Date of publication JUN-30, 2025, date of current version MAR-02, 2025.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.24.2.4012

Text Document Dimensionality Reduction and Classification using R8, R21578 Data Sets and Machine Learning Models

SURESH REDDY GALI[1], SREENIVASA RAO ANNALURI[1], N SUDHAKAR YADAV[2], KRANTHI KIRAN JEEVANGAR[1], BHUVANA MANCHIKATLA[1], DHANUSH GUMMADAVALLI[1], NAGA SHIVANI KARRA[1]

[1]VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India
[2]Chaitanya Bharathi Institute of Technology, Hyderabad, India

Corresponding author: Suresh Reddy Gali (e-mail: gali.sureshreddy@gmail.com).

ABSTRACT In today's world of vast textual information, the ability to categorize documents based on their content is crucial. A text classification system that automatically assigns predefined categories to documents proves invaluable in managing the vast volume of text data. This study uses natural language processing (NLP) and machine learning and decides which algorithm generates high accuracy in classification. The main goal is to develop a system that is used to classify text documents accurately. A text classification system can make accessing a required document simple and information retrieval fast. This study describes the working of different classification algorithms and evaluates their accuracy. Before feeding the dataset to the classification models, selecting the right features is important. This study focuses on features that are crucial for classification and eliminates unnecessary words using proper preprocessing approaches. It uses information gain to select the important features. Among the considered algorithms, logistic regression has given top results with 98.49% balanced accuracy, followed by KNN with 96.81% and decision tree with 93.30%. Thus, by reducing the dimensionality of the documents before feeding them to classification models, this study aims to provide a method to classify them.

KEYWORDS Decision tree; KNN; Logistic regression; Machine learning.

I. INTRODUCTION

Within the vast fields of natural language processing and machine learning, text classification is crucial in organizing unstructured textual data. Text classification assigns each document to a class according to its content. In many ways, text classification algorithms are now a part of online life. Email classification, social media monitoring, and many more applications can benefit from text classification. In e-commerce, text categorization can offer clients a customized product based on their preferences and backgrounds. Automating patient questions based on their previous records for evaluation and treatment history can also be useful in the medical industry.

As there are many documents and each word is considered a feature, dimensions are a major issue and a challenging step in document classification. To reduce dimensions, we have traditionally used information gain and eliminated each feature.

Reducing dimensions helps reduce the complexity of textual data. Text classification is performed best when dimensionality is reduced ahead of model application. Reducing the number of features, or dimensionality, facilitates calculations and helps prevent overfitting. Furthermore, dimensionality reduction helps manage sparse data and addresses the difficulty of managing high numbers of dimensions, resulting in more accurate text classifiers [1, 2].

After eliminating stop words and stemming words, the Bag of Words approach is used to help with this. To improve the computer performance in this task, each document should be visualized as a list with the frequency of each stemmed word. This will help the computer understand the content without worrying about the order [3]. Information gain is the process used here, which determines the contribution of each word to the computer's efficiency. During the dataset's classification model training, a stratified K-fold cross-validation method is used to ensure a fair proportion of class distributions in each fold. The stratified K-fold cross-validation model evaluation is protected from biases. Now, a diverse array of classification algorithms powers document classification and text similarity tasks. Supervised learning algorithms, including KNN, logistic regression, and decision trees, utilize labelled data to learn from examples. They then adapt to sort documents using similarity measures such as Euclidean and Cosine [6]. We use these similarity measures to assess the similarity between documents.

The rest of the paper is organized as follows. Section 2 outlines the literature survey, Section 3 outlines the motivation, and Section 4 highlights the preprocessing tasks performed where dimensionality is reduced. In Section 5, various employed models are discussed in detail. Section 6 discusses in detail the methodology of this study. In Section 7, results, and descriptions of the experimental campaign are reported. Section 8 describes the conclusions obtained from the results.

II. LITERATURE SURVEY

There are many current studies conducted by researchers about text document classification using machine learning algorithms and measuring similarity using NLP methods. Here are a few examples.

In this study, authors mainly focus on reducing the dimensionality of documents. In unsupervised as well as supervised text mining, high-dimensional data is always a challenge to deal with because it increases complexity [7]. In this research, they proposed an approach to cluster the features of documents, which would result in dimensionality reduction. They compared their proposed approach with existing methods like SVD and IG [8]. They obtained an optimal matrix that is equivalent to the input high-dimensional document feature matrix. That optimal matrix had fewer features than the initial matrix [9]. The proposed method gave better results than existing dimensionality reduction methods like SVD (singular value decomposition) and IG (information gain). The method demonstrated effectiveness while preserving the initial distribution of words within each document [10, 30].

K. Torkkola's study revealed potential flaws in the use of Latent Semantic Indexing (LSI) and sequential feature selection. To address these issues, the author suggested using linear discriminant analysis (LDA) for feature transformation. The author worked with Reuters21578 and performed feature reduction and then classification [11]. To get better results, the researcher used random transformation or latent semantic indexing to reduce intermediate features. After the features were reduced by LSI or random transformation, LDA was applied, and the results were compared. Upon applying LDA, only 0.2% of features were identified as crucial. It was found that this substantial reduction in features significantly enhanced the classification accuracy. The primary objective was achieving improved classification accuracy and effective retrieval of documents within specific categories. After reducing the number of features, the support vector machine (SVM) algorithm was used for classification. This made the error rate during classification much lower [11, 31].

