International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(10s) 2026 pp. 1224-1243
https://doi.org/10.70917/ijcisim-2026-3710
Article

Feature-Optimized Machine Learning Framework for Multiclass Kashmiri Text Classification: A Comparative Analysis

Masooda Masarat[1*], Rumaan Bashir[2]

[1*] Department of Computer Science, Islamic University of Science & Technology, Kashmir.
[2] Department of Computer Science, Islamic University of Science & Technology, Kashmir.
Email: rumaan.bashir@islamicuniversity.edu.in
Corresponding author: **Masooda Masarat, Email: masooda.masarat@iust.ac.in*

Abstract: Classifying text in low resource languages poses significant difficulties because of the sparsity of annotated data sets and morphological complexity of low resource languages. This paper fixes this deficiency by establishing a strong machine learning system to perform multi-classification of Kashmiri text. The main objective of this study is the evaluation of the conventional methods called as machine learning techniques on the large scale, self-curated Kashmiri corpus consisting of about 27,000 sentences in nine semantic categories. The data is systematically preprocessed to normalize and tokenize it, remove noise and then using the feature extraction approach that quantifies term importance using frequency within documents and inverse corpus-level distribution called as (TF-IDF) having the vocabulary size 5000. This approach includes the implementation and optimization of five supervised learning algorithms, such as linear classification models, Bayesian probabilistic classifiers, distance-based instance learning technique, maximum-margin classifiers and ensemble learning approaches based on decision trees with the help of GridSearchCV and RandomisedSearchCV with k-fold cross-validation. F1-score, precision, Accuracy, recall and Receiver Operating Characteristic-Area Under Curve (ROC-AUC) are used to assess the models and are backed-up with analysis of the confusion matrix. The findings show that SVM has the best classification accuracy of 93%, which shows that its performance is outstanding in comparison to other models. Results demonstrate the usefulness of optimization of the features representation and model-tuning in low-resource context and offer a predictable baseline to the future evolution of natural language processing investigations in the Kashmiri language.

Keywords: SVM, Kashmiri language, Logistic Regression, Precision, Recall

1. INTRODUCTION

Natural Language Processing (NLP) has become a core research field of artificial intelligence, and it allows machines to read and comprehend human language, as well as produce it [1]. As digital communication sites continue to expand, a large amount of textual information created systematically on regular basis as news stories, social media platforms, emails, as well as Internet documents [2]. Automatic organization and interpretation of this information have been particularly significant in the applications of sentiment analysis, information retrieval, filtering of spams, detection of topic and classification of document. Text classification is one of such tasks which is crucial in attaching labels or categories to textual data which have been previously established and thus make the management of the data and the discovery of knowledge to be efficient [3]. In the last 10 years, it has progressed to great heights in text classification of high-resource languages like English, Chinese as well as Spanish. Conventional supervised learning paradigms like Bayesian probabilistic classifiers, maximum-margin classifier, and logistic link function have proven to be very effective when applied in conjunction by using the feature extraction approach that quantifies term importance using frequency within documents and inverse corpus-level distribution called as (TF-IDF) as a statistical method [4].

Recently, artificial neural network designs and models called as transformer-based have also enhanced the accuracy in classifications. But such developments are highly focused on the language where there are numerous annotated datasets and computationally developed resources [5]. Conversely, most of the regional and indigenous languages are underrepresented in NLP studies because of lack of labelled corpora, low number of online materials and absence of standard linguistic tools. These languages are generally termed as the low-resource languages. Linguistic groups with resources and those without have a growing digital gap since no benchmark datasets or baseline models exist to create a robust language technology [6]. One of these low-resource languages is the Kashmiri language that is used by millions of individuals in the valley of Kashmir. Computational research in Kashmiri is, however, immature, in spite of its cultural and linguistic importance. The language has distinctive morphological patterns, sophisticated syntax, and various scripts, which makes it even difficult to process the text automatically [7]. Moreover, this is complicated because annotated datasets are not publicly available to support the development and testing of classification systems.

Consequently, the standardization of datasets and the development of baseline procedures of Kashmiri NLP work is in urgent need. Machine learning solutions provide a viable and feasible solution to low-resource cases. In comparison with deep neural models, the classical machine learning methods use smaller databases, less computational resources, and less complex training process yet offer competitive results [8]. SVM and Naive Bayes are among the best methods to use when dealing with high dimensional sparse text representations that are produced by TF-IDF features. Thus, such models are good beginning points to constructional systems of the new languages.

Inspired by these observations, this paper aims at creating a labelled Kashmiri text classification dataset and performing a systematic comparison of popular machine learning classifier. This is aimed at creating good baseline results and proving that automated topic classification of Kashmiri text in a possible way [9]. Offering the dataset and experimental analysis, this work will be helpful in future research and motivate the further advancement of language technologies in the community of Kashmiri.

The principal contribution of this study is summarized below:

- Construction of an annotated Kashmiri sentence-level corpus of nine domain categories.
- The overall evaluation of five classical machine learning classifiers is compared.
- It provides a baseline of future low-resource NLP research.

The remainder of the paper is modeled into a couple of coherent chunks, and it will begin with a related works to be able to ascertain the state of things of the research and the gaps that were left behind by the literature. It will be then detailed by how the dataset has been prepared and constructed, how the data collection and preprocessing has been carried out. The layout of the proposed computational framework will then be presented with the focus being on the aspects of architectural design and implementation that will be proposed. Fig 1. Shows the NLP in some high and low-resource languages.

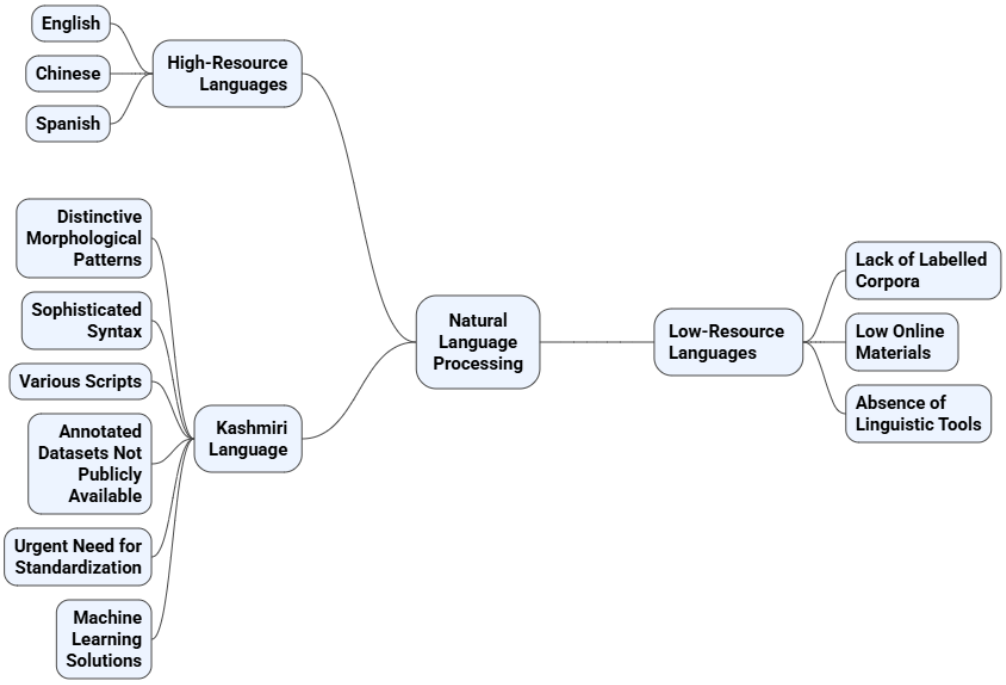

Fig 1. NLP in some High and Low Resource Languages.

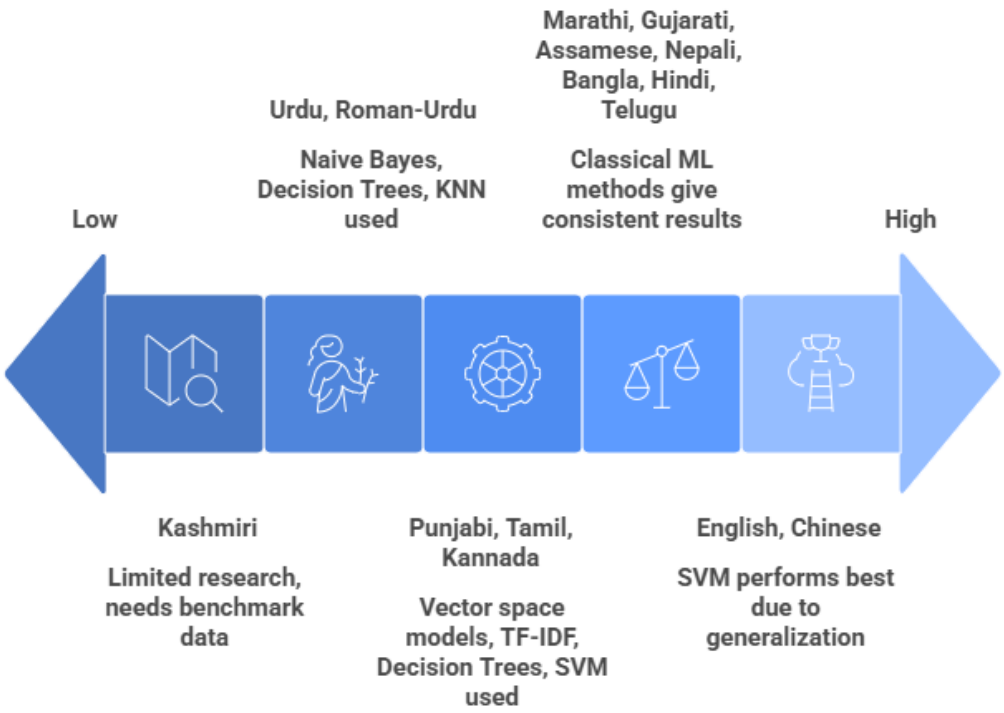

Fig 2. Various ML models used in some high and low resource languages.

