



# Emotion Classification with Reduced Feature Set SGDClassifier, Random Forest and Performance Tuning

Kaushika Pal<sup>1</sup>(✉) and Biraj V. Patel<sup>2</sup>

<sup>1</sup> Sarvajanic College of Engineering & Technology, Surat, Gujarat, India  
Kaushika.pal@scet.ac.in

<sup>2</sup> G. H. Patel P. G. Department of Computer and Technology Science, Sardar Patel University,  
V. V. Nagar, Gujarat, India

**Abstract.** Text Classification is vital and challenging due to varied kinds of data generated these days; emotions classification represented in form of text is more challenging due to diverse kind of emotional content and such content is growing on web these days. This research work is classifying emotions written in Hindi in form of poem with 4 categories namely Karuna, Shanta, Shringar and Veera. POS tagging is used on all the poem and then features are extracted by observing certain poetic features, two types of features are extracted and the results in terms of accuracy is measured to test the model. 180 Poetries were tagged and features were extracted with 8 different keywords, and 7 different keywords. The model is build with Random Forest, SGDClassifier and was trained with 134 poetries and tested with 46 Poetries for both types of features. The results with 7 keyword feature is comparatively better than 8 keyword feature by 7.27% for Random Forest and 10% better for SGDClassifier. Various combinations of hyper parameters are used to get the best results for statistical measure precision and recall for performance tuning of the model. The model is also tested with k – fold cross validation with average result 62.53% for 4 folds and 60.45% for 8 folds with Random Forest and 54.42% for 4 folds and 48.28% for 8 folds with SGDClassifier, the experimentation result of Random Forest is better than SGDClassifier on the given dataset.

**Keywords:** Emotions · Poetry · Feature extraction · Machine learning · POS tagging · SGDClassifier

## 1 Introduction

Feelings, emotions, sentiments are beautiful substance which human being cannot get rid of, we feel we express, the emotions either by crying, laughing, singing, dancing, jumping or by writing. The cyberspace has given every individual an opportunity to freely express by various means, videos on YouTube with poetry and story telling episodes, or using any short videos featuring applications. Some sing, some speak and some write, applications like YouQuote, allows individual to express by writing statements, quotes or poetries, there are many such applications. When it comes to express, anyone

prefer in the language they are comfortable, India with 22 major languages expressed using 13 different scripts with approximately 720 dialects give options to everyone to express in the language they know, they understand and can write. This huge literature in multiple languages in India, gives the researcher a challenge to provide computerized and automated solutions for almost all problems.

Hindi known to be an official language of India, with nearly 420 million speakers needs special attention from researchers of this country. There are many researchers who are consistently trying to contribute for enriching the web with all options to get the things we need. Poetries are written emotions, which are expressed and measured in Navrasas in Hindi, Ras means sentient, which also implies to sensation, sensation of feelings. This research work is classifying those sensations in 4 categories Karuna, Shanta, Shringar and Veera. The details of the data set used along with meaning and associated emotions are shown in Table 1.

**Table 1.** Data set class and it’s meaning and associated emotions

Class	Karuna	Shanta	Shringar	Veera
Meaning	Pity, sadness	Peace	Romance, love	Heroic, courage
Associated emotions	Compassion, sympathy	Calmness, relaxation	Devotion, beauty	Confidence, pride