The authors of study [12] considered the problems that Bayes classifier faces when it comes to automated text classification and suggested two useful ways to make it work better. The study comes up with a way to normalize text for each document using a multivariate Poisson model and a feature weighting method. The goal is to fix the problems with the old I Bayes classifier. The author specifically addressed issues related to parameter estimation within the multinomial model and proposed alternative solutions [13]. Additionally, the study explored the application of the Poisson model in text classification, a relatively unpopular area.

By working on the Reuters21578 and 20 Newsgroups collections, the working of the proposed techniques is shown, particularly in categories where there is a limited number of training documents. Comparative analyses with traditional multinomial classifiers and SVM highlight the promising performance and computational efficiency of the proposed Poisson classifier [14]. Results show that the proposed approach can generate probabilistic text classifiers, but they require more time and space. When the documents are fewer in number focusing on classification weights is very effective in improving classification accuracy [15].

The authors used the BBC news dataset and tried different classification algorithms to organize the news data. Their system for classification started with pre-processing, Data representation, classification model implementation, and lastly classification [16]. The authors chose Logistic regression, random forest, and K nearest neighbors as their models. They compared the results of these models based on 5 parameters which are precision, accuracy, F1-score, support, and confusion matrix. The news was of different categories like business, sports, technology, entertainment, and politics. They observed that the Logistic regression model with TF-IDF gives the highest accuracy (97%). The second best was the random forest classifier and the last was K nearest neighbors. They considered accuracy as an important parameter, and they concluded [17].

In some studies, new similarity measures were introduced which help to find the closeness between the documents. The authors focused on dimensionality reduction and proposed a new similarity measure that can be useful for classification and clustering [18]. They used the SVM concept for dimensionality reduction and performed classification and clustering using their proposed similarity measure. Another similarity measure was introduced which performed better than existing similarity measures. They changed the binary matrix into a similarity matrix by applying the similarity measures, which they introduced in [7]. In both studies about similarity measures the authors analyzed the similarity measures by considering best-case, worst-case, and average-case scenarios [7]. All the studies observed here were helpful in creating this model. These studies have helped in deciding what aspects our study should focus on and what can be potentially improved.

III. PREPROCESSING

The Reuters-8 dataset provides a well-established platform for researchers and practitioners to develop and test text classification algorithms. Its manageable size, multi-class nature, and role in text representation exploration solidify its importance in the field. R8 has become a standard test collection for evaluating the performance of text classification algorithms. Researchers use it to compare different techniques and measure their effectiveness in categorizing documents. The R8 subset contains around 7,200 documents, making it a practical size for training and testing algorithms without overwhelming computational resources. Unlike some datasets with binary classification (spam/not spam), R8 documents belong to multiple categories (e.g., acquisition, economics), making them valuable for training multi-label classification models.

The dataset will contain a significant quantity of unnecessary data and can be highly dimensional. It is important to reduce the dimensionality of the dataset in order to reduce the complexity of the classification process [19]. The first step is handling stop words. Stop words are the most used words everywhere; they do not have any weightage in classification, so they are removed [20]. After eliminating stop words, we reduce all words to their stem form by removing prefixes, suffixes, or roots. We use the Porter Stemmer [21] to perform the stemming. Additionally, we select only alphabetic words, eliminating alphanumeric and numeric words [22].

We represent the resulting documents in matrix form for further processing. A binary matrix and a frequency matrix are used to represent the data, where the binary matrix indicates the presence of words in text files and the frequency matrix indicates the count of each feature in a certain document [23], [24]. The binary matrix is obtained from the frequency matrix by replacing non-zero fields with 1. The matrices are converted into data frames where features are columns and text file names are indices. The shape of the data frame is 7126 x 16455. Information gain is used for feature selection. The information gain for the R8 dataset (7126 x 16455) is 3.535548084867161. We sort the features in descending order based on their information gain. The features whose information gain adds up to 90% of the total information gain of all attributes are final features, and the remaining features are dropped. This ensures the retention of 90% of the dataset's information. The total information gain of all attributes is 13.9685; 90% of it is 12.5717. The shape of the data after feature selection is 7126 x 5599. Only 33.53% of total features are obtained after information gain, and they will be used for further processes. These 33.53% attributes account for 90% of the data.

IV. MACHINE LEARNING MODELS

In this section, we discuss implementation details using different types of machine learning models on the dataset. The dataset used here was obtained after preprocessing, retaining only useful data. The different models that were used are K-Nearest Neighbors (KNN), Logistic Regression and Decision tree, each having its respective uniqueness. The description of the above-mentioned models is as follows.

A. KNN

KNN is a classification-focused supervised non-parametric machine learning model [17]. The popular and versatile machine-learning technology (K-NN) method is known for its simplicity and ease of usage. There is no need to make any assumptions about how the underlying data will be distributed. Based on a distance measure, the K-NN algorithm determines a data point’s closest neighbors.