2. RELATED WORKS

Text classification work has been widely tackled using the traditional techniques of machine learning by transforming textual data into quantitative feature illustration of the text, which includes feature weighting approach that quantifies term importance using document-level frequency and inverse corpus-level rarity called as (TF-IDF) and a frequency-based vector representation that model's text as an unordered collection of lexical items called as (BoW) [10]. Some of the commonly used algorithms are margin-optimized classifier, logistic -based classification, instance-driven methods and ensemble tree models were used because of being computationally efficient, simple and applicable to dimensionally high sparse text data. SVM is one of these approaches that have performed best due to its good generalization property and the capability to build optimum separating hyperplanes [11]. A number of researches have also tested classical supervised learning methods in classification of documents in other languages. According to Luo (2021), SVM, Naive Bayes, and the linear models were compared in terms of English text classification and the SVM resulted in the highest accuracy [12]. Saigal and Khanna (2020) used KNN, SVM, and ARAM classifiers to categorize Chinese news as well, and once again they found that SVM was worse than competing methods [13]. Other experiments in Persian and Arabic text classification problems have also yielded identical results as the features of TF-IDF used with SVM or Naive Bayes to give good results at a very low cost. In Indian and regional languages, classical machine learning techniques still assume a major role. In the case of Urdu and Roman-Urdu texts, Decision Trees, KNN and Naive Bayes as classifiers have been effectively valued in sentiment and topic classification [14]. Naive Bayes and hybrid feature-based methods have been used to categorize Punjabi documents, Tamil and Kannada texts have been categorized by using the vector space models, TF-IDF weighting, and supervised learning methods such as SVM and Decision Trees. Research in languages such as Marathi, Gujarati, Assamese, Nepali, Bangla, Hindi and Telugu, has also shown that classical ML methods give consistent and competitive results at moderately large datasets and with limited computational capabilities [15].

Despite these attempts, the research on Kashmiri is very scarce in relation with applicability of methods known as machine learning to some of the regional languages [16]. The existing literature has been more software and language resource preparation with little attention given to the preparation of labelled corpora or testing automated classification systems [17]. Based on the review of the literature conducted, no systematic research has compared multiple machine learning classifiers on Kashmiri text categorization with standardized evaluation metrics [18]. Accordingly, there is a definite gap in research in creating benchmark data and base classification models to Kashmiri. Present work attempts to advance the field by creating a manually annotated Kashmiri corpus and conducting a thorough analysis of comparatively and commonly used classifiers of machine learning to lay the groundwork of future research on Kashmiri Natural Language Processing [19].

Table 1: Techniques and Dataset used for the classification of text in different languages.

Authors	Year	Dataset	Techniques Used	Key Findings
Talukdar, C., & Sarma, S. K. [20]	2023.	Assamese text dataset	CNN, LSTM, Hybrid (LSTM + SVM), Hybrid (CNN + LSTM, i.e., C-LSTM-ATC)	C-LSTM-ATC model achieved the best results with 97.2% accuracy, 95% F1-score, 94% precision, and 95% recall
Rakholia and Saini [21]	2023	280 documents from different domains.	Naïve Bayes (NB) with, without feature selection, cross-validation of k-fold.	88.96% with feature selection and accuracy of 75.74% without feature selection; Gujarati text classification is efficient in this.
Z., Naqvi, M Abdel majeed I. R., M. P., Jiangbin, Akhter, M., & Fayyaz [22].	2022	Urdu Text Documents	CNN, CLSTM, LSTM AND BiLSTM	CNN performed best than other models.
R. K., Rajesh, N. & Vallapuneni, J., Ande, P. K., Indurthi & Gampala, V., [23]	2021	Telugu news articles (approx. 800 documents from 4 categories)	Naïve Bayes (NB) classifier with TF-IDF weighting scheme	Supervised classification using NB with normalized TF-IDF was effective for Telugu text categorization
Saigal, P., & Khanna, V [24]	2020	Chinese texts	KNN, SVM and ARAM	SVM outperformed other models.

Authors	Year	Dataset	Techniques Used	Key Findings
Elnagar, A., Al-Debsi, R., & Einea, O [25]	2020	SANAD and NADIA	HANGRU and CNN	CNN achieved highest accuracy of 70.34% for NADIA
Boeker, M, R., Thomczyk, F., Jaravine, V., Tippmann, P., & Scheible [26],	2020	145GB plain text corpus	GottBERT (RoBERTa-based German language model)	GottBERT outperformed other tested models.
Goel, P., & Joshi, R [27]	2020	Hindi text dataset	weighted n-gram methods and Traditional n-gram.	Utilized for analysis of text of Hindi sentiments.
Bolaj, Kim et al [28]	2019	English sentences	CNN	For sentence classification CNN-based architecture was proposed.
Al-Harbi et al. [29]	2021	1,71,721 Arab Text Documents	Stop words list, feature extraction, weighting schemes	Developed Arabic Text Categorization (ATC) tool
Nidhi & Vishal Gupta [30]	2022	Punjabi Text Documents	A combination of Ontology and Navie Bayes techniques.	Hybrid method outperforms better than Navie Bayes and Ontology.
Rajan et al. [31]	2019	Tamil Text Documents.	VSM, Artificial Neural Network (ANN).	Achieved 93.33% accuracy using ANN on Tamil Text dataset.
Tummalapalli et al [32]	2018	Translated datasets for Hindi and Telugu texts	LSTM, Multi-Input CNN, basic CNN SVM using n-gram features.	CNN-based models outperformed LSTM and SVM; focused on capturing morphological variations using character-level and word level features.
Nidhi & Vishal Gupta [33]	2016	Punjabi Text Documents	A combination of Ontology and Navie Bayes techniques.	Hybrid method outperforms better than Navie Bayes and Ontology.
Khan [34]	2021	OCR of Urdu text dataset	Decision Tree with Hu moments	Using Hu moments with Decision Tree classifier, best recognition accuracy was obtained.
Jaydeep Jalindar and Nagaraju Bogiri [35]	2015	200 documents from 20 categories	Based on VSM Label Induction Grouping (LINGO) algorithm was used	Based on user profiles including the user's browsing history, LINGO method demonstrates effective performance clustering for Marathi textual documents.
Kabir et al [36]	2015	Text documents of Bangla from 9 different domains.	Stochastic Gradient Descent (SGD)	Achieved 93.85% accuracy; slightly better performance than previous works.

3. DATASET DESCRIPTION

In order to perform supervised text classification on the Kashmiri language, an annotated corpus at sentence level was prepared. A total of 27000 sentences were created using various tools and a sample of 6208 sentences were experimented as a small test first. The Perso-Arabic script was used in transcribing the sentences and checked by native speakers and language professionals to be linguistically correct and consistent in semantics [37]. All the sentences were given different label of the class which is one of the nine categorizations. The last dataset will have two items; the Kashmiri sentences and the class label. Pre-processing of the corpus involved normalization, deletion of punctuations, stop-wording removal and tokenizing to clean and standardize the corpus [38]. To perform an

experimental assessment, self-created dataset was divided into predominant portion (80%) of training data and the rest was retained for testing purpose (20%), which made it possible to rely on the evaluation of the model generalization. The summarized corpus is elaborated as given below in fig 3.

