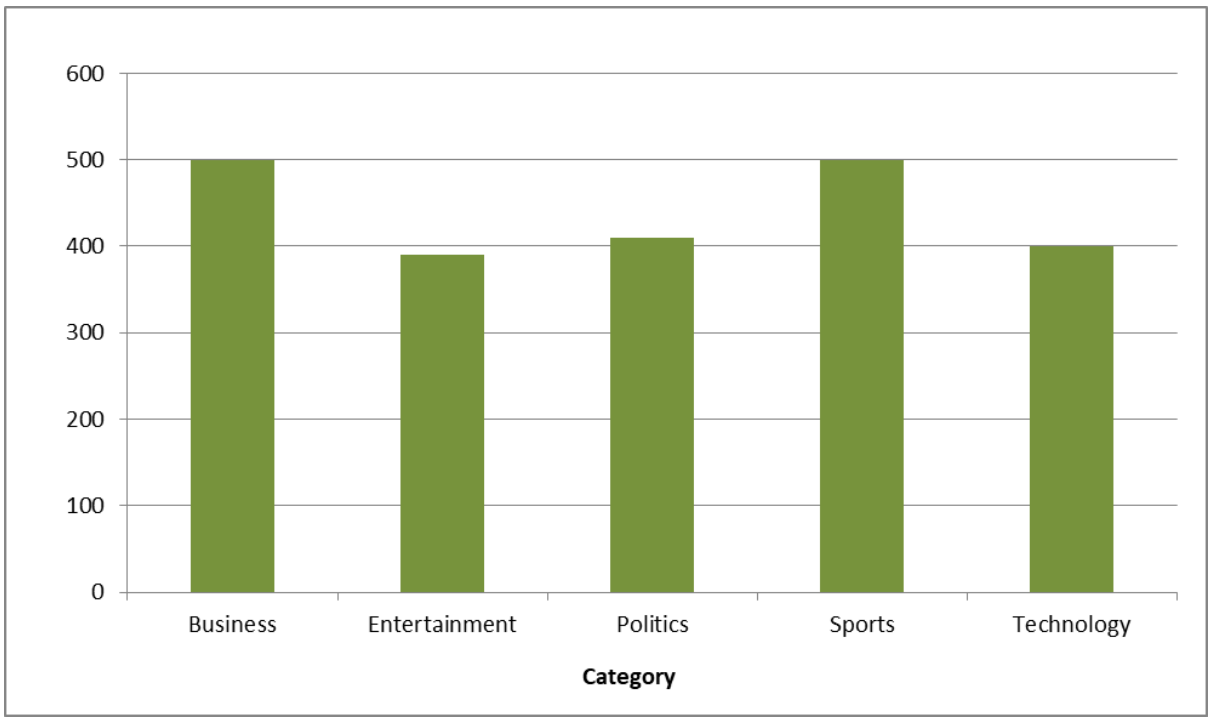

Fig. 2 Bar graph depicting the number of texts lines for different classes.

Pre-processing or cleaning documents is an integral step in classifying them, so this section outlines all of its steps. Certain words, including prepositions, conjunctions and pronouns, do not significantly contribute to discriminating classes and are known as stop words. To increase classification accuracy, it is crucial that these stop words (such as 'the', 'a, 'and', but' or' etc.) are removed - for instance, by eliminating stopwords like these: 'the'; "a", "and", but" "or", etc.). To accomplish this task, a list of English stop words was downloaded and utilized to filter out irrelevant words [8]. Furthermore, stemming was applied in order to normalize the text; stemming helps condense sentences by eliminating tense variations while maintaining meaning. Porter Stemmer algorithm was chosen for this task, following pre-processing text with the lambda function and followed by joint operation on stemmed text. Sub-operations were then performed to assess whether all text consisted of lowercase and uppercase alphabets in order to unify and facilitate the classification of all the text. All lowercase alphabets were converted to uppercase. Once all pre-processing and cleaning had been completed, as illustrated in Table 1, cleaned text was obtained and included within it.

Table 1 illustrates this step of data preprocessing and representation by showing the categories utilized in implementation in one column; then, original text without any preprocessing performed is showcased in another; finally, the cleaned text shows up, including the removal of unnecessary words such as "of," "in," and "the."

After cleaning is complete, the next step should be preparing text in a format easily understood by machine learning algorithms. A Vectorizer is used for this task - this tool converts sentences or text into numerical arrays called Vectors. In this study, the TF-IDF Vectorizer is utilized to convert text into a numerical representation that conveys meaningful information.

Table 1. Text before and after cleaning

Sr. no.	Category	Text	Cleaned
1	Tech	Tv future in the hand of viewers with home th...	Tv future hand viewer home theatr system plasma…sss
2	Business	Worldcom boss left books alone former worldc…	Worldcom boss left book alon former worldcom b…
3	Sport	Tigers wary of Farrell gamble Leicester say…	Tiger wary farrel gamble leicest say rush make…
4	Sport	Yeadling face Newcastle in fa cup premiership s…	Yead face Newcastl fa cup premiership side new…
5	Entertainment	Ocean a twelve raids box office ocean s twelve…	Ocean twelve raid box office ocean twelve crime c…

The term frequency normalization method takes into account how frequently each word appears within its dataset, while its counterpart, the inverse document frequency measure, removes words that add little meaning to sentences. Therefore, when terms frequently appear throughout a text, their value decreases as their frequency becomes greater than expected. The TF-IDF technique isolates less common words while extracting relevant features from the corpus [9]. Notably, this algorithm emphasizes high-frequency words within texts but with limited presence across corpora to focus on words with discriminatory power to distinguish different classes within text documents.