B. LOGISTIC REGRESSION

Logistic regression models are primarily made to work for binary classification where there are only two classes in the target vector. However, the problem here is multi-class classification (there are more than two classes in the target variable), and logistic regression includes expansions that make it easier to apply for multi-class classification [25]. One-vs-rest is one of the extensions that enhance logistic regression to train a distinct model for each class in comparison with all the other classes when the target vector contains more than two classes [26].

C. DECISION TREE

Decision trees are produced by iteratively dividing the dataset by features. This is a simple, yet efficient, method of classifying documents into predetermined groups according to their features. These trees are made up of leaf nodes that indicate the expected class labels and internal nodes that reflect decision points depending on document attributes like word existence or frequency [27, 32]. Decision trees are useful tools for document categorization tasks because, despite their simplicity, they can handle both continuous and categorical information and record complex decision boundaries [6, 28].

V. METHODOLOGY

We start building machine learning models on the dataset after preprocessing.

A. KNN

Steps followed while building the KNN model:

(i). The data is split into training and testing data.

(ii). The training data is split into 10 folds using a stratified K-fold to ensure each fold has instances from each class proportionally. K-fold cross-validation does not ensure the distribution of all classes in each fold. So, we had to use stratified K-fold cross-validation.

(iii). Then the data is fed to the KNN model using Cosine and Euclidean metric.

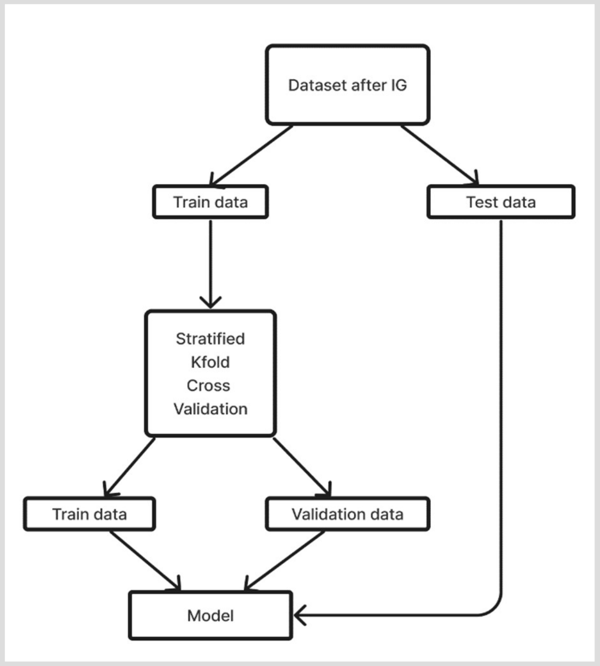

Figure 1. Flow chart

Choosing a K value is very crucial in the KNN algorithm to know how many nearest neighbors should be considered. To select a K value, we iterated the model from 1 to sqrt (no. of documents) with an offset of 2. K value which gives the highest balanced accuracy will be chosen as shown in Fig.1and Fig.2.

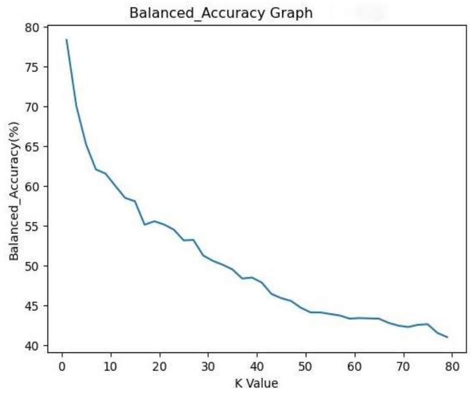

Figure 2. K value selection

B. LOGISTIC REGRESSION

The splitting of training and testing data is the same as KNN that is shown in Fig (2). In logistic regression, the probabilities will be calculated for each instance and there will be a threshold value to check if the probability of a certain instance is enough to consider it in a particular class. If the probability acquired by an instance is greater than the threshold value, then it will be categorized into a particular class or else it will be put in another class. Now this works well with binary classification where there are only two classes. For multi-class classification, we use the one-versus-rest method where a single class is compared to the remaining classes.

In logistic regression, the threshold is 0.5 by default. In variable threshold, a list of 9 different thresholds which range from 0.1 to 0.9 by an offset of 0.1 is considered. For each class, a model is trained by considering a particular class as true and remaining as false. Balanced accuracy will be observed when each model is being trained with different thresholds. The threshold with the highest balanced accuracy will be set to that respective model. By doing this we get thresholds for each class. Then the probabilities of each instance will be calculated. The calculated probabilities of each instance will be checked with the threshold values of each class. If the probability is greater than the threshold it will be considered as true for a respective model of a class.

Another method for fixing thresholds is done through specificity and sensitivity. While calculating threshold values for the classes, a graph is drawn between the sensitivity and specificity (vs) thresholds. The intersection points where specificity and sensitivity will meet are taken as the threshold for each class. Even though the thresholds were selected from the intersection of specificity and sensitivity the results were the same as shown in Fig. 3.

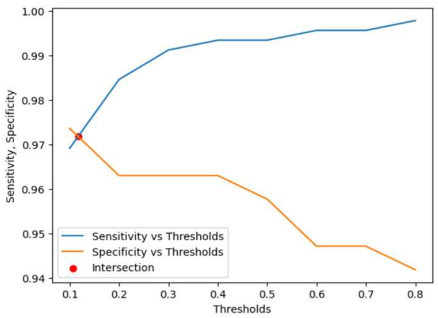

Figure 3. Intersection point of sensitivity and specificity.