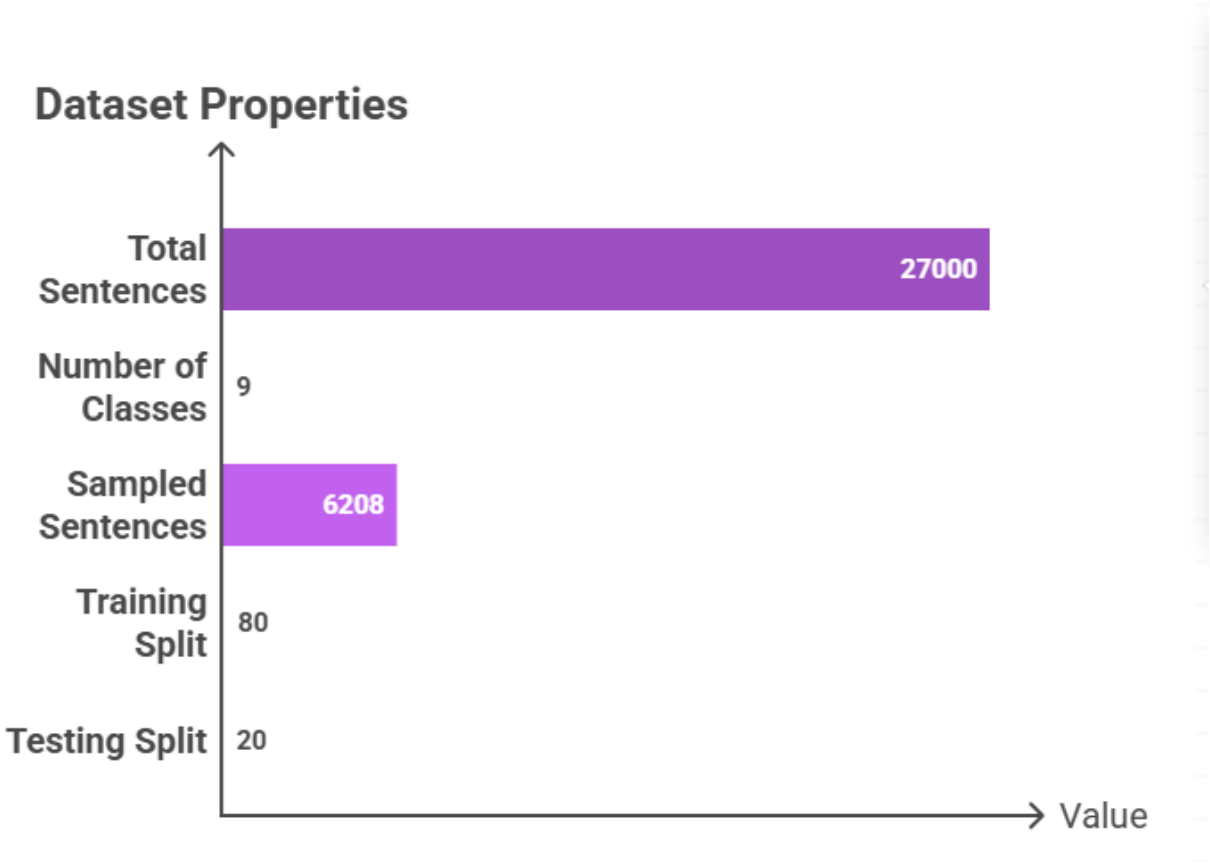

Fig 3. Dataset Summary

4. COMPARISON WITH OTHER STUDIES

All the five classifiers are compared in a comparative analysis provided in Fig.4, in which the summaries of overall accuracies of the rival methods are provided. The model called as Support Vector Machine (SVM) shows of 93 % best classification rate among the models analyzed while Logistic Regression had 92% and Multinomial Naive Bayes 91% respectively. Random Forest showed relatively lower results of 89% while K-nearest neighbor shows 84-86%and respectively [39]. The observation from results can be dedicated to the underlying parameters that enhanced accuracy of Logistic Regression and SVM is validated by the fact that they are capable of building linear decision boundaries that are capable of separating high-dimensional sparse TF-IDF feature spaces. These are the linearized models that can be used in the case of text classification where the discriminative word distributions are very separable by class [40]. On the same note, the competitive performance of Naive Bayes is an indication that, when term independence assumptions are more or less true, probabilistic methods are still effective. Distance-based and tree-based algorithms like KNN and Random Forest, in turn, demonstrated rather lower accuracy [41]. This is explained by the fact that when dealing with text data, distance measures and hierarchical splits are sensitive to sparsity and high dimensionality, and thus their ability to generalize may be constrained. Altogether the experimental results indicate that classical machine learning methods are very powerful and computationally efficient baselines to use in the Kashmiri text classification [42]. The findings confirm that linear and probabilistic models with the use of TF-IDF representations are especially appropriate in low-resource language cases when there is a limitation in data availability and computing resources [43]. These have been the baselines that can be used as a baseline in future developments of Kashmiri Natural Language Processing [44]

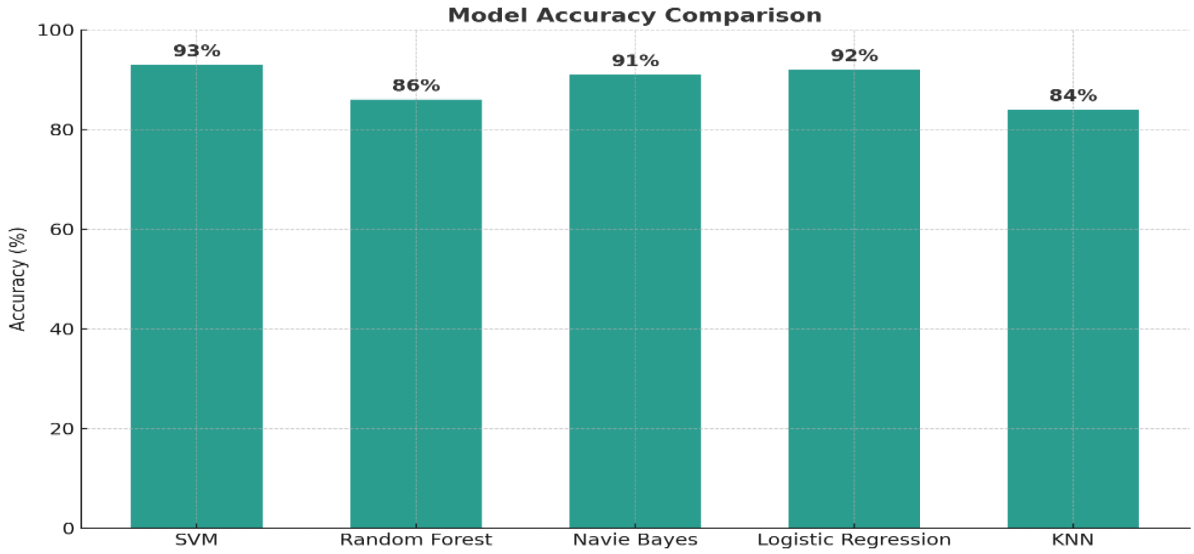

Fig 4. Model comparisons

5. METHODOLOGY

Five supervised algorithms were analyzed in terms of their effectiveness and their discriminative power of methods used in classical learning of machines in low-resource text classification. These are adequately suited to sparse TF-IDF feature space and have lower computational costs.

- Support Vector Machine (SVM): This algorithm builds an optimum plane of separation which augments class separation in high-dimensional space [45]. It is also well known in its ability to generalize well and has always performed better when used in categorizing texts.
- Random Forest (RF): This algorithm uses a set of decision trees which are trained separately and these tree-based estimates then combined to maximize the generalization trained on data subsets based on the random features [46]. This will minimize overfitting and enhance the stability of classification [47].
- Multinomial Naive Bayes (NB): A classifier which is probabilistic in nature and makes an assumption of independence within the set of attributes and Bayes theorem is used in this [48]. From the perspective of the text classification, naive bayes acts an effective text classifier and has become a popular baseline because it is compatible with term-frequency features [49]. Equation(a) below explains the Bayesian Probability Rule [44].

$$P(x|y) = P(y|z) * P(z) \div P(y)(a) \quad \text{(a)}$$

Here:

➢ P (x | y) represents the probability of a sample to be assigned to a class given the observed features y.

➢ P (y | z) represents the probabilistic element i.e., it indicates the probability associated with seeing the features c given that the instance belongs to class z.

➢ P (z) stands for class z prior probability. It is actually the probability of encountering class z without considering any additional information from the features.

➢ P (y) represents the probability of observing the features y. Also known as the marginal likelihood or evidence [50].

- Logistic Regression (LR): A discriminative model linear in nature called as Logistic Regression is used in estimating posterior probabilities of each class by the use of a SoftMax [51]. It gives strong and interpretative answers to multi-class issues.
- K-Nearest Neighbors (KNN): This non-parametric and distance-driven method like KNN which assigns labels, depending on similarity to other samples in the feature space [52]. The number of neighbors was empirically set to k=5 to strike a balance between bias and variance [53]. These models along with some basic functionality and features is shown below in fig 5.

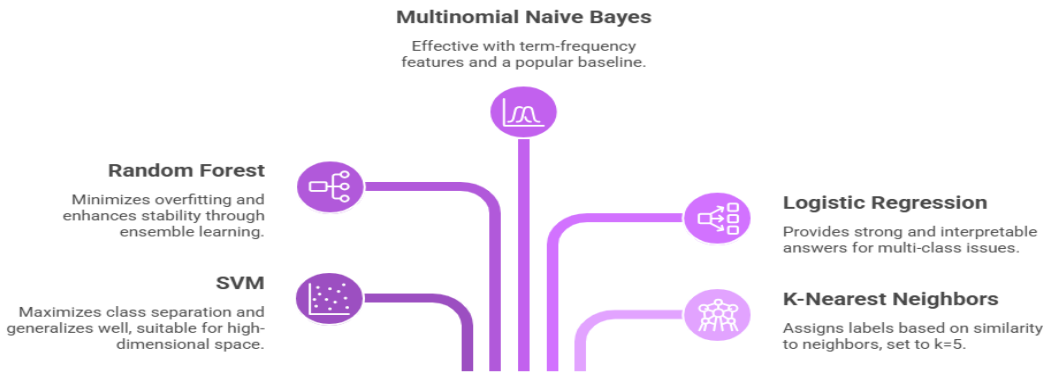

Fig 5. ML models in our experimentation

6. EVALUATION METRICS:

Many metrics used for the evaluation by the models for evaluating the performance. Evaluation matrices used in the experiment are discussed below [54]:

- Confusion Matrix: Predicted labels and actual class labels are compared in confusion matrix and also provides the insights the incorrect classifications as positives and negatives which is used in understanding the behavior of a model [55].
- Normalized Confusion Matrix: The performance of the data percentage wise is calculated by Normalized confusion matrix [56]. It is actually useful in ensemble learning where it enhances the performance of the classifier by guiding the code matrices iteratively [57].
- Precision: This metric measures the Positive Predictive Values which are actually correct. Mathematically Precision is written as [58]:

$$Precision = TP \div FP + TP$$

- Recall: Measures the instances called as True Positive which are correctly predicted. Recall is mathematically represented as [59]:

$$Recall = TP \div NP + TP$$

- F1-Score: This quantifies performance of model in combining Recall with precision through their harmonic mean. It combines balanced measure of the two [60].

$$\text{F1-Score} = 2 \times (Precision \times Recall) \div (Precison + Recall)$$

- Accuracy: Classification results are measured by their exactness. Mathematically, Accuracy is shown as [61]:

Evaluation metrices along with their basic functionality in fig 6 is depicted as below.