The equation representing the TF–IDF algorithm is expressed as:

$$TF - IDF(w) = TF(w) * IDF(w) \qquad (2)$$

The authors used an n-gram parameter as a measuring stick, representing adjacent letters or words in the text and helping predict subsequent items in sequence. N-gram captures language structure by identifying what follows after preceding letters or words in succession. Facilitates the generation of word vectors based on text context [10]. Furthermore, the norm function was employed to return a specific matrix based on the Ord parameter value; the L2 norm minimized sum-of-squared differences between target and estimated values and all elements were returned in array form after this process was complete. Subsequently, logistic regression, random forest and K-NN classifiers were individually employed to calculate accuracy, precision, F1-score and support scores.

True positives were defined as correctly predicted values where both classes predicted, and actual classes matched up; true negatives represented correctly predicted negative values where both predicted and actual classes matched up (i.e. both classes predicted and actual classes did not match up).
False positives were identified when predicted classes were considered 'yes' but actual classes were actually no. Conversely, false negatives occurred when predicted classes were identified as no, but actual classes were, in fact yes (Table 2).

Table 2. Diagram depicting four parameters

Predicted class			
Actual class	S	Class = Yes	Class = No
	Class = Yes	True positive	False negative
	Class = No	False positive	True negative

Table 3. True Positive + false negative = total predicted positive

	Negative	Positive
Negative	True Negative	False Positive
Positive	False Negative	True Positive

Table 4 True Positive + false negative = actual positive

	Negative	Positive
Negative	True Negative	False Positive
Positive	False Negative	True Positive

Accuracy is a metric used to understand the proportion of correct predictions made by a model relative to the overall number of observations. It is mathematically represented as:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} \quad (3)$$

Precision is another metric, focusing on the proportion of positive identifications that were actually correct. It is crucial when the costs of false positives are high. It is calculated as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (4)$$

Recall, or Sensitivity, measures the proportion of actual positives that were correctly identified. It is essential when the cost of false negatives is high. It is represented as:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (5)$$

The F1 Score is the harmonic mean of Precision and Recall and is a better measure when there are imbalanced classes. It is calculated using the formula:

$$F1\ score\ = 2 \cdot \frac{Precision\ \cdot\ Recall}{Precision\ +\ Recall} \qquad (6)$$

These metrics are pivotal in evaluating the performance of classification models, providing different perspectives on the model's ability to correctly identify observations, especially in cases where the classes are imbalanced or when the costs of false positives and false negatives are significantly different. They serve as a comprehensive approach to understanding the model's capability in various aspects of classification tasks, allowing for nuanced adjustments and optimizations.

Support refers to the total number of samples within a class that accurately correspond to its true response, while confusion matrices assess a classifier's ability to predict values accurately by measuring true positives - meaning values belonging to their proper class instead of misclassification into another one.

Once text pre-treatment and representation steps have been completed, implementation of classifiers takes place. Three classifiers - logistic regression, random forest, and K-nearest neighbours (KNN) are considered in order to determine optimal outputs. First, the dataset is divided into training and testing sets, with the latter comprising 25% and the former 75%, respectively, of data. Classifiers are then implemented through a pipeline approach. Pipelining assists the flow of algorithms by organizing data transformation and model testing and evaluation in an orderly manner. A machine learning pipeline typically comprises four stages: pre-processing, learning, evaluation and prediction. Adopting the pipeline approach serves multiple functions. First, it enhances the overall functionality of the model. Second, it improves data pre-processing and helps eliminate overfitting caused by datasets while providing for improved hyperparameter tuning within the pipeline. Implementation of a classifier via a pipeline also incorporates pickle as part of its implementation, which is a Python library used for object serialization and deserialization. By employing pickle, programs can be stored on disk for ease of access and predictions to be made without rewriting the entire codebase (Figure 3).

3.2 Random Forest

Random Forest's algorithm takes advantage of an ensemble of decision trees working together to form predictions. Decision trees serve as fundamental building blocks in this algorithm; Random Forest entails multiple such trees with nodes being defined during the pre-processing stage [13].

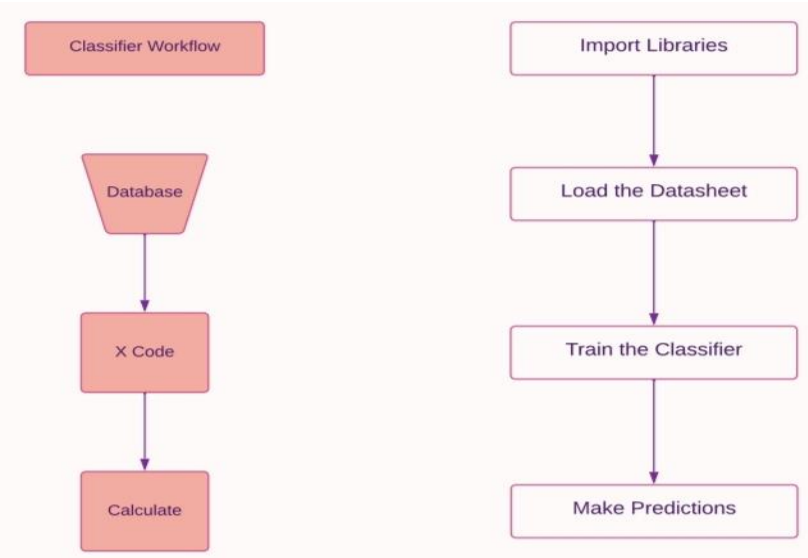

Fig. 3 Workflow of classifiers

Multiple trees are built using features from random subsets to select the optimal feature(s). Each decision tree is generated using an established decision tree algorithm. Collectively, these Random Forest trees help classify new objects from input vectors. Random Forest predictions primarily contain two sources of error, particularly with regard to decision trees: class membership and voting results across trees. However, there may still be potential sources of misclassification errors due to two key parameters:

(i) Correlated trees in a forest can increase error rates significantly.