If there is more than one true value, then the instance will be assigned to the class with the highest probability.

C. DECISION TREE

The C4.5 algorithm is an extension of the ID3 algorithm, and it can be used in document classification. C4.5 uses information gain to select features that help in splitting the data at each node of the decision tree [27, 28].

When applying decision trees, especially with the C4.5 algorithm, handling continuous data becomes essential. Decision trees inherently prefer categorical data, prompting the need for appropriate preprocessing. Continuous features in the dataset are often discretized, converting them into categorical variables. The data is discretized by using equal-width binning and converting it into categorical data. After that C4.5 Decision Tree algorithm is used for prediction. To use the C4.5 algorithm we have to set the parameter criterion as 'entropy 'which is an inbuilt C4.5 in Python. No other parameters are considered or adjusted while experimenting.

In Quinlan's ID3 algorithm, only Information gain was considered for dividing the attributes. However, because these qualities will give more outcomes to be added when selected as the dividing attribute, as information gain is selected as the dividing attribute it has a bias towards the attribute which has the most outcomes. Given this problem, Quinlan's C4.5 algorithm, the successor of the ID3 algorithm optimizes the information gain calculation. To take the number of results that an attribute will yield into consideration information gain is normalized in the C4.5 algorithm [1]. The normalized attribute selection metric is called the Gain ratio.

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \quad , (1)$$

where A is a given attribute in a data set D.

Information gain is normalized using Split Info. It is calculated by using the following formula:

$$SplitInfo(A) = -\sum_{j=1}^{n} p_j \log_2(p_j) \quad , (2)$$

where n = no of partitions, p_j = probability that a record in the node is in partition j.

Since Split Info increases as the number of partitions increases, it normalizes the information gain of attributes with many partitions.

VI. EXPERIMENTS & RESULTS

The datasets considered in this study are Reuters21578(R21578) and Reuters8(R8). One of the metrics used to assess a classification model's performance is balanced accuracy. It is the average of sensitivity and specificity. The proportion of a model's capacity to identify positive cases is called sensitivity, and the proportion of its capacity to identify negative cases is called specificity.

A. KNN

The KNN model is fed to the data by adding cosine and Euclidean measures. The accuracy rates observed are 94.81 and

96.81, respectively while balanced accuracies are 91.58 and 94.76, respectively. These are the results observed when the dataset is R8 as shown in Fig.4. When the R21578 dataset is trained, the accuracy observed is 85.07(cosine) and 92.84(Euclidean) and balanced accuracy is 79.84(cosine) and 88.84(Euclidean) as shown in Fig.5.

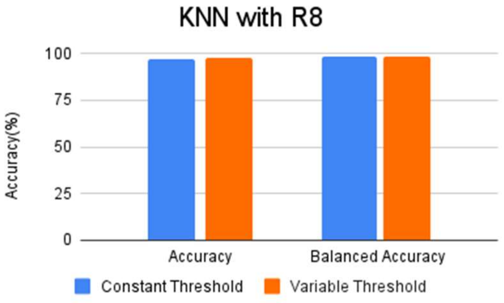

Figure 4. KNN results for R8

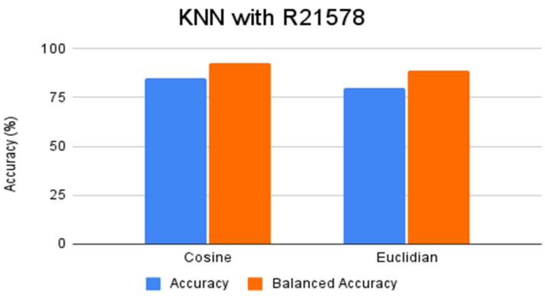

Figure 5. KNN results for R21578

B. LOGISTIC REGRESSION

Logistic regression is applied on the dataset by using constant threshold and variable threshold. When Logistic regression with a constant threshold is applied to the R8 dataset, the accuracy is observed to be 96.91 and the balanced accuracy is 98.17. When an array of thresholds (in variable threshold) is used, the accuracy and balanced accuracy obtained are 97.47 and 98.49 as shown in Fig. 6.

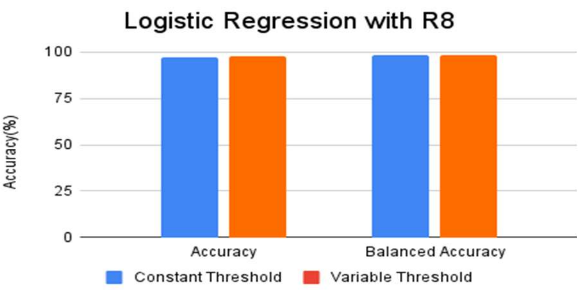

Figure 6. Logistic regression for R8

A balanced dataset is referred to as the dataset where all categories have an equally distributed number of input samples. When balanced R8 is fed to logistic regression with a constant threshold, the accuracy and balanced accuracy obtained are 89.13 and 90.55. But when the threshold is changing, the accuracy and balanced accuracy were observed to be 90.19 and 89.13. The R21578 is trained with logistic regression with a constant threshold, and the accuracy observed is 90.41, while the balanced accuracy is 94.50. When logistic regression with standard scaling is trained, the accuracy and balanced accuracy obtained are 82.11 and 89.00, respectively as shown in Fig.7. The variable threshold for R21578 is not a recommended idea, as the dataset has 48 classes and 48 models that must be trained. In testing, each instance gets a probability from each class, and one class will be selected based on probability. Hence, it is not considered as an efficient approach since this consumes an enormous amount of time.