## 2 Study of Related Work and Motivation

The feasibility to implement current work needed exploration of work done in Indic Language and Hindi Language, The work done in Hindi is focused to get insight, and image processing which are representing characters by some researchers is also studied. M. Shalini [1] used neural network to recognize Hindi words from image, the researcher used line segmentation, word segmentation techniques to extract word from the image. Shalini Puria [2] introduced tri-layered segmentation and bi-leveled-classifier-based classification system for Hindi printed documents using Support Vector Machine and Fuzzy. Jasleen [3] classified Punjabi poetry using linguistic features and weighing, she found 72.04% accuracy with TF weighing and 66.43% with TF-IDF weighing using Support Vector Machine with dataset of 2034 Poetries. Mandal AK [4] explored machine learning and used Decision Tree (C4.5), Naïve Bayes, K-Nearest Neighbor, and Support Vector Machine for Bangla corpus, the author performed classification of corpus into business, sports, health, technology, and education classes. Vandana Jha [5] proposed a method for opinion about Hindi movie review and used lexicon based classification techniques using Naïve Bayes, Support Vector Machine. Jasleen [6] analyzed performance comparison of different Techniques used in Formal and Informal Text Classification for sentiment classification. Noraini Jamal [7] classified Malay poetry into different genre using Support Vector Machine with ‘rbf’ and ‘linear’ kernel and found maximum accuracy of 58.44% with dataset of 1500 Poetries. The author also experimented to identify poetry and non-poetry contents and the accuracy found was 99.9%, the author claims Support Vector

Machine with 'linear' kernel is giving better results than with 'rbf' kernel. Hamid R [8] proposes novel poetic features and classified poem from normal text with 5 different approaches namely Text Classification, Shape features, combining Rhyme and shape, combining Rhyme, meter and Shape, combining rhyme and shape with word frequency. He concludes that using all approaches very efficient classifier is build to classify Normal Text from Poetry. Shalini Puria [11] proposed a model for devanagari character classification using Support Vector Machine for printed and handwritten image based characters and claiming to have accuracy of 99.54% for printed characters and 98.35% for handwritten characters by using dataset of 60 Documents, there accuracy is high as they are only categorizing into characters. K Pal [21] surveyed on research done in Indic Languages and found that the research needs more attention from feature extraction, feature selection, Classification, Text Summarization aspects using Artificial Intelligence. Ishaan [12] used Naïve Bayes to build spam filter for Hindi language. C Anne [14] developed multiclass document Classification using ML and NLP techniques. Experiments by Noraini Jamal [7], evidently shows classifying poetries into poetic genre is very challenging and achieved accuracy of 58.44% using Support Vector Machine with 1500 Poetries. Yu Meng [15] proposed weakly Supervised Neural Text Classification, which addresses the lack of training data for text classification using Neural Networks by using pseudo document generator for generating pseudo training data. Qiancheng Liang [16] has combined word meaning and semantic features for text classification using neural networks and machine learning. Tu Cam Thi Tran [17] proposed a model, which uses keywords with different thresholds for Text Classification. Md Zahidul Islam [18] claims random Forest is good to deal with noisy data in Text Classification and proposes semantic aware random forest for text classification. Wanwan Zheng [19] claims that feature selection helps to have 66.67% less training samples. Rui Yao [20] proposed a model, which identifies false promotions by webpages using sensitive word filtering method. Cannannore Nidhi Kamath [9] compared performance of many machine learning algorithms and CNN for text classification and found that Logistic regression is performing better than other machine learning algorithms but CNN is performing better than all. Mariem Bounabi [10] have raised issues in TF-IDF for text classification and proposes extended form of it called as FTF-IDF which uses fuzzy to increase the performance of classification. Anna Surkova [13] uses cognitive approach and linguistic approach for text classification and claims that linguistic approach does not improve classification.

Studying all the work carried out by diverse researchers motivated to experiment the capabilities of machine learning techniques for emotion classification represented in Hindi, which is yet to be explored.

Human beings can understand emotions but training machines to understand "emotions" is challenging due to words order, rhythm, Shape, different way of expressing emotions by different writer; so much information is fused in short sentences; writing style of each poet is very different from another poet of same genre poetry; special characters used to end or express certain emotions is also used by some writers but same special characters are used by different writers in other way.

### 3 System Architecture

The System comprises of **POS tagging Module**, **Feature Extraction**, **Training the Classifier**, and **Testing of the Classifier** with new unseen test data. The System Architecture for the classifier using Part-of-the-speech tagging for feature extraction is shown in Fig. 1.