(ii) Each tree possesses its own strength; lower error rates signify stronger classifiers and vice versa.

The random forest algorithm features two notable characteristics:

(i) It can handle large sets of input variables without necessitating variable deletion.

(ii) It reveals insights into which variables hold importance during classification processes.

(iii) It boasts efficient performance even with large databases.

(iv) Once produced, generated forests or trees can be stored for future use.

Implementation of the random forest algorithm involves several steps.

Step 1: Randomly select K data points from the training data set.

Step 2: Construct a decision tree using all K data points selected.

Step 3: Before repeating steps 1-3, determine your desired number of trees (NTree). Steps 1-3 are repeated accordingly.

Step 4: Predict the value of target variable y for a new data point by aggregating all predictions of NTree trees and selecting an average value as the final prediction.

The mathematical formula for random forest classifier is:

$$n_{ij} = w_i C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \qquad (7)$$

ni sub*(j)* = the importance of node *j*
w sub*(j)* = weighted number of samples reaching node *j*
C sub*(j)* = the impurity value of node *j*
left*(j)* = child node from left split on node *j*
right*(j)* = child node from right split on node *j*

3.3 K-nearest Neighbours

KNN (K-nearest Neighbours) is an algorithm for grouping similar items together within a dataset by using class labels and feature vectors as classifiers while also employing similarity measures for classifying new cases [12]. When applied to text classification, this approach represents texts using spatial vectors such as S(T1, W1; T2, W2; etc.). To determine similarity, it compares each text against its training set to find those with the highest similarity before selecting text with K nearest Neighbours classes, as shown in Figure 4.

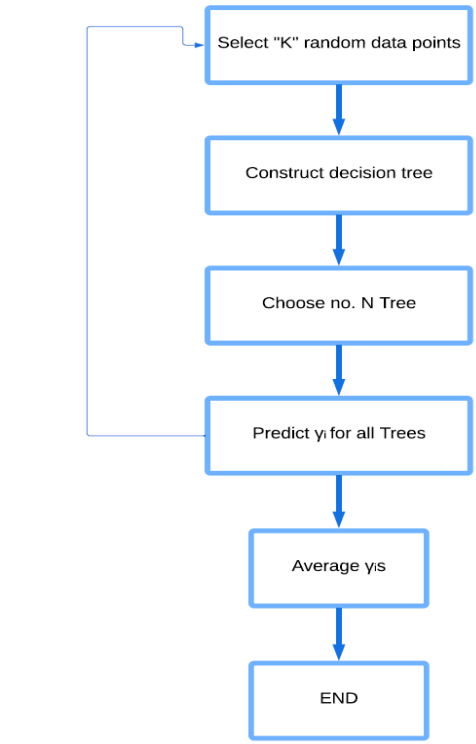

Fig. 4 Flow chart for random forest classifier.

Implementation of the K-nearest Neighbours (KNN) algorithm includes several steps. These include:

Step 1: Initially, select the desired number of neighbours, known as K.

Step 2: Calculate Euclidean distance to identify K-nearest neighbours for a new data point.

Step 3: To count the data points belonging to each category within K neighbours.

Step 4: Assign the new data point to the category with the highest count among its neighbours, as shown in Figure 5.

The K-Nearest Neighbors (KNN) algorithm, when applied for text classification, follows a sequence of steps to categorize incoming text based on its similarity to the training text. Here's a more detailed explanation of the steps and the mathematical formula involved:

1. Representation of Texts as Feature Vectors: Initially, both the training text and the incoming text are represented as feature vectors within a vector space. This representation is crucial for quantifying the similarity between different texts.

2. Comparison of Feature Vectors: Subsequently, a comparison is made between the feature vector of the incoming text and the feature vectors of each training text. This comparison is quantified using a mathematical formula that calculates the similarity between the vectors. The similarity between two feature vectors, $sim(di, dj)$ is calculated using the following formula:

$$sim(di, dj) = \frac{\sum_{k=1}^{M} W_{ik} W_{jk}}{\sqrt{\sum_{k=1}^{M} {W_{ik}}^2} \sqrt{\sum_{k=1}^{M} {W_{jk}}^2}} \quad (8)$$

where,
di and dj are the feature vectors of the incoming and training text, respectively.
M represents the dimension of the feature vector.
W_{ik} and W_{jk} are the k-th elements of vectors di and dj, respectively.

This method ensures that the incoming text is compared with each piece of training text in a structured manner, allowing for the identification of similarities based on the feature vectors, and subsequently enabling accurate classification of the incoming text.

3. In the K-Nearest Neighbors (KNN) algorithm, the final step involves selecting the K-nearest neighbors of the incoming text. This selection is grounded on the computed similarity or comparison between texts. The similarity between the incoming text, represented as $sim(d_i, d_j)$ is a crucial component in determining the proximity or 'neighborhood' of the texts. Here, $\delta\ (d_i, C_m)$ represents the distance or dissimilarity measure between the incoming text and its neighbors, facilitating the identification of the most similar or 'nearest' neighbors in the feature space. This approach ensures that the incoming text is compared and classified accurately, considering its resemblance to the existing texts in the training dataset, allowing for a more nuanced and precise categorization based on the inherent characteristics of the texts.

$$sim(d_i, d_j)\ \delta\ (d_i, C_m) \qquad (9)$$

The formula behind the KNN are

$$Q(d_i, C_m) = \sum_{j=1}^{K} sim(d_i, d_j)\ \delta\ (d_i, C_m) \sum_{j=1}^{K} sim(d_i, d_j)\ \delta\ (d_i, C_m)$$
$$\delta(d_i, C_m) = \{1, if\ d_i \epsilon\ C_m\ and\ 0, \quad if\ d_i \epsilon\ C_m \qquad (10)$$

3.4 Logistic regression

Logistic regression is a popular supervised classification algorithm that has experienced immense growth over the last several years and widespread implementation. It serves to classify individuals into categories according to a logistic function [14].