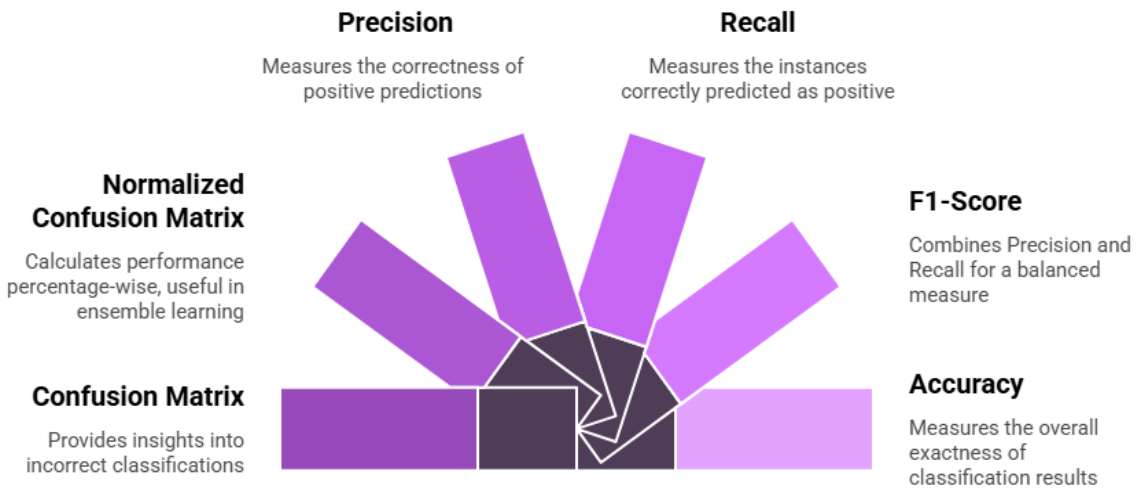

Fig 6. Evaluation Metrices

7. PROPOSED METHODS ON THE KASHMIRI DATASET

This part gives the empirical assessment of five standard supervised learning algorithms using the Kashmiri suggested text corpus. In all the models, the same pre-processing and feature extraction frames were used so that a fair comparison can be made. The textual data were first normalized by lowercasing and eliminating punctuations, stop-words and normalizing whitespaces [62]. Sentences were then changed into numerical feature vectors by using TF-IDF representation having maximum term of 5,000 terms of vocabulary [63]. Collected Corpus was splitted into 80% of training (21,600 sentences) and 20% of testing (5400 sentences) respectively. To optimize the models using the cross-validation, hyperparameter optimization was utilized to enhance the model generalization [64].

Support Vector Machine (SVM):

The corpus created is represented as.

$$D = \{(x_i, y_i)\}_{i=1}^{N} \tag{1}$$

where x_i denotes the TF-IDF feature vector corresponding to the i^{th} Kashmiri sentence and $y_i \in \{1, 2\ldots,9\}$ represents its class label. Before training the models, textual data were processed with normalization i.e., lowercasing, punctuation marks, stop-word filtering and scraping of whitespaces. The numerical representations obtained were the processed sentences, which were changed to numerical representation using weighting through a feature extraction approach that quantifies term importance using frequency within documents and inverse corpus-level distribution called as [TF-IDF] [65]. This conversion therefore transforms every document into large dimensional sparse space.

$$x_i \in \mathbb{R}^{5000}, \tag{2}$$

and the letters 5,000 represent the largest capacity of vocabulary. In order to have an objective and quantitative assessment, training and testing were carried out by dividing the dataset that sets through an 80:20 hold-out method which gave the dataset 4,966 training points and 1,242 testing points. Model was trained when applied exclusively in estimating the parameters whereas the testing of model was applied in assessing the generalization. The classifier known as Support Vector Machine was used in order to learn an optimum boundary with decision which augments by separating the classes within the margin. A one-versus-rest approach was used towards multi-class classification. Mathematically, SVM addresses the following optimization problem which is of convex nature:

$$\min_{a,z,c} \frac{1}{2} \| d \|^2 + C \sum_{i=1}^{N} e_i \tag{3}$$

subject to:

$$y_i(d \ . \ x_i + z) \geq 1 - e_i, \quad e_i \geq 0, \tag{4}$$

where weight vector is denoted by a, regularization parameter by C, bias term by z, and the slack variables of misclassification by e_i. GridSearchCV with cross-validation of five-fold was used to find the most effective hyperparameters of a certain model (like kernel and regularization strength). Separate data of testing and training were used for analyzing the effectiveness using the labelled data for training and independent set of data for testing. SVM obtained overall 93% of accuracy and indicates that there is a strong discriminative ability and high level of robustness in the multidimensional feature spaces of TF-IDF. The classification report is shown below in fig 7.

The normalized confusion matrix and the confusion matrix for SVM model are shown as fig 8 and 9.

.

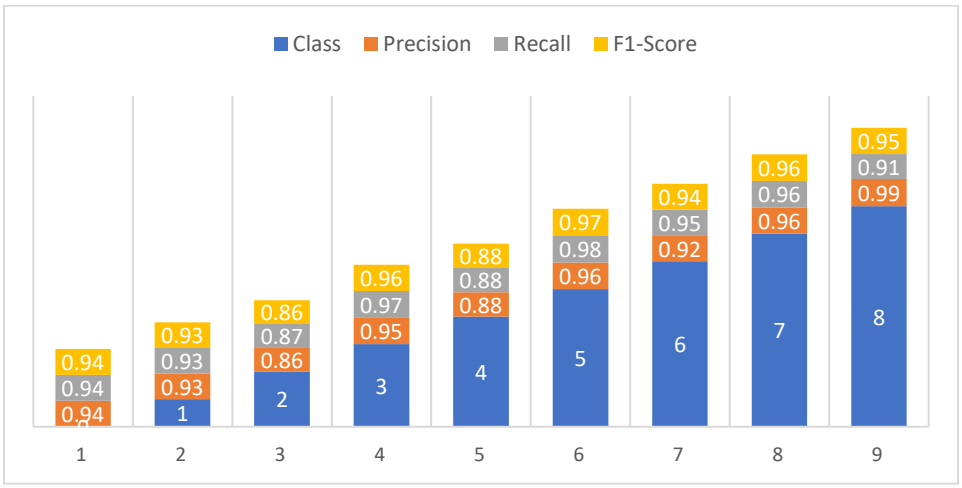

Fig 7. Classification Report of SVM

The normalized confusion matrix and the confusion matrix for SVM model are shown as fig 8 and 9.

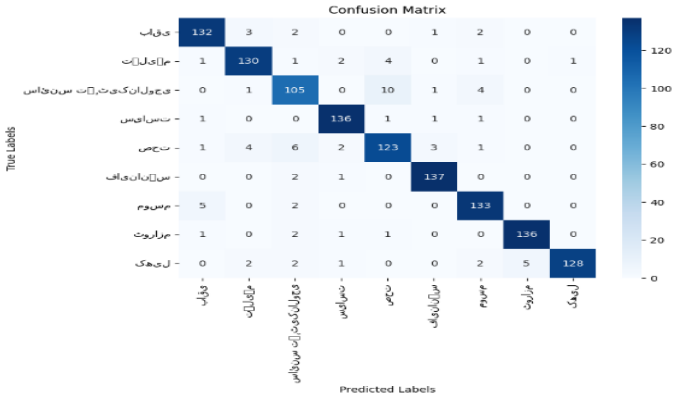

Fig 8. Confusion Matrix

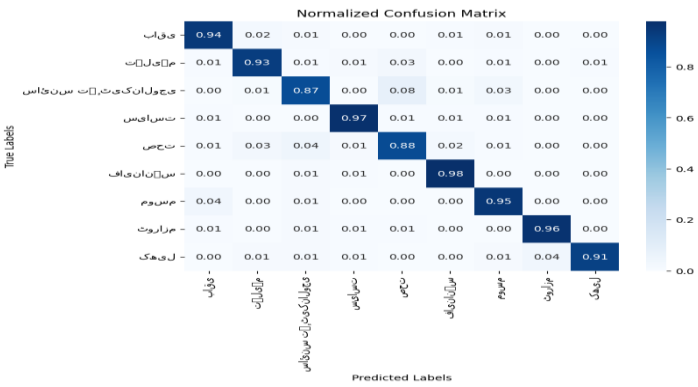

Fig 9. Normalized Confusion Matrix.

Random Forest:

Again, with the same pre-processing pipeline, every sentence in Kashmiri was converted to a numerical feature vector with the TF-IDF representation. The dataset as written mathematically is shown below:

$$Z = \{(a_i, b_i)\}_{i=1}^{M}, \quad a_i \in \mathbb{R}^{5000}, \quad b \in \{1,2,\ldots,9\}, \tag{5}$$

where ai shows sparse features of TF-IDF vector, bi shows similarity of the class of labels. In order to assess it experimentally, training of (21,600samples) and testing (5400 samples) groups based on the 80: 20 hold-out approach. The entire model parameters were only estimated on the model training and the results were evaluated on the previous unseen data to help in unbiased generalization. Random Forest was used as a method of learning of ensemble, that

involves combination of many trees of decisions to elevate predictive performance so that overfitting may be reduced. Each classifier ht(x) fitted on a bootstrapped training subset obtained through sampling is trained on a randomly selected bootstrap sample and replaced. Moreover, during the node splitting, set of randomly features is sampled so that it can bring diversity to the trees. The last prediction is voted by the majority:

$$\hat{Z} = \text{mode}\{a_1(b), a_2(b), \ldots, a_N(b)\}. \quad (6)$$

Such an aggregation approach minimizes variation and is more robust than that of individual decision tree. A default parameter model was run first. Thereafter, hyperparameters such as the maximum tree depth (d max) number of trees (n estimators), and cross-validation was done by using GridSearchCV to optimize criterion (Gini impurity or entropy). The Gini impurity of splitting nodes is the one defined as.

$$G = 1 - \sum_{k=1}^{C} p_k^2 \quad (7)$$

Where pk represents the fraction of samples to class k within a node. The trained model was made optimal using the subset of training, tested using testing of model. The illustration of the corresponding classification reports and confusion matrices is presented in Fig 10, fig 11,12 and 13. Following hyperparameter optimization, classification model called as the Random Forest attained a total of 89% accuracy which is better than the stability and generalization of the baseline setup (86%).

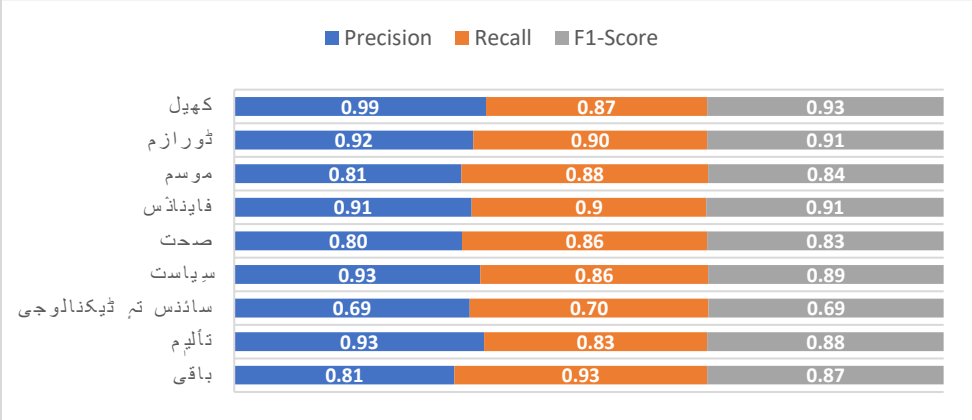

Fig 10. Classification Report on default parameters.