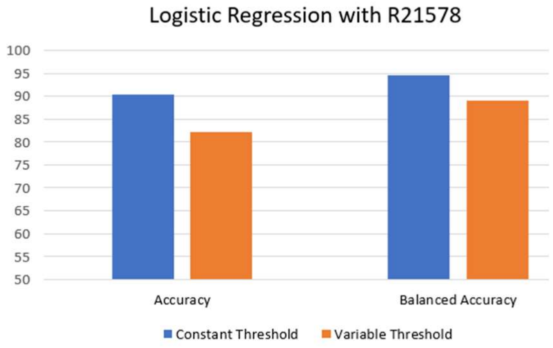

Figure 7. Logistic Regression results for R21578

C. DECISION TREE

The decision tree is performed on the R8 dataset by using the C4.5 algorithm. Before performing discretization, the accuracy and balanced accuracy are observed as 89.78 and 93.30. After discretization, the accuracy and balanced accuracy are observed as 94.24 and 91.16. For R21578, before discretization, the results are 74.38 and 81.20. After discretization, the results are 76.77 and 86.50 as shown in Fig.8 and Fig.9.

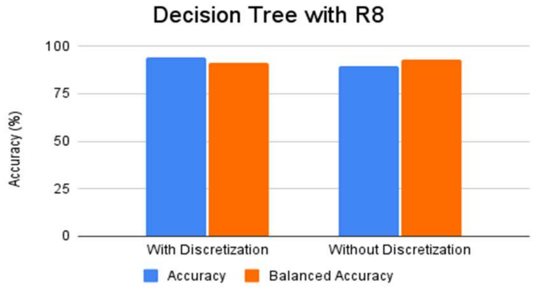

Figure 8. Decision Tree for R8

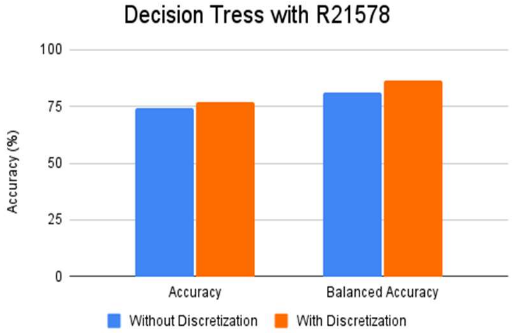

Figure 9. Decision Tree for R21578

VII. DISCUSSIONS

In this study, a comparison is made between three different classification algorithms, which are KNN, Logistic Regression, and Decision tree. These three are applied to two variants of the Reuters dataset which are Reuters21578 and Reuters8. Among these three algorithms, it is observed that Logistic Regression gives the best performance. When R8 dataset is fed to the Logistic Regression model the accuracy and balanced accuracy are observed as 97.47 and 98.49 as shown in Fig.10.

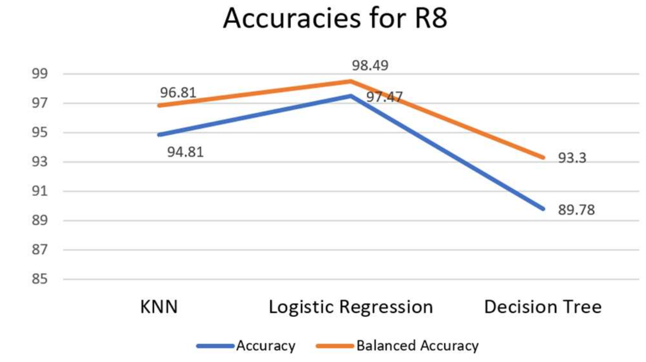

Figure 10. Comparison of all models for the R8 dataset

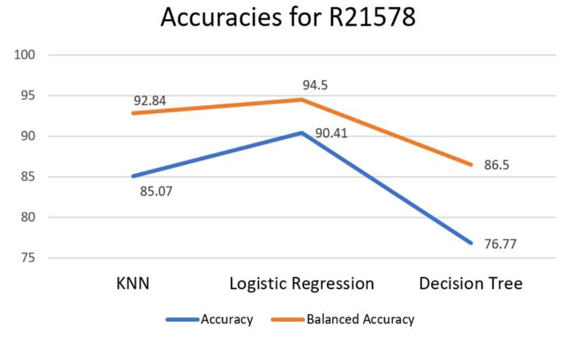

Figure 11. Comparison of all models for the R21578 dataset

When the Reuters21578 dataset is used, the results are high for Logistic Regression. Logistic Regression gave the best results when standard scaling was used. The accuracy and balanced accuracy of the Logistic Regression model are 90.41 and 94.50 as shown in Fig.11. Logistic Regression has given better accuracy while standard scaling is used. The accuracy and balanced accuracy of Logistic regression using a constant threshold without standard scaling are 90.41 and 95.40. The use of standard scaling significantly improved the model's performance, highlighting the importance of feature normalization in text classification. Logistic Regression proved to be effective across various preprocessing techniques, consistently achieving high accuracy and balanced accuracy. The model's consistency in achieving high accuracy suggests its effectiveness in handling the Reuters21578 dataset.