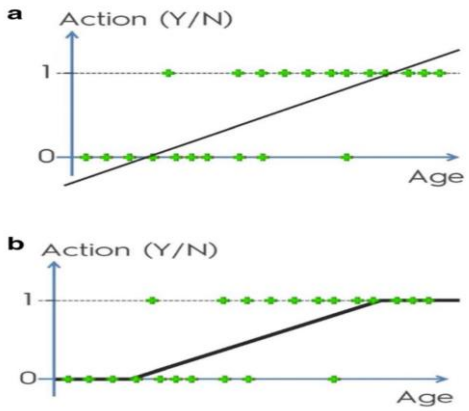

Fig. 6 a Graph when data points do not fit properly. b Graph when logistic regression is applied and one gets a perfect curve.

Figure 6a shows an example where one graph does not accurately represent all of the data points, showing how action changes with age; however, this graph fails to provide an ideal fit. For this issue, logistic regression algorithms are applied to data points, creating the graph in Figure 6b as shown by the logistic regression algorithm and its visual representation in Figure 7 by way of logistic regression's S-shaped curve (the so-called sigmoid curve).

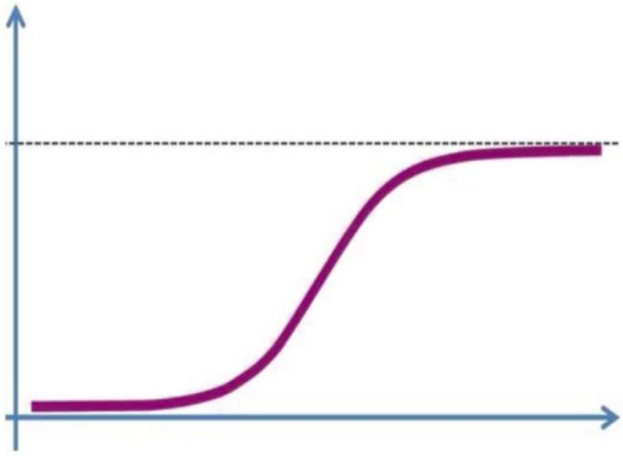

Fig. 7 Sigmoid curve.

Logistic regression stands out among similar methodologies by its unique S-shaped curve that is characteristic of its use - often known as its unique feature of application in real-world situations.

To delve into the mathematical aspect, let's start with the fundamental linear regression equation:

$$y = b_0 + (b_1 * x) \qquad (11)$$

Subsequently, we apply the sigmoid function to this, represented as:

$$p = \frac{1}{1 + e^{-y}} \qquad (12)$$

By substituting the value of y from the linear regression equation into the sigmoid function, we derive the logistic regression equation as:

$$ln\left(\frac{p}{1-p}\right) = b_0 + (b_1 * x) \qquad (13)$$

or more generally,

$$logit(S) = b_0 + b_1 M_1 + b_2 M_2 + b_3 M_3 + \ldots\ldots\ldots b_k M_k \qquad (14)$$

Here, S denotes the probability of the presence of the characteristic of interest, $M_1, M_2, M_3, \ldots\ldots, M_k$ are the predictor values, and $b_0, b_1, b_2, b_3, \ldots\ldots, b_k$ are the intercepts of the model.

The assumptions underlying logistic regression classifiers are crucial:

1. It does not assume a linear relationship between the dependent and independent variables.
2. The dependent variable is dichotomous, meaning it cannot be divided into two parts.
3. The dependent variables do not need to be normally distributed, but they should be linearly related.

In the realm of text classification, the Logistic Regression (LR) model interprets a vector comprising variables, calculates the coefficients corresponding to each input variable, and subsequently predicts the text class in the form of a word vector, enhancing the understanding of the relationship between variables and the classification of texts.

IV. RESULTS

To delve into the results obtained post-execution of the code, it is crucial to acknowledge that the assessment of the deployed algorithms is meticulously performed, taking into account five pivotal parameters. These include accuracy, precision, F1-score, support matrix, and confusion matrix. These metrics serve as the cornerstone for evaluating the effectiveness and reliability of the algorithms in classifying and predicting outcomes accurately. The emphasis on these diverse metrics ensures a holistic and comprehensive analysis, allowing for a nuanced understanding of the algorithm's performance in various aspects, thereby facilitating the refinement and optimization of the model for enhanced precision and reliability.

4.1 Random Forest Algorithm

The Random Forest algorithm is a sophisticated ensemble technique aimed at addressing both classification and regression challenges. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes for classification or mean prediction of the individual trees for regression. This method leverages the concept of ensemble learning, where the decision trees, once formed, cast votes to determine the most prevalent class in classification scenarios or provide approximations of the dependent variables in regression scenarios. The results derived from employing the Random Forest algorithm are meticulously detailed in Table 5, where the Precision, Accuracy, F1-score, and Support values for each class are meticulously computed and analyzed. To illustrate, the Business class demonstrates a precision of 90.6% and an accuracy of 93%, with an F1 score of 91% and a support value of 144. The Entertainment class mirrors this performance, achieving a precision of 93%, accuracy of 90%, an F1-score of 91%, and a support value of 96. The Politics class aligns closely with a precision of 91%, accuracy of 90%, an F1-score of 91%, and a support value of 93. The Sports class excels with a precision of 97%, an accuracy of 97%, an F1-score of 97%, and a support value of 136. Lastly, the Technology class maintains a precision of 94%, an accuracy of 94%, an F1-score of 94%, and a support value of 88, as depicted in the corresponding confusion matrix. These detailed metrics offer profound insights into the efficacy of each classifier in accurately categorizing instances into the respective classes, allowing for a nuanced understanding of the algorithm's performance across different domains. The comprehensive analysis of these metrics is pivotal in evaluating the robustness and reliability of the Random Forest algorithm in diverse classification contexts.