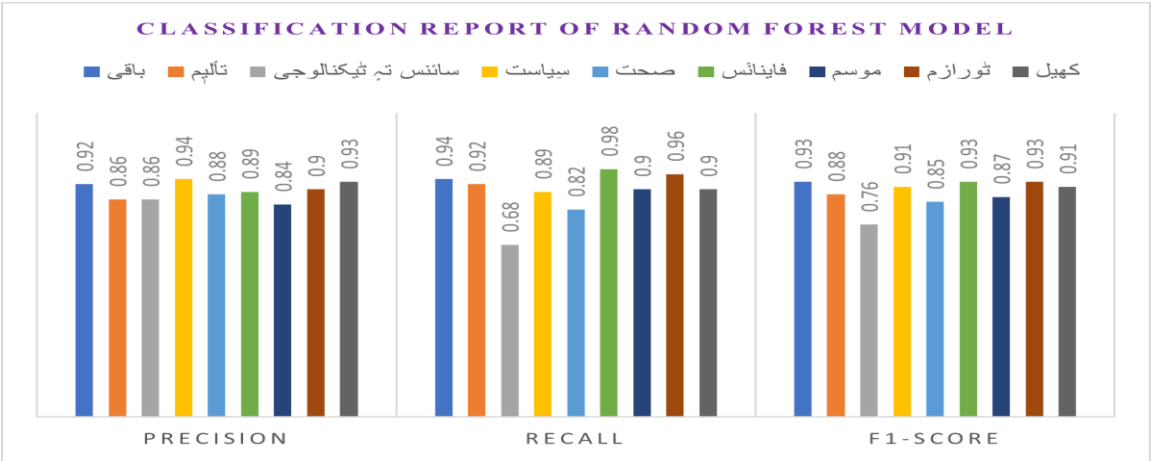

Fig 11. Classification Report with hyperparameter Tuning

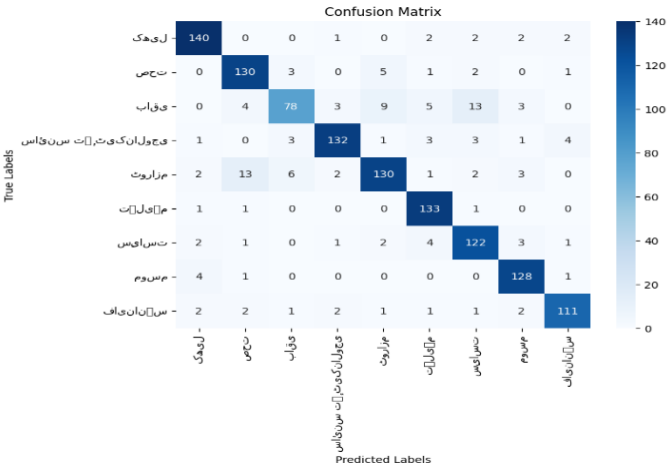

Fig 12. Confusion Matrix

Fig 13. Normalized confusion Matrix

Multinomial Naive Bayes

The Kashmiri sentences were developed in the form of pre-processed sentences that were encoded into weighted feature representation through the TF-IDF of numerical values representation. Collected corpus was split into training of 80% (21,600) and testing of 20% (5400) subsets. The estimation of model parameters was carried on the trained model, the test was performed on the unseen testing data. Multinomial Naive Bayes classifier was used because it was appropriate with text representations at the word frequency level. It is a model of probabilistic that is founded on the theorem of Bayes and features of independence which are conditional. For a document, given x = (x1, x2…, xd), the value of class c which is probability can be calculated as below:

$$P(c|x) = \frac{P(c)\prod_{j=1}^{d} P(x_j|c)}{P(x)} \tag{8}$$

The predicted label of the class under the independence assumption is derived using the maximum aposteriori (MAP) estimation as:

$$\hat{y} = \arg\max_{c} P(c)\prod_{j=1}^{d} P\left(x_j|c\right). \tag{9}$$

Laplace smoothing was used to avoid the problem of zero probability of unseen terms. Conditional estimates of probabilities were made to be:

$$P(x_i|c) = \frac{n_{jc}+\alpha}{\sum_k n_{kc}+\alpha d}, \tag{10}$$

where n_{jc} denotes feature j in class c having the frequency, d is vocabulary size, and α=0.1 is the smoothing parameter. The classifier was tested on unknown test cases after training in order to test its predictive performance and performance of classification is shown in Fig. 14. Competitive Multinomial Naive Bayes model had overall 91% of accuracy with low computational complexity, as well as quick training time.

The normalized confusion matrix and confusion and is shown below in fig 15 and 16.

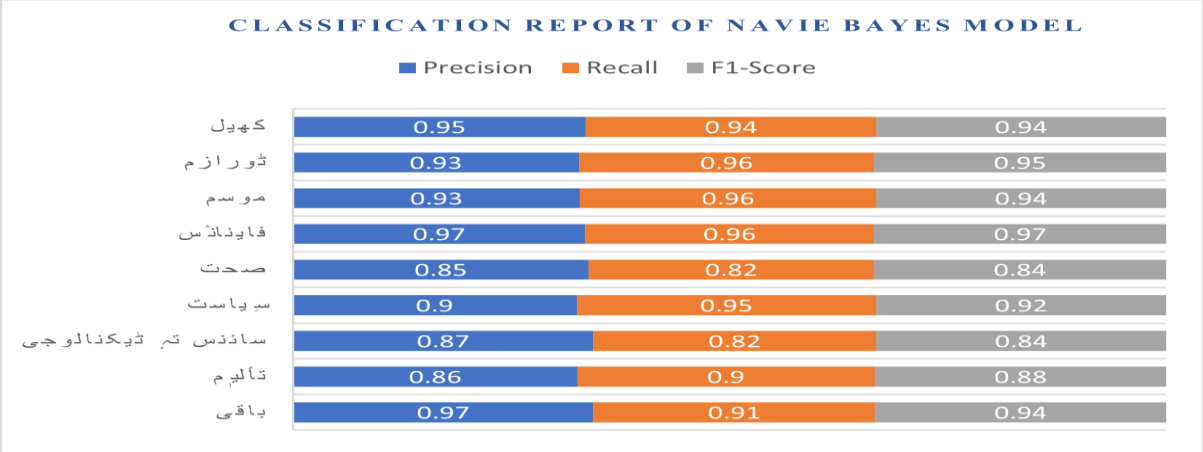

Fig 14. Classification Report of Naive Bayes Model

The normalized confusion matrix and confusion and is shown below in fig 15 and 16.

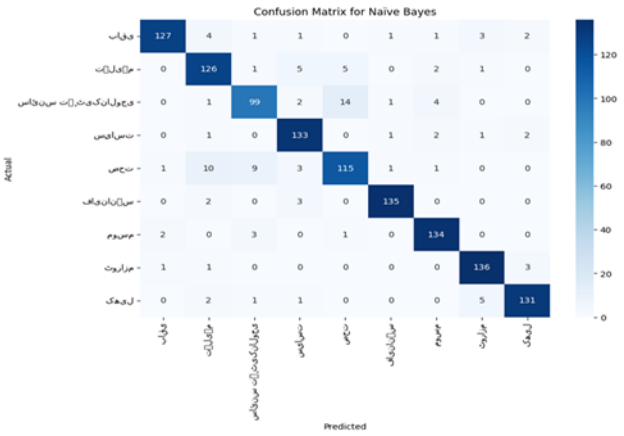

Fig 15. Confusion Matrix

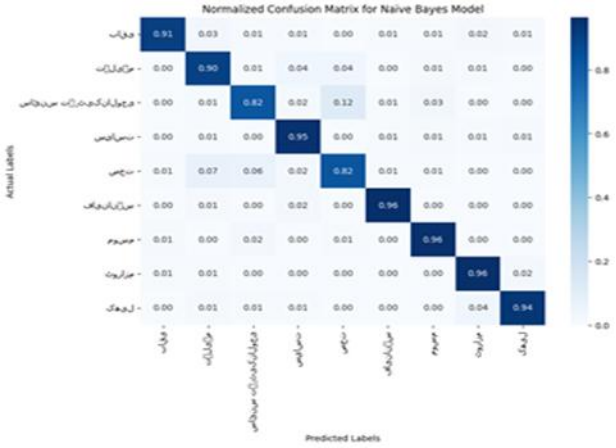

Fig 16. Normalized Confusion Matrix

Logistic Regression:

Following the feature extraction and pre-processing pipeline, a data set in Kashmiri speaking was converted to a numerical Kashmiri sentence representing the words. To be able to assess the performance in an unbiased way, collected corpus of 80% was of training and 20% of testing subsets. Logistic Regression was used as a linear discriminative classifier which is a model that directly represents posterior class probabilities in a linear manner. In the multi-class classification, multinomial (SoftMax) formulation was embraced. Given an input vector a, the probability of class k is defined as:

$$P(y = k|a) = \frac{\exp(w_b^P a + b_k)}{\sum_{n=1}^{C} \exp(w_k^P a + b_k)}, \tag{11}$$

Where b_k and w_k represents bias of class and weight of the class and C stands out as classes or labels. The postulated label is determined through the maximum a posteriori (MAP) rule:

$$\hat{y} = \arg\max_k P(y = k|x). \tag{12}$$

The parameters were estimated using function of regularized loss of cross-entropy that was minimized:

$$\mathcal{L}(\omega) = \sum_{i=1}^{N} \sum_{k=1}^{C} y_{ik} \log P(y = k|x_i) + \lambda R(\omega), \tag{13}$$

Here R_w represents the regularization loss and Lambda=1/C is a control of the strength of regularization.