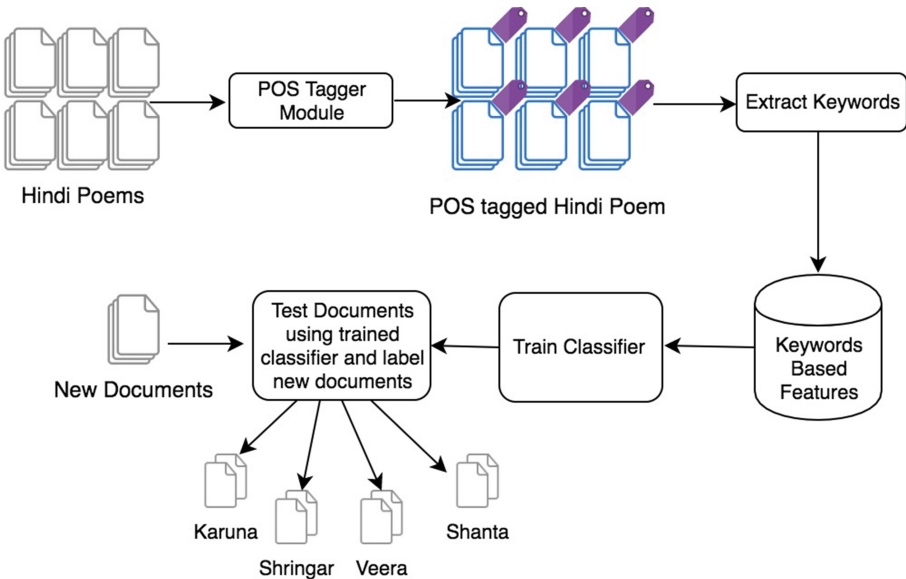
The System is implemented in Python 3.6 using PyCharm Community Edition on macOS High Sierra version 10.13.1 with 1.8 GHz Intel Core i5 Processor.

The Data Set comprises of 180 Poems of 4 Categories namely Shringar, Karuna, Veera and Shanta Ras and represents emotions of Love, Pity, Heroic and Peace respectively.

Bulk POS tagging Module is developed which perform part of the speech tagging and tagged 48 Shringar, 49 Karuna, 43 Veera and 40 Shanta Ras Poems. This module generates tagged poem files, which are stored and used for Feature Extraction.

POS tags which are used for tagging are ‘PRP’ Pronoun, ‘NNP’ Proper noun, ‘NN’ Noun, ‘JJ’ Adjective, ‘VAUX’ auxiliary Verb, ‘RP’ Particle, ‘RB’ Adverb, ‘CC’ Conjunction, ‘QF’ Quantifiers, ‘PREP’ Postposition, ‘VFM’ Verb Finite main, ‘INTF’ Intensifier, ‘NLOC’ Noun Location, ‘NNC’ Compound, ‘NEG’ Negative, noun ‘QFNUM’ Quantifiers number, ‘QW’ Question words, ‘PUNC’ punctuation, ‘NNPC’ Compound proper nouns, ‘VNN’ Verb non-finite nominal, ‘NVB’ Noun in Kriyamula, ‘VJJ’ Verb non-finite adjectival and ‘Unk’ Unknown. Figure 2 shows a Sample of one-tagged poetry.

Feature Extraction is crucial for efficient classification; predicting feature set for classification without experimenting on given data set is not possible, starting experiment using large Feature set and carefully observing the results the features can be reduced



**Fig. 1.** System architecture for the classifier using part-of-the-speech tagging for feature extraction



Fig. 2. Sample file showing tagging of each word of the poetry

using feature selection. For this research work the POS tagged poems were used to extract features by monitoring the tagged poem document. There were two ways the features were extracted using this experiment. Since the poetry express emotions, the words tagged with 'Unk' was ignored for one experiment but was considered for the second experiment. The words, which were given more importance for this classification, were Adverbs, Adjectives as they have higher chance to represent emotions. The feature Set that used 'Unk' meaning unknown words was challenging as it was having certain important features but were also loaded with lot of garbage values including printed and non-printed characters, this characters were removed by observing keywords extracted with 'Unk' tag and writing script in python to remove unwanted characters from the Feature Set.