Table 5. Resultant outcome of all parameters in five different categories using random forest classifier

Category	Precision	Accuracy	F1-score	Support
Business	0.90	0.93	0.91	144
Entertainment	0.93	0.90	0.91	96
Politics	0.91	0.90	0.91	93
Sports	0.97	0.97	0.97	136
Technology	0.94	0.94	0.94	88
Accuracy			0.93	557

4.2 K-Nearest Neighbours Algorithm

K-Nearest Neighbors (KNN), like random forest, is widely employed across industries for solving classification and regression problems. This algorithm takes into account k-closest training examples in feature space to determine where a data point falls within its grouping scheme. According to Table 6's results, KNN shows an impressive accuracy rate of 92%. Particularly, for business class passengers, precision of 96% was reached, accuracy of 86% was obtained and an F1-score of 91% and support value of 130 was reached. Entertainment class passengers found themselves enjoying precision of 99% with accuracy 91%, F1 score of 95% and support values of 99 respectively. Politics classes achieved precision of 77%, accuracy of 98%, an F1-score of 86% and support of 109 while sports classes saw precision of 100%, 97% accuracy and an F1 score of 98% and support from 129 students. Precision of 98%, accuracy of 89%, F1-score of 93% and support of 90 were achieved in the technology class using confusion matrix analysis to reveal classifier performance for each of its classes. Assuming the classifier correctly classified 112 lines belonging to the business class and all remaining ones were assigned another class, then this value indicates that 112 of those lines belong to that category. Similarly, in the second row, 90 is an indicator that indicates that the classifier correctly classified 90 lines of text into entertainment class while assigning the remaining ones to another category. The third row depicts that 108 lines of text belong to the Politics class and were correctly classified, while some lines were misclassified into other classes. In the fourth row, the classifier successfully classified 125 lines of text for politics while misclassifying some as business, politics and technology classes. Finally, 80 lines were successfully classified for technology but misclassified into other classes.

Table6. Resultant outcome of all parameters in five different categories using K-NN classifier

Category	Precision	Accuracy	F1-score	Support
Business	0.96	0.86	0.91	130
Entertainment	0.99	0.91	0.95	99
Politics	0.77	0.99	0.86	109
Sports	0.99	0.97	0.98	129
Technology	0.98	0.89	0.93	90
Accuracy			0.92	557

4.3 Logistic regression Algorithm

The logistic regression model can be used to evaluate the statistical significance of each independent variable in relation to probability. It has proven incredibly effective at modelling binomial outcomes. Logistic regression models can accurately determine the likelihood that someone will develop cancer-based on various explanatory variables, as shown in Table 5.

Table 7. Resultant outcome of all parameters in five different categories using logistic regression classifier

Category	Precision	Accuracy	F1-score	support
Business	0.94	0.99	0.97	133
Entertainment	1.00	0.98	0.99	91
Politics	0.97	0.94	0.96	103
Sports	0.99	0.99	0.99	131
Technology	0.98	0.96	0.97	99
Accuracy			0.97	557

In Table 7, the calculated values for precision, accuracy, F1 score, and support for each class are presented. For the business class, the recorded precision is 94%, accuracy is 99%, the F1 score is 97%, and the support is 135. The entertainment class exhibits a precision of 100%, accuracy of 98%, an F1 score of 97%, and a support value of 91.1. The politics classes have a precision of 97%, accuracy of 94%, an F1 score of 96%, and support of 103. The sports classes have showcased a precision of 99%, an accuracy of 100%, F1 scores of 100%, and support of 131. Lastly, the technology classes have achieved a precision of 98%, an accuracy of 96%, an F1 score of 97%, and support of 99.

A confusion matrix is instrumental in providing insights into the ability of classifiers to accurately categorize instances into their respective classes, offering a detailed view of the performance of the classification model across different categories. This detailed perspective is crucial for understanding the model's strengths and areas that may require refinement to enhance its classification capabilities.

4.4 Comparison Analysis of Three Algorithms

To compare logistic regression, random forest, and KNN algorithms, they have been assessed based on four parameters - precision, accuracy, F1-score, and support - which allowed for comparisons across each algorithm across various categories. They presented these comparison results graphically using bar graphs to illustrate them further [15].

4.4.1 Precision

Figure 8 displays the precision values for various algorithms across classes. Logistic regression achieved precision values of 0.94 for business classes, 1 for entertainment classes, 0.97 for politics classes, 0.99 for sports classes, and 0.98 for technology. Random forest recorded preciseness values between 0.9-0.993, while KNN recorded 0.99-99-77-1.99-1 for each of their respective classes.