Hyper parameters such as regularization coefficient (C), type of penalty (elastic-net), solver (liblinear and saga), and the number of iterations were optimized in the RandomizedSearchCV and then refined with the help of the GridSearchCV with five-fold cross-validation. The process allowed the stable convergence and avoided overfitting. The trained version of the model was tested with unseen test samples on the training subset, where the model was optimized. The associated classification reports and confusion matrices are given in Fig 17,18 and 19. The last Logistic Regression classifier had a total accuracy of 92% on an overall scale and exhibited competitive performance with SVM and at the same time was computationally efficient.

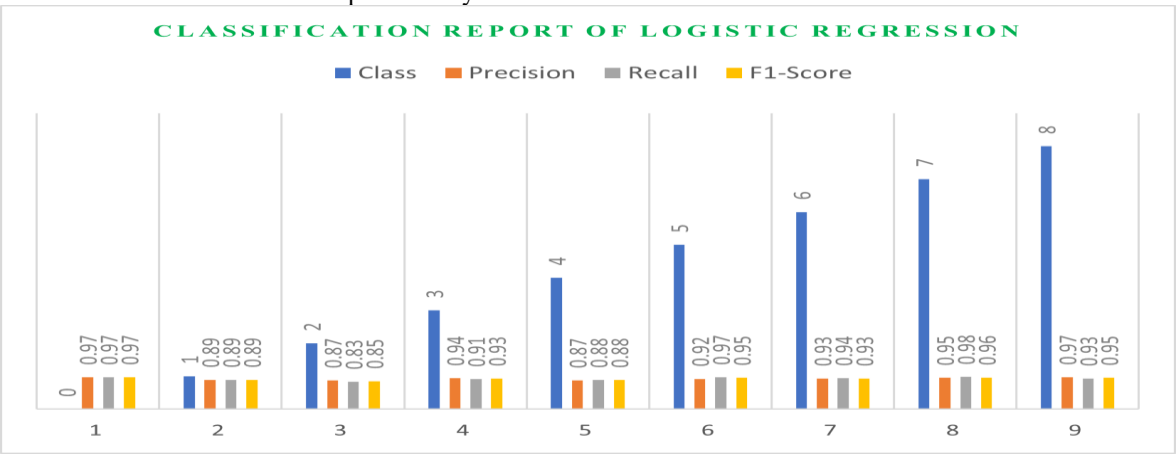

Fig 17. Classification Report of Logistic Regression

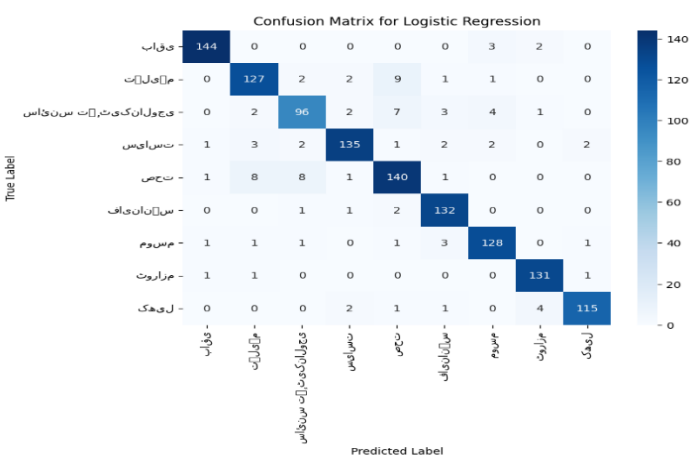

Fig 18. Confusion Matrix

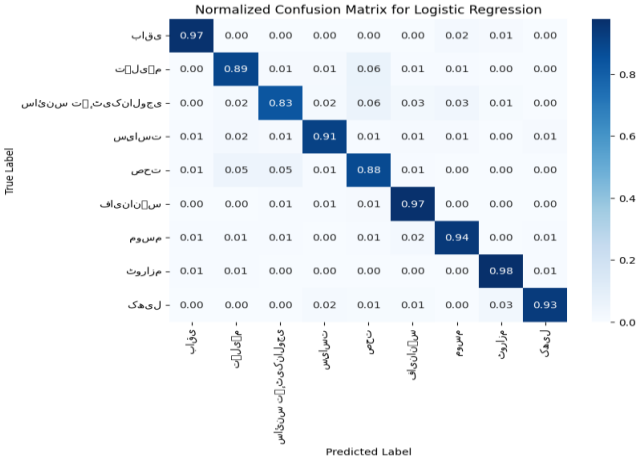

Fig 19. Normalized Confusion matrix

KNN Model:

Corpus processing of the Kashmiri language was done through the same steps of normalization and TF-IDF feature extraction list mentioned earlier. The documents were represented in a sparse feature vector each.

$$x_i \in \mathbb{R}^{5000}, \tag{14}$$

forming the labelled dataset as:

$$D = \{(x_i, y_i)\}_{i=1}^{N}, \quad y_i \in \{1,2,\ldots,9\}, \tag{15}$$

The data was segregated into training (21,600 samples) and testing (5400 samples) groups through an 80:20 hold out to provide objective test results. K-Nearest neighbor was adopted as instance-based non-parametric algorithm of learning in which a test sample is classified using labels in the features space of the nearest neighbors. For a query point x, spacing between samples was obtained as a similarity measure d (x, xj). It was the Euclidean distance that first came to use:

$$d(x, x_i) = \sqrt{\sum_{k=1}^{d} (x_k - x_{jk})^2}, \tag{16}$$

where d represents dimensions of the features.

Majority voting was used to obtain the predicted class label among the k nearest neighbors:

$$\hat{y} = \arg\max_{c} \sum_{x_j \in N_k(x)} 1\left(y_j = c\right) \tag{17}$$

Where the set of k nearest neighbors is represented by Nk(x) represents and 1(·) is the indicator function. The hyperparameter tuning was done with a GridSearchCV with a cross-validation of five-fold in order to figure out best size of neighborhood. Parameter k was optimized between odd values 3≤k≤11 within the range to reduce class tie cases. It achieved its best configuration at k = 11. The optimized model performance evaluation is shown in Fig. 20. with the classifier having an accuracy of 84%. Further tests were performed by searching other distance measures such as Manhattan and Minkowski distance to determine the effect that similarity measures had on the performance during classification. The personalized configuration achieved a slight improvement and Fig.21 below shows the results and 86% of accuracy was achieved in general.

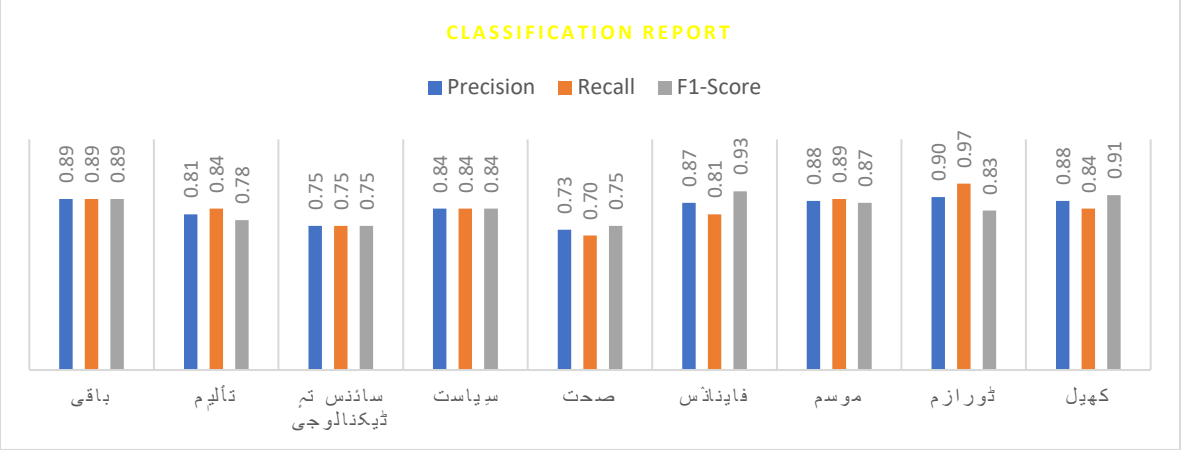

Fig 20. Classification Report of Optimized KNN model.

In comparison with linear and probabilistic models KNN showed relatively poorer performance: this is attributable to the fact that distance-based features are sensitive to high-dimensional sparsity feature spaces, and TF-IDF representations of features present high-dimensional sparsity features.

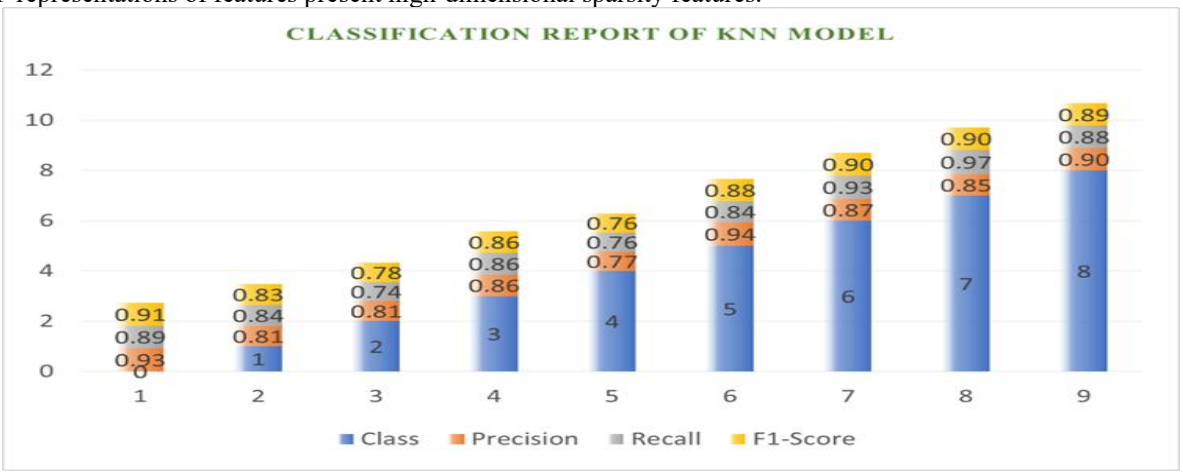

Fig 21. Classification Report by performing Hyper parameter Tuning of KNN model.