Table 1. Results of all classification algorithms

Dataset	Algorithm		Accuracy	Balanced Accuracy	f1-score	Precision	Sensitivity	Specificity
R8	KNN	Euclidian	91.58	94.76	91	92	92	97.93
		Cosine	94.81	96.81	95	95	95	98
	Logistic Regression	Constant Threshold	96.91	98.17	0.97	0.97	0.97	0.99
		Variable Threshold	**97.47**	**98.49**	0.97	0.98	0.97	0.99
	Decision Tree	Without Discretization	89.78	93.30	90	90	90	96
		With Discretization	94.24	91.16	91	91	91	97
R21578	KNN	Euclidian	79.84	88.84	0.79	0.79	0.80	0.97
		Cosine	85.07	92.84	0.85	0.86	0.86	0.98
	Logistic Regression	Constant Threshold	**90.41**	**94.50**	0.90	0.90	0.90	0.99
		Variable Threshold	82.11	89.00	0.83	0.85	0.82	0.96
	Decision Tree	Without Discretization	74.38	81.20	0.76	0.77	0.77	0.96
		With Discretization	76.77	86.50	0.76	0.76	0.77	0.96

The R8 dataset yields higher performance metrics across all algorithms, with accuracies ranging from 89.78% to 97.47%, compared to 74.38% to 92.84% for R21578. This disparity underscores R8's relative simplicity, allowing models to achieve near-optimal results with less tuning. In contrast, R21578 demands careful algorithm selection and configuration, particularly to address potential class imbalances, as seen with Cosine KNN's superior balanced accuracy despite its lower overall accuracy, as shown in the Table 1.

Logistic regression emerges as the most consistent performer, and it has achieved top results on R8 (Variable Threshold) and strong outcomes on R21578 (Constant Threshold). Its adaptability to the threshold strategies highlights its robustness across the two dataset types considered. KNN shows dataset-specific behavior: Cosine similarity boosts performance on R8 but falters in accuracy on R21578, though it compensates with better minority class detection. This implies that dataset properties, like feature distribution or class balance, should guide the selection of similarity measures. Decision Tree benefits from discretization in both datasets, improving accuracy and, in R21578, balanced accuracy. This preprocessing step appears critical for tree-based models, particularly when dealing with complex or noisy data. Every model has its own way to classify the given data depending on the density of the dataset, and the observations have proved that point.

VIII. CONCLUSION

Modification of the traditional logistic regression and improvement of the model accuracy is described in this paper. We applied these methods with two datasets: Reuters-8 and Reuters-21578. Logistic regression showed the best results with 98.49% of balanced accuracy, followed by KNN with 96.81% and decision trees with 93.30%.

Conclusion is drawn that by using these methods, we could increase the accuracy of the model. In the increasingly vital fields of machine learning and data mining, classification is an essential problem. We must be able to collect and understand relevant information from the continually growing amount of data that we produce reliably and efficiently. The F1 score, precision, recall, and specificity measures, which are properly reported along with accuracy and balanced accuracy, show how thoroughly the algorithm is reviewed and proven.

Each of the indicators in Table 1 shows a deep understanding of how difficult categorization tasks can be. This allows for a full evaluation of the model's effectiveness. By carefully using dimensionality reduction techniques before fitting the models, the algorithms have successfully broken through traditional boundaries and performed to their full potential. This strategy has improved overall performance and opened the door for developments in categorization techniques. We modify the classification algorithms to achieve maximum accuracy during dataset training. Out of all the algorithms considered in this study, logistic regression gave maximum accuracy by considering different thresholds for each category, whereas the decision tree gave the least accuracy because it couldn't handle all the conditions while assigning categories to text documents.

References

[1] R. Devakunchari, "Analysis on big data over the years," 2014. [Online]. Available at: https://www.semanticscholar.org/paper/Analysis-on-big-data-over-the-years-Devakunchari/d0401ba28280625e22b6e1cbe0b43af8d5e9ad93.

[2] T. T. Dien, B. H. Loc, and N. Thai-Nghe, "Article classification using natural language processing and machine learning," *Proceedings of the 2019 IEEE International Conference on Advanced Computing and Applications (ACOMP)*, Nha Trang, Vietnam, Nov. 2019, pp. 78–84. https://doi.org/10.1109/ACOMP.2019.00019.

[3] R. Angelova and G. Weikum, "Graph-based text classification: learn from your neighbors," Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle Washington, USA, Aug. 2006, pp. 485–492. https://doi.org/10.1145/1148170.1148254.

[4] A. Chahar, N. Patil, D. Walunj, Sai Rohith T, R. Shah, and H. Saratkar, "An Indispensable Contemplation on Natural Language Processing Using Ensemble Techniques for Text Classification," *Proceedings of the 2022 8th IEEE International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, Mar. 2022, pp. 406–410. https://doi.org/10.1109/ICACCS54159.2022.9785015.