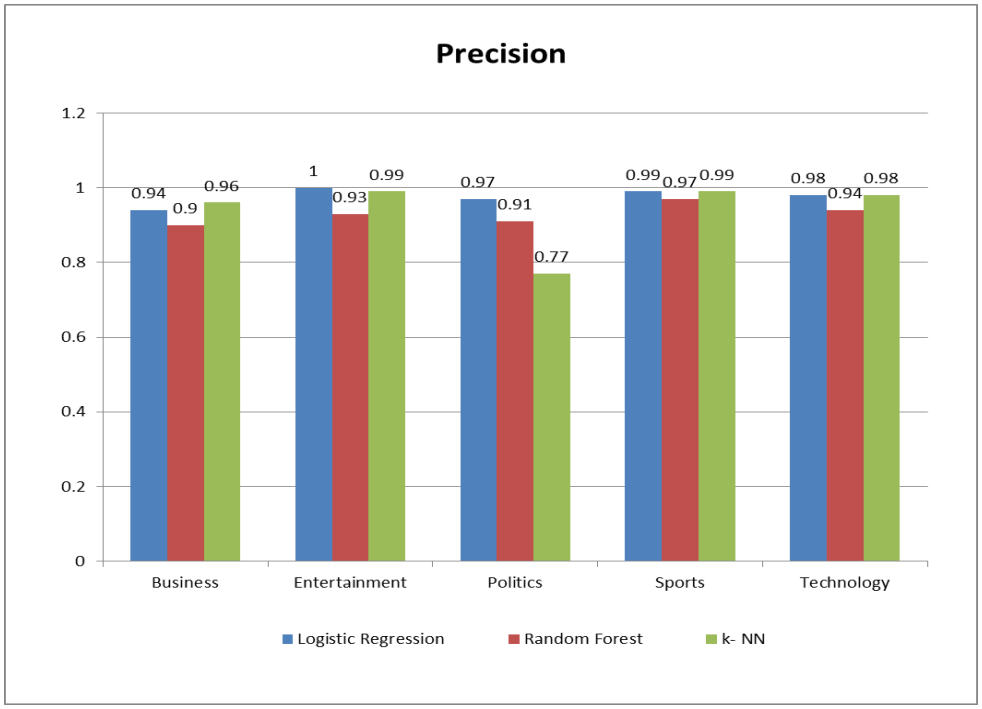

Fig. 8 Categories versus precision showing variations of data set classes with respect to change in precision

4.4.2 Accuracy

Figure 9 presents accuracy values for various classes for each algorithm. Logistic regression reached accuracy values of 0.99 in business, 0.98 for entertainment, 0.94 for politics, 0.99 for sports and 0.96 for technology, while Random Forest and KNN both achieved 0.93 accuracy levels within those same classes; KNN obtained values between 0.86, 0.91, 0.99, 0.97 and 0.89.

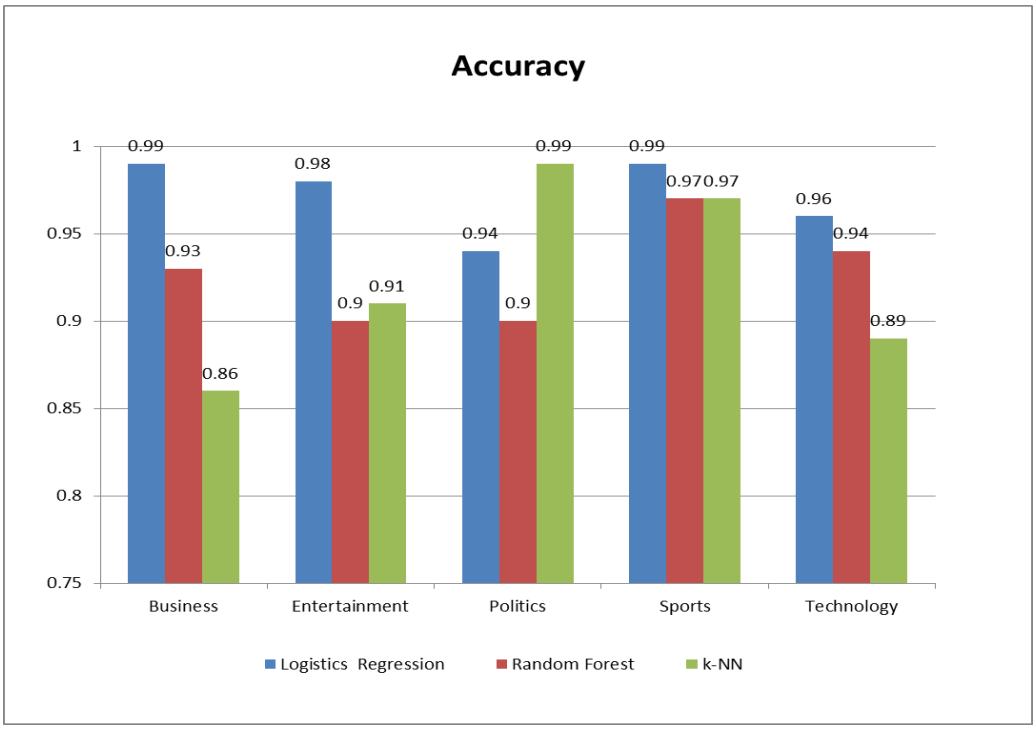

Fig. 9 Categories versus Accuracy showing variations of data set classes with respect to change in accuracy

4.4.3 F1-Score

Figure 10 provides F1-score values for various algorithms across classes. Logistic regression achieved F1-score values of 0.97 in business, 0.99 for entertainment, 0.96 for politics, 0.99 for sports and technology, while random forest achieved F1-scores between 0.91, 0.91, 0.91, 0.97, 0.94 with KNN achieving values between 0.91 0.91 0.95 0.86 0.98 and 0.93 respectively for their classes.