The Pictorial representation of all the models used above along with their basic mode of operation is shown below in Fig.22.

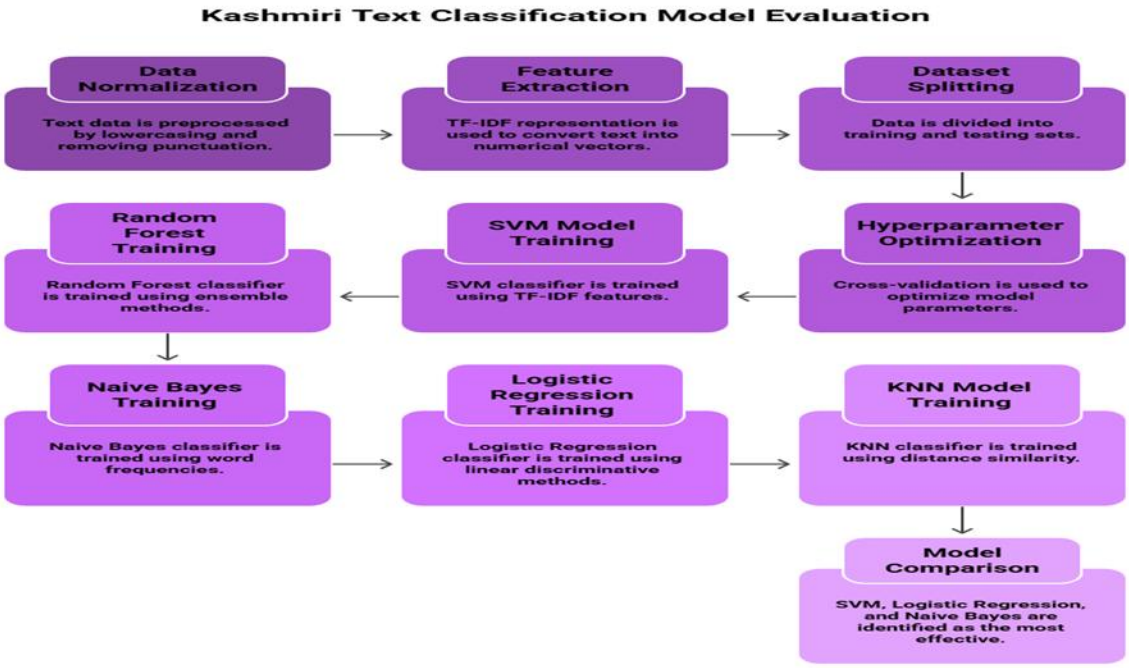

Fig 22. All models' evaluation Process.

8. EVALUATION AND RESULTS

An empirical evaluation of the deigned framework was conducted on the created corpus consisting of Kashmiri texts of 27, 000 sentences and nine classes. The corpus was systematically split into two segments in which 80% was utilized for training and 20% was retained for testing its performance. The training of all models was done based on the same preprocessing and TF-IDF feature representation to provide a fair comparison. To quantitatively evaluate the effectiveness of the model, the performance measures like recall, precision, F1-score and accuracy were employed. Findings indicate that the classical machine learning models are effective in text classification of Kashmiri. Of the considered models, Support Vector Machine (SVM) demonstrated the best overall performance, with the next place being occupied by the Logistic Regression and Multinomial Naive Bayes. Random Forest was performing averagely well and K-Nearest Neighbors (KNN) was performing relatively low because it is sensitive to high-dimensional sparse features space. The overall summary of the performance metrics of all the models evaluated is detailed in Table 2 as below:

Table 2: Outcomes of the Evaluated Models:

Technique Used	Accuracy (%)	Precision	Recall	F1 Score
Support Vector Machine	93	0.94	0.94	0.94
Logistic Regression	92	0.93	0.93	0.93
Naive Bayes	91	0.91	0.91	0.91
Random Forest	89	0.88	0.88	0.88
KNN	85	0.84	0.85	0.84

9. DISCUSSIONS

This paper critically assessed the classical machine learning algorithms to classify Kashmiri text based on a manually annotated and curated corpus. The experimental findings indicate that there are apparent differences in evaluation results of compared classifiers. Most accurate of all methods was the Support Vector Machine with a close second place going to Probabilistic model, distance-based, ensemble-based and linear models in their turn, demonstrated relatively low performance. The higher performance of SVM and Logistic Regression could be explained by the capability to effectively operate on thin feature representation with high-dimensional created by TF-IDF. These models construct linear decision boundaries that allow text classification problems to be better classified,

which involves a predominant role of discriminative lexical features in the model. Similarly, competitive performance of Naive Bayes shows how well the probabilistic model can be applied to word-frequency-based representations. In contrast, the distance-based analyses like KNN had lower accuracy because of their tendency to be susceptible to the curse of dimensionality, in which the distance measures become less informative when the space sparseness. Random Forest tree-based ensemble methods were also performing moderately, probably because splits are not optimal when the textual features are very sparse. The misclassifications as seen in some of the categories could be ascribed to the limited scale of training samples, lexical similarity between classes and the language complexity in Kashmiri. These aspects indicate the difficulties related to processing low-resource languages and the necessity to use larger and more diverse annotated corpora. All in all, experiment evaluation showed that classical learning methods are trustworthy, computationally efficient, and provide interpretation possibilities of Kashmiri text tasks of classification.

10. CONCLUSION AND FUTURE RESEARCH

Research conducted in this study was a detailed analysis on the Kashmiri text categorization based on various supervised machine learning methods. A tagged sentence-level database was created and used to test five popular classifiers such as ensemble approach, distance-based model, probabilistic, linear probabilistic model and margin-based classifier. Result evaluation analysis revealed that SVM had overall accuracy of 93% which is good performance than Logistic Regression and Naive Bayes, which implies the strength of linear and probabilistic models in textual representations of TF-IDF. These findings are very good baseline standards that can be used in future studies in Kashmiri Natural Language Processing. However, in spite of encouraging results, there are a number of restrictions. Dataset being limited in size, and additional gains in performance can be made with the use of corpus expansion, better class balancing, and better pre-processing plans. Discrimination between classes might be also enhanced further by incorporating more linguistic features like morphological normalization, stemming and syntactic cues.

Future functionality will involve expanding the data-set, developing new sophisticated feature engineering methods, and finding more compelling classification systems. Corpus and baseline models developed offer the learning basis of future studies on Kashmiri NLP, such as sentiment analysis, information retrieval, understanding of the language.

ACKNOWLEDGEMENT

The author (first author) acknowledges the sincere gratitude to Dr. Nawaz Ali Lone, faculty member in the department of computer science, IUST, for his encouragement and help from time to time during the writeup of research papers. Special thanks and deep acknowledgement to Dr. Nazima Nazir whose support and motivation is unweaving. I also appreciate the suggestions received from my junior colleagues especially Sheezan Farooq and Ishrat Nabi during the completion of this paper.

Services and Facilities involved

The author acknowledges the use of HPC lab facility provided by the IUST, Kashmir while performing the experimentation of the dataset.

Data availability: NA

Reference

1. Joshi, A. Dabre, R., Kanojia, D., Li, Z., Zhan, H., Haffari, G., & Dippold, D (2025). Natural language processing for dialects of a language: A survey. ACM Computing Surveys, 57(6), 1-37.
2. Fields, J., Chovanec, K., & Madiraju, P. (2024). A survey of text classification with transformers: How
3. wide? how large? how long? how accurate? how expensive? how safe? IEEE Access, 12,
4. 6518-6531.
5. Taha, K., Yoo, P. D., Yeun, C., Homouz, D., & Taha, A. (2024). A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights. Computer Science Review, 54, 100664.
6. Chai, D., Wu, W., Han, Q., Wu, F., & Li, J. (2020, November). Description based text classification with reinforcement learning. In International conference on machine learning (pp. 1371-1382). PMLR.
7. Tejaswini, V., Sathya Babu, K., & Sahoo, B. (2024). Depression detection from social media text analysis using natural language processing techniques and hybrid deep learning model. ACM Transactions on Asian and Low-Resource Language Information Processing, 23(1), 1-20.
8. Khan, T., Sarkar, R., & Mollah, A. F. (2021). Deep learning approaches to scene text detection: a comprehensive review. Artificial Intelligence Review, 54, 3239-3298.