[5] K. M. Chaitrashree, T. N. Sneha, S. R. Tanushree, G. R. Usha, and T. C. Pramod, "Unstructured medical text classification using machine learning and deep learning approaches," *Proceedings of the 2021 IEEE International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, Bangalore, India, Aug. 2021, pp. 429-433. https://doi.org/10.1109/RTEICT52294.2021.9573667.

[6] F. Gorunescu, "Classification and decision trees," in Data Mining: Concepts, Models and Techniques, F. Gorunescu, Ed., Berlin, Heidelberg: Springer, 2011, pp. 159–183. https://doi.org/10.1007/978-3-642-19721-5_4.

[7] G. SureshReddy, T. V. Rajinikanth, and A. A. Rao, "Design and analysis of novel similarity measure for clustering and classification of high dimensional text documents," *Proceedings of the Proceedings of the 15th ACM International Conference on Computer Systems and Technologies*, Ruse, Bulgaria, Jun. 2014, pp. 194–201. https://doi.org/10.1145/2659532.2659615.

[8] Pm. Lavanya and E. Sasikala, "Deep learning techniques on text classification using natural language processing (NLP) in social healthcare network: A comprehensive survey," *Proceedings of the 2021 3rd IEEE International Conference on Signal Processing and Communication (ICPSC)*, Coimbatore, India, May 2021, pp. 603–609. https://doi.org/10.1109/ICSPC51351.2021.9451752.

[9] H. Guan, B. Xiao, J. Zhou, M. Guo, and T. Yang, "Fast dimension reduction for document classification based on imprecise spectrum analysis," *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, Toronto, Canada, Oct. 2010, pp. 1753–1756. https://doi.org/10.1145/1871437.1871721.

[10] G. S. Reddy, "Dimensionality reduction approach for high dimensional text documents," *Proceedings of the 2016 IEEE International Conference on Engineering & MIS (ICEMIS)*, Agadir, Morocco, Sep. 2016, pp. 1–6. https://doi.org/10.1109/ICEMIS.2016.7745364.

[11] K. Torkkola, "Discriminative features for text document classification," *Form. Pattern Anal. Appl.*, vol. 6, no. 4, pp. 301-308, 2004. https://doi.org/10.1007/s10044-003-0196-8.

[12] Z. Li et al., "A unified understanding of deep NLP models for text classification," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 12, pp. 4980–4994, 2022. https://doi.org/10.1109/TVCG.2022.3184186.

[13] R. Mao, W. L. Miranker, and D. P. Miranker, "Dimension reduction for distance-based indexing," *Proceedings of the Third ACM International Conference on SImilarity Search and APplications*, Istanbul, Turkey, Sep. 2010, pp. 25–32. https://doi.org/10.1145/1862344.1862349.

[14] J. Kolluri, S. Razia, and S. R. Nayak, "Text classification using machine learning and deep learning models," *SSRN Electron. J.*, 2020. https://doi.org/10.2139/ssrn.3618895.

[15] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng, "Some effective techniques for naive Bayes text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 11, pp. 1457–1466, Nov. 2006. https://doi.org/10.1109/TKDE.2006.180.

[16] G. Pang, H. Jin, and S. Jiang, "An effective class-centroid-based dimension reduction method for text classification," *Proceedings of the 22nd ACM International Conference on World Wide Web*, Rio de Janeiro, Brazil, May 2013, pp. 223–224. https://doi.org/10.1145/2487788.2487903.

[17] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and KNN models for the text classification," *Augment. Hum. Res.*, vol. 5, no. 1, p. 12, 2020. https://doi.org/10.1007/s41133-020-00032-0.

[18] Y. V. Singh, P. Naithani, P. Ansari, and P. Agnihotri, "News classification system using machine learning approach," *Proceedings of the 2021 3rd IEEE International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, Dec. 2021, pp. 186–188. https://doi.org/10.1109/ICAC3N53548.2021.9725409.

[19] D. Patil, R. Lokare, and S. Patil, "An overview of text representation techniques in text classification using deep learning models," *Proceedings of the 2022 3rd IEEE International Conference for Emerging Technology (INCET)*, Belgaum, India, May 2022, pp. 1–4. https://doi.org/10.1109/INCET54531.2022.9825389.

[20] Y. Zheng, "An exploration on text classification with classical machine learning algorithm," *Proceedings of the 2019 IEEE International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, Taiyuan, China, Nov. 2019, pp. 81–85. https://doi.org/10.1109/MLBDBI48998.2019.00023.

[21] S. K. Mohapatra, P. K. Sarangi, P. K. Sarangi, P. Sahu, and B. K. Sahoo, "Text classification using NLP based machine learning approach," *Proceedings of the International Conference on Recent Innovations in Science and Technology (RIST'2021)*, Malappuram, India, 2022, p. 020006. https://doi.org/10.1063/5.0080301.

[22] H. Sharma, "Improving natural language processing tasks by using machine learning techniques," *Proceedings of the 2021 5th IEEE International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India, Oct. 2021, pp. 1–5. https://doi.org/10.1109/ISCON52037.2021.9702447.