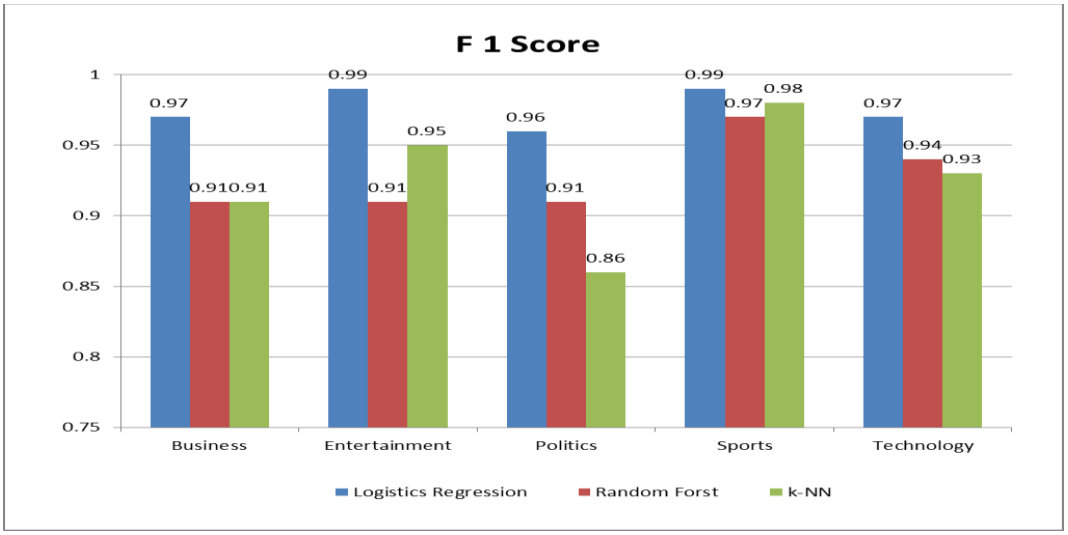

Fig. 10 Categories versus F1- score showing variations of data set classes with respect to change in F1-score

4.4.4 Support

Figure 11 displays the support values for each algorithm across various classes. Logistic regression achieved support values of 133 for business, 91 for entertainment, 103 for politics, 131 for sports and 99 for technology, respectively. Random forest achieved support values of 144, 96, 93, 136 88, while KNN reached 130, 99, 110, 129,90 for different classes.

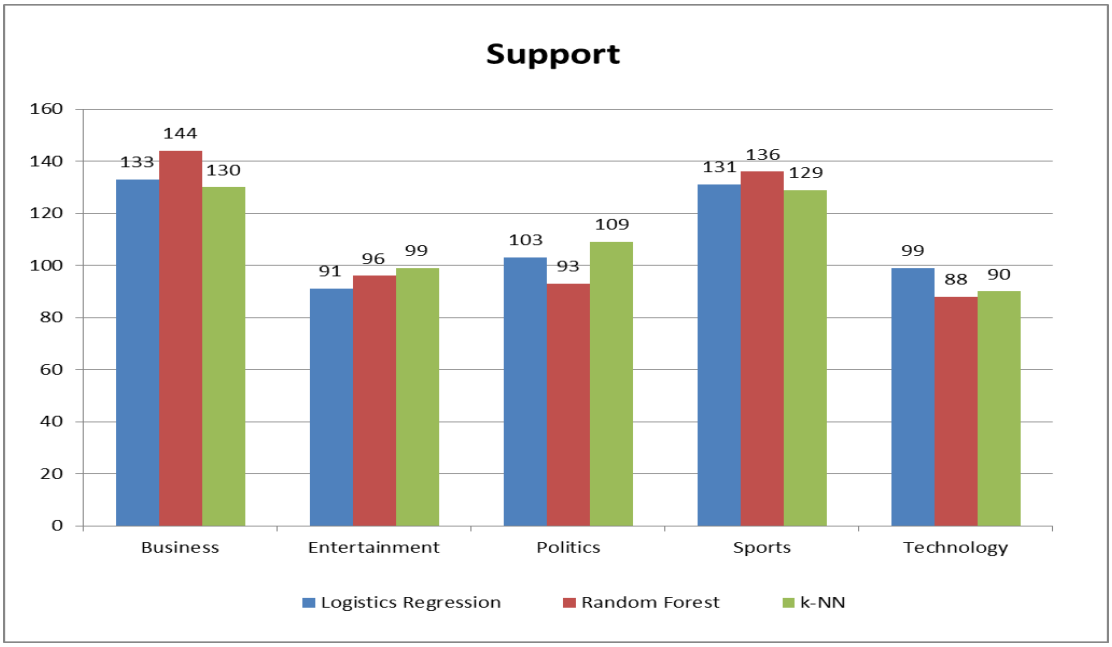

Fig. 11 Categories versus support showing variations of data set classes with respect to change in support

Overall, the comparison analysis allows us to assess the performance of three algorithms in terms of precision, accuracy, F1-score, support across various classes, and bar graphs provide an effective visual display of these results.

V. CONCLUSIONS

In the pursuit of developing advanced text classification models, this study meticulously crafted a model focusing on the categorization of BBC News articles, leveraging prominent machine learning algorithms like logistic regression, random forest, and K-nearest neighbour algorithms. The evaluation metrics, pivotal in assessing the efficacy of these algorithms, were meticulously delineated and examined in the context of this research. The importance of accuracy is undeniable in the application of machine learning algorithms to distinct datasets. The findings of this research underscore the efficacy of the logistic regression classifier, coupled with the TF-IDF Vectorizer feature, marking an impressive accuracy of 97% on the dataset in question. This classifier has manifested its superiority in accuracy, especially when dealing with smaller datasets, establishing itself as a reliable choice. Following closely, the random forest classification algorithm manifested a commendable accuracy rate of 93%. Meanwhile, the K-nearest neighbour classification algorithm secured an accuracy of 92%. The logistic regression classifier exhibited steadfast performance across diverse parameters, aligning well with the anticipated outcomes. The selection rationale for these specific algorithms is elaborated in the related works section of this study. This document serves as a comprehensive guide, aiding in discerning the most apt algorithm amongst the evaluated ones, based on the meticulous analysis and comparisons conducted, keeping in view the advancements and nuances in the field of text classification using machine learning methodologies.