9. Luo, X. (2021). Efficient English text classification using selected machine learning techniques. Alexandria Engineering Journal, 60(3), 3401-3409.
10. Feuerriegel, S., Maarouf, A., Bär, D., Geissler, D., Schweisthal, J., Pröllochs, N., ... & Van Bavel, J. J. (2025). Using natural language processing to analyse text data in behavioural science. Nature Reviews Psychology, 4(2), 96-111.
11. Bolaj, P., & Govilkar, S. (2019). A survey on text categorization techniques for Indian regional languages. IJCSIT, 7(2), 480-483.
12. Chen, X., Cong, P., & Lv, S. (2022). A long-text classification method of Chinese news based on BERT and CNN. IEEE Access, 10, 34046-34057.
13. Saigal, P., & Khanna, V. (2020). Multi-category news classification using support vector machine-based classifiers. SN Applied Sciences, 2(3), 458.
14. Cozman, F. G., Cohen, I., & Cirelo, M. C. (2023, August). Semi-supervised learning of mixture models. In ICML (Vol. 4, p. 24).
15. Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. Information Processing & Management, 57(1), 102121.
16. Abdeen, M. A., AlBouq, S., Elmahalawy, A., & Shehata, S. (2019). A closer look at arabic text classification. International Journal of Advanced Computer Science and Applications, 10(11).
17. Sayed, M., Salem, R. K., & Khder, A. E. (2019). A survey of Arabic text classification approaches. International Journal of Computer Applications in Technology, 59(3), 236-251.
18. Scheible, R., Thomczyk, F., Tippmann, P., Jaravine, V., & Boeker, M. (2020). Gottbert: a pure German language model. arXiv preprint arXiv:2012.02110.
19. Cordobés H, Fernández A, Chiroque LF, Pérez F, Redondo T, Santos A (2014) Graph-based techniques for topic classification of tweets in Spanish. Int J Artif Intell Interac Multimed 02:31–37
20. Farhoodi, M., & Yari, A. (2010, November). Applying machine learning algorithms for automatic Persian text classification. In 2010 6th International Conference on Advanced Information Management and Service (IMS) (pp. 318-323). IEEE.
21. Akhter, M. P., Jiangbin, Z., Naqvi, I. R., Abdelmajeed, M., & Fayyaz, M. (2022). Exploring deep learning approaches for Urdu text classification in product manufacturing. Enterprise Information Systems, 16(2), 223-248.
22. Kandhro, I. A., Jumani, S. Z., Kumar, K., Hafeez, A., & Ali, F. (2020). Roman Urdu headline news text classification using RNN, LSTM and CNN. Advances in Data Science and Adaptive Analysis, 12(02), 2050008.
23. Zin, H. M., Mustapha, N., Murad, M. A. A., & Sharef, N. M. (2022). A Meta-heuristic Algorithm for the Minimal High-Quality Feature Extraction of Online Reviews. Journal of Information and Communication Technology, 21(4), 571-593.
24. Khan, K., Khan, R. U., Alkhalifah, A., & Ahmad, N. (2015, December). Urdu text classification using decision trees. In 2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET) (pp. 1-4). IEEE.
25. Nidhi, Vishal Gupta, "Punjabi Text Classification using Naïve Bayes, Centroid and Hybrid Approach", DOI: 10.5121/csit.2012.2421.
26. Kaur, J., & Saini, J. R. (2020). Designing Punjabi poetry classifiers using machine learning and different textual features. Int. Arab J. Inf. Technol., 17(1), 38-44.
27. Swamy, M. N., Hanumanthappa, M., & Jyothi, N. M. (2014, March). Indian language text representation and categorization using supervised learning algorithm. In 2014 International Conference on Intelligent Computing Applications (pp. 406-410). IEEE.
28. ArunaDevi, K., & Saveetha, R. (2014). A novel approach on Tamil text classification using C-feature. Int. J. Sci. Res. Dev, 2, 343-345.
29. Rajan, K., Ramalingam, V., Ganesan, M., Palanivel, S., & Palaniappan, B. (2019). Automatic classification of Tamil documents using vector space model and artificial neural network. Expert Systems with Applications, 36(8), 10914-10918.
30. Jayashree, R., & Srikanta, M. K. (2011, October). An analysis of sentence level text classification for the Kannada language. In 2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR) (pp. 147-151). IEEE.
31. Patil, J. J., & Bogiri, N. (2015, October). Automatic text categorization: Marathi documents. In 2015 International Conference on Energy Systems and Applications (pp. 689-694). IEEE.
32. Patil, M., & Game, P. (2014). Comparison of Marathi text classifiers. International Journal on Information Technology, 4(1), 11.
33. Talukdar, C., & Sarma, S. K. (2023). Hybrid Model for Efficient Assamese Text Classification using CNN-LSTM. International Journal of Computing and Digital Systems, 14(1), 10183-10192.
34. Talukdar, C., & Sarma, S. K. (2023, December). Assamese document classification using CNN, multi-channel CNN and CNN-SVM. In AIP Conference Proceedings (Vol. 2901, No. 1). AIP Publishing.

35. Rakholia, R. M., & Saini, J. R. (2017). Classification of Gujarati documents using Naïve Bayes classifier. Indian Journal of Science and Technology, 5, 1-9.
36. Shahi, T. B., & Pant, A. K. (2018, February). Nepali news classification using Naive Bayes, support vector machines and neural networks. In 2018 international conference on communication information and computing technology (iccict) (pp. 1-5). IEEE.
37. Kabir, F., Siddique, S., Kotwal, M. R. A., & Huda, M. N. (2015, March). Bangla text document categorization using stochastic gradient descent (sgd) classifier. In 2015 International Conference on Cognitive Computing and Information Processing (CCIP) (pp. 1-4). IEEE.
38. Mandal, A. K., & Sen, R. (2014). Supervised learning methods for bangla web document categorization. arXiv preprint arXiv:1410.2045.
39. Joshi, R., Goel, P., & Joshi, R. (2020). Deep learning for hindi text classification: A comparison. In Intelligent Human Computer Interaction: 11th International Conference, IHCI 2019, Allahabad, India, December 12–14, 2019, Proceedings 11 (pp. 94-101). Springer International Publishing.
40. Tummalapalli, M., Chinnakotla, M., & Mamidi, R. (2018, March). Towards better sentence classification for morphologically rich languages. In Proceedings of the international conference on computational linguistics and intelligent text processing.
41. Dixit, N., & Choudhary, N. (2014). Automatic classification of Hindi verbs in syntactic perspective. International Journal of Emerging Technology and Advanced Engineering, 4, 2250-2459.
42. Swapna, N., Subhashini, P., & Rani, B. P. (2019). Impact of stemming on telugu text classification. International Journal of Recent Technology and Engineering (IJRTE), 8(2), 2767-2769.
43. Gampala, V., Vallapuneni, J., Ande, P. K., Indurthi, R. K., & Rajesh, N. (2021, June). Comparative study on telugu text classification using machine learning and deep learning models. In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1393-1398). IEEE.
44. Saigal, P., & Khanna, V. (2020). Multi-category news classification using Support Vector Machine based classifiers. SN Applied Sciences, 2(3), 458.
45. Moreo, A., Esuli, A., & Sebastiani, F. (2021). Word-class embeddings for multiclass text classification. Data Mining and Knowledge Discovery, 35, 911-963.
46. Islam, M. Z., Liu, J., Li, J., Liu, L., & Kang, W. (2019, November). A semantics aware random forest for text classification. In Proceedings of the 28th ACM international conference on information and knowledge management (pp. 1061-1070).
47. Sun, Y., Li, Y., Zeng, Q., & Bian, Y. (2020, April). Application research of text classification based on random forest algorithm. In 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE) (pp. 370-374). IEEE.
48. Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. Knowledge-Based Systems, 192, 105361.
49. Su, X. A., Wang, R., & Dai, X. (2022, May). Contrastive learning-enhanced nearest neighbor mechanism for multi-label text classification. In Proceedings of the 60th annual meeting of the association for computational linguistics (volume 2: short papers) (pp. 672-679).
50. Ali, D., Missen, M. M. S., & Husnain, M. (2021). Multiclass event classification from text. Scientific Programming, 2021(1), 6660651.
51. Krichen, M. (2023). Convolutional neural networks: A survey. Computers, 12(8), 151.
52. Hasib, K. M., Azam, S., Karim, A., Al Marouf, A., Shamrat, F. J. M., Montaha, S., ... & Rokne, J. G. (2023). Mcnn-lstm: Combining cnn and lstm to classify multi-class text in imbalanced news data. IEEE Access, 11, 93048-93063.
53. Adamuthe, A. C. (2020). Improved text classification using long short-term memory and word embedding technique. Int J Hybrid Inf Technol, 13(1), 19-32.
54. Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-label confusion matrix. Ieee Access, 10, 19083-19095.
55. Guo, B., Han, S., Han, X., Huang, H., & Lu, T. (2021, May). Label confusion learning to enhance text classification models. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 14, pp. 12929-12936).
56. Hassan, S. U., Ahamed, J., & Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification. Sustainable Operations and Computers, 3, 238-248.
57. Yacouby, R., & Axman, D. (2020, November). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the first workshop on evaluation and comparison of NLP systems (pp. 79-91).
58. Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augmented Human Research, 5(1), 12.

59. Lu, H., Ehwerhemuepha, L., & Rakovski, C. (2022). A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. BMC medical research methodology, 22(1), 181.
60. Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. Scientific Reports, 14(1), 6086.
61. Erbani, J., Portier, P. É., Egyed-Zsigmond, E., & Nurbakova, D. (2024). Confusion matrices: A unified theory. Ieee Access, 12, 181372-181419.
62. Rahmad Ramadhan, L., & Anne Mudya, Y. (2024). A comparative study of Z-score and min-max normalization for rainfall classification in Pekanbaru. Journal of Data Science, 2024(04), 1-8.
63. Furube, T., Takeuchi, M., Kawakubo, H., Maeda, Y., Matsuda, S., Fukuda, K., ... & Kitagawa, Y. (2024). Automated artificial intelligence–based phase-recognition system for esophageal endoscopic submucosal dissection (with video). Gastrointestinal Endoscopy, 99(5), 830-838.
64. Hricak, H., Mayerhoefer, M. E., Herrmann, K., Lewis, J. S., Pomper, M. G., Hess, C. P., ... & Weissleder, R. (2025). Advances and challenges in precision imaging. The Lancet Oncology, 26(1), e34-e45.
65. Yang, J., Yang, S., Gupta, A. W., Han, R., Fei-Fei, L., & Xie, S. (2025). Thinking in space: How multimodal large language models see, remember, and recall spaces. In Proceedings of the Computer Vision and Pattern Recognition Conference (pp. 10632-10643).
66. Mustapha, M., Zineddine, M., Kaufman, E., Friedman, L., Gmira, M., Majikumna, K. U., & Alaoui, A. E. H. (2025). A hybrid machine learning approach for imbalanced irrigation water quality classification. Desalination and Water Treatment, 321, 100910.
67. Saricaoglu, A., & Bilki, Z. (2025). The capacity of ChatGPT-4 for L2 writing assessment: A closer look at accuracy, specificity, and relevance. Annual Review of Applied Linguistics, 1-21.