[23] Y. Zhang et al., "Weakly supervised multi-label classification of full-text scientific papers," *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Long Beach, USA, Aug. 2023, pp. 3458–3469. https://doi.org/10.1145/3580305.3599544.

[24] K. Taneja and J. Vashishtha, "Comparison of transfer learning and traditional machine learning approach for text classification," *Proceedings of the 2022 9th IEEE International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, Mar. 2022, pp. 195–200. https://doi.org/10.23919/INDIACom54597.2022.9763279.

[25] D. Nunes De Oliveira and L. H. D. C. Merschmann, "An Auto-ML approach applied to text classification," *Proceedings of the ACM Brazilian Symposium on Multimedia and the Web*, Curitiba, Brazil, Nov. 2022, pp. 108–116. https://doi.org/10.1145/3539637.3557054.

[26] D. Weisburd, D. B. Wilson, A. Wooditch, and C. Britt, "Logistic regression," *Advanced Statistics in Criminology and Criminal Justice*, Cham: Springer International Publishing, 2022, pp. 127–185. https://doi.org/10.1007/978-3-030-67738-1.

[27] J. R. Quinlan, *C4.5: Programs for Machine Learning. in the Morgan Kaufmann Series in Machine Learning*, San Mateo, California: Morgan Kaufmann Publishers, 2014.

[28] A. Rizka, S. Efendi, and P. Sirait, "Gain ratio in weighting attributes on simple additive weighting," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 420, p. 012099, 2018. https://doi.org/10.1088/1757-899X/420/1/012099.

[29] A. Rizka, S. Efendi, and P. Sirait, "Gain ratio in weighting attributes on simple additive weighting," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 420, p. 012099, 2018ю https://doi.org/10.1088/1757-899X/420/1/012099.

[30] D. Singh, A. Bhure, S. Mamtani and C. Krishna Mohan, "Fast-BoW: Scaling bag-of-visual-words generation," *Proceedings of the British Machine Vision Conference (BMVC)*, 2018, pp. 287.

[31] P. Jeripothula, C. Vishnu, and C. Krishna Mohan, "Attentive contextual network for image captioning," *Proceedings of the IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, 2021, pp. 1-8. https://doi.org/10.1109/IJCNN52387.2021.9533970.

[32] E. P. Ijjina and C. Krishna Mohan, "Human action recognition in RGB-D using motion sequence and deep learning," *Pattern Recognition*, vol. 72, pp. 504-516, 2017. https://doi.org/10.1016/j.patcog.2017.07.013.

***Suresh Reddy Gali**, an accomplished academician holding a Ph.D. in Computer Science and Engineering, serves as a Professor in the Department of Information Technology. With over 27 years of teaching experience and 15 years dedicated to research. His teaching interests encompass diverse domains including Computer Forensics, Cyber Security, Data Mining, and Advanced Computing. He actively contributes to curriculum development and has organized numerous technical events within the department. His research contributions focus on areas such as Data Mining and Computer Forensics, with 35 publications in esteemed journals and conferences. He supervises numerous undergraduate and postgraduate projects and engages in sponsored research endeavors. Acknowledged for his scholarly achievements, he has received accolades such as the Best Paper Award.*

***Sreenivasa Rao Annaluri**, an accomplished Assistant Professor in the Information Technology department, boasts over 24 years of teaching experience coupled with a decade-long research tenure. With a Ph.D. in Computer Science and qualifications including an M.Tech and MCA, his teaching interests span a wide array of subjects, from Database Management Systems to Artificial Intelligence. He has actively contributed to curriculum design and updating, emphasizing practical learning through industrial visits and hackathons. His research contributions extend to publications in esteemed journals and conferences, focusing on topics like data mining and machine learning.*

***Dr. N. Sudhakar Yadav** is an Associate Professor in the Department of Information Technology, CBIT at Hyderabad, India. He has 15 years of teaching experience in various reputed institutions. He obtained his Ph.D in Computer Science and Engineering from Jawaharlal Nehru Technological University, Anantapur, Andhra Pradesh, India. His research is mainly focused on Big Data, IoT and Machine Learning. His papers have been published in various International Journals. He has more than 20 international journals and conferences.*

***Kranthi Kiran Jeevangar** is a B.Tech. student majoring in Information Technology at VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India. He is deeply engrossed in researching Educational Data Mining and Machine Learning. His areas of specialization encompass data mining and machine learning. He demonstrates proficiency in Python and Java.*

***Bhuvana Manchikatla** is a B.Tech. (Information Technology) student at VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India. She is deeply fascinated by the potential of data mining techniques to uncover valuable insights and patterns from large datasets. Her area of specialization is web development and area of interest in languages are Java, Python, Reactjs.*

***Dhanush Gummadavalli** is a B.Tech. (Information Technology) student at VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India. His research interest includes Educational Data Mining and Machine Learning. His area of specialization is web development, cloud and his area of interest is Python, Java, Reactjs and Nodejs.*

***Karra Naga Shivani** enrolled in the B.Tech program for Information Technology at VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India, she is deeply engaged in researching Educational Data Mining and Machine Learning. Her keen interests lie in exploring the realms of Python and Java for programming.*