VI. FUTURE SCOPE

In developing a proficient model, this study has attained noteworthy precision and accuracy levels; however, several aspects in this domain still necessitate deeper exploration. A notable challenge emerges when implementing Support Vector Machine (SVM) with the dataset utilized in this study. Additionally, the reliance on solely statistical and text-based information in this investigation presents its own set of challenges. The Random Forest algorithm has demonstrated encouraging results in diverse practical scenarios; however, addressing the learning from text data with class imbalance remains a pivotal challenge that warrants direct confrontation.

The applicability of these algorithms extends beyond the current scope, encompassing datasets inclusive of images and audio. The integration of existing technologies, such as image recognition for image datasets and Part-Of-Speech text recognition, broadens the realms of applicability in this study significantly. Moreover, delving into machine learning algorithms like Logistic Regression or Decision Trees can offer enhanced insights into comprehending the decision-making processes of models and interpreting the resolutions derived through them. It's crucial to continuously explore and address the challenges and advancements in machine learning methodologies like SVM, Random Forest, and Logistic Regression to enhance their efficacy and applicability in diverse research domains.

Automation's current advancements can reap great benefits from text classification applications. Such apps have the power to streamline the execution of user commands that direct machines directly. Furthermore, computer security vulnerabilities require effective policies and configuration of computer systems; intrusion detection systems play an invaluable role in detecting attacks against such vulnerabilities and mitigating attacks effectively.

Text classification faces another difficulty due to its complex feature space. Text domains often consist of features that do not directly relate to classification tasks - some features could even negatively impact classification accuracy.

Therefore, efforts should be directed toward feature selection and dimensionality reduction techniques to maximize the efficiency and effectiveness of text classification models.

REFERENCES

[1]. Yen SJ, Lee YS, Ying JC, Wu YC (2011) *A logistic regressionbased smoothing method for Chinese text categorization.* Expert Syst Appl 38(9):11581–11590.
[2]. Aseervatham S, Antoniadis A, Gaussier E, Burlet M, Denneulin Y (2011) *A sparse version of the ridge logistic regression for large-scale text categorization.* Pattern Recogn Lett 32(2):101–106. https://doi.org/10.1016/j.patrec.2010.09.023
[3]. Elghazel H, Aussem A, Gharroudi O, Saadaoui W (2016) *Ensemble multi-label text categorization based on rotation forest and latent semantic indexing*. Expert Syst Appl 57:1–11. https://doi.org/10.1016/j.eswa.2016.03.041.
[4]. Nadi A, Moradi H (2019) *Increasing the views and reducing the depth in random forest*. Expert Syst Appl. https://doi.org/10.1016/ j.eswa.2019.07.018
[5]. N. Kumar and A. Kumar, "*Australian Bushfire Detection Using Machine Learning and Neural Networks*," 2020 7th International Conference on Smart Structures and Systems (ICSSS), Chennai, India, 2020, pp. 1-7, doi: 10.1109/ICSSS49621.2020.9202238.
[6]. Tan Y (2018) *An improved KNN text classification algorithm based on K-Medoids and rough set*. In: 10th international conference on intelligent human–machine systems and cybernetics (IHMSC), pp 109–113
[7]. Szymaski J (2014) *Comparative analysis of text representation methods using classification*. Cybern Syst 45(2):180–199
[8]. Kumar R, Kaur J (2020*) Random forest-based sarcastic tweet classification using multiple feature collection.* In: Tanwar S, Tyagi S, Kumar N (eds) Multimedia big data computing for IoT applications. Intelligent systems reference library, vol 163. Springer, Singapore.
[9]. Bafna P, Pramod D, Vaidya A (2016) *Document clustering: TFIDF approach. In: 2016 international conference on electrical, electronics, and optimization techniques* (ICEEOT), Chennai, pp 61–66
[10]. Miao F, Zhang P, Jin L, Wu H (2018) *Chinese news text classification based on machine learning algorithm*. In: 2018 10th international conference on intelligent human-machine systems and cybernetics (IHMSC), Hangzhou, pp 48–51
[11]. Liu YY, Yang M, Ramsay M, Li XS, Coid JW (2011) *A comparison of logistic regression, classification and regression tree, and neural networks models in predicting violent re-offending*. J Quant Criminol 27(4):547–553
[12]. M. Kaur, C. Thacker, L. Goswami, T. TR, I. S. Abdulrahman and A. S. Raj, "Alzheimer's Disease Detection using Weighted KNN Classifier in Comparison with Medium KNN Classifier with Improved Accuracy," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2023, pp. 715-718, doi: 10.1109/ICACITE57410.2023.10183208.
[13]. S. Saxena and S. Rathor, "An Ensemble-Based Model of Detecting Plant Disease using CNN and Random Forest," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-6, doi: 10.1109/ISCON57294.2023.10112023.
[14]. P. S. Harshini, K. Naresh, S. R. Pamulapati and A. Lavanya, "Diagnosis of Liver Diseases Using Machine Learning Algorithms and their Prediction Using Logistic Regression and ANN," 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2023, pp. 1-6, doi: 10.1109/CONIT59222.2023.10205819.
[15]. G. Kumar Sahoo, K. Kanike, S. K. Das and P. Singh, "Machine Learning-Based Heart Disease Prediction: A Study for Home Personalized Care," 2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP), Xi'an, China, 2022, pp. 01-06, doi: 10.1109/MLSP55214.2022.9943373.

AUTHORS

Ravikant Kholwal - graduated from PDPM IIITDM Jabalpur, has significantly advanced intrusion detection systems, increasing accuracy by 25% through a novel algorithm. With publications in IJEAT and expertise in image classification, he's contributed to diverse projects using Django, ReactJs, and OpenCV. As a Software Engineer and CTO, he's proficient in JavaScript, React.js, AngularJS, Firebase, and is an AWS Certified Cloud Practitioner. Ravikant's blend of research and application skills distinguishes him in computing.