Table 2. Statistics of POS tagging and Keywords Extracted

Poem class	No. of documents	No. of words tagged	No. of keywords extracted	Duration of process (HH:MM:SS)	No. of keywords ignoring 'Unk' tag	Duration of process (HH:MM:SS)
Karuna	49	7400	5246	00:01:33	1405	00:01:25
Shanta	40	5144	3818	00:00:51	926	00:00:47
Shringar	48	6875	5066	00:01:11	1002	00:01:05
Veera	43	10537	7966	00:03:10	2022	00:02:59
Total	180	29,956	22,096	00:06:45	5,352	00:06:16

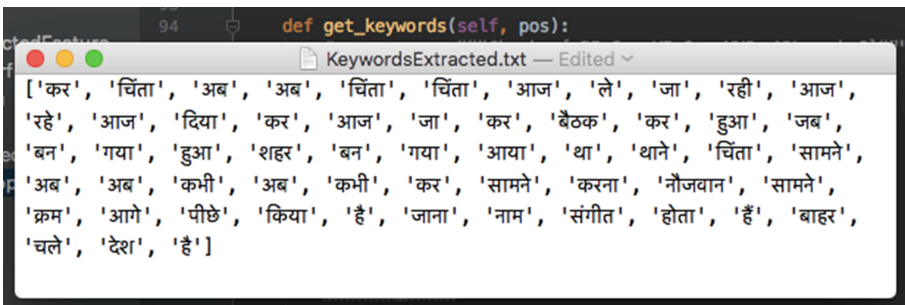
The Statistics of number of words tagged and feature extracted ignoring 'Unk' and including it are shown in Table 2.

The Sample file for Keywords extracted is shown in Fig. 3., and keywords extracted ignoring 'Unk' is shown in Fig. 4.



```
def get_keywords(self, pos):
KeywordsExtractedUnk.txt
['खेल', 'रचो', 'साथी', 'जीने', 'आनंद', 'मिले', 'मरने', 'आनंद', 'मिले', 'दुनिया', 'सूते',
'आँगन', 'खेल', 'रचो', 'साथी', 'मरघट', 'सन्नाटा', 'रह', 'रह', 'काटे', 'जाता', 'है', 'दुःख',
'दर्द', 'तबाही', 'दबकर', 'मुफ्तलिस', 'दिल', 'चिल्लाता', 'है', 'झूठा', 'सन्नाटा', 'टूटे', 'पापों',
'भरा', 'घड़ा', 'फूटे', 'तुम', 'जंजीरों', 'झनझन', 'खेल', 'रचो', 'साथी', 'उपदेशों', 'संचित',
'रस', 'फीका', 'फीका', 'लगता', 'है', 'सुन', 'धर्म', 'कर्म', 'बातें', 'दिल', 'अंगार', 'सुलगता',
'है', 'चाहे', 'दुनिया', 'जल', 'जाए', 'मानव', 'रूप', 'बदल', 'जाए', 'तुम', 'आज', 'जवानी',
'क्षण', 'खेल', 'रचो', 'साथी', 'दुनिया', 'सिर्फ', 'सफलता', 'उत्साहित', 'क्रीड़ा', 'कलरव', 'है',
'जीवन', 'जीतों', 'मोहक', 'मतवाला', 'उत्सव', 'है', 'तुम', 'चेतो', 'साथी', 'तुम', 'जीतो',
'साथी', 'संघर्षों', 'निष्ठुर', 'रण', 'खेल', 'रचो', 'साथी', 'जीवन', 'चंचल', 'धारा', 'धर्म', 'बहे',
'बह', 'मरघट', 'राखों', 'लिपटी', 'लाश', 'रहे', 'रह', 'जाने', 'आँधी', 'अंधड़', 'बवंडर', 'लाने',
'नवजीवन', 'नवयौवन', 'खेल', 'रचो', 'साथी', 'जीवन', 'वैसे', 'सबका', 'है', 'तुम', 'जीवन',
'श्रृंगार', 'बनो', 'इतिहास', 'तुम्हारा', 'राख', 'बना', 'तुम', 'राखों', 'अंगार', 'बनो', 'अव्याश',
'जवानी', 'होती', 'है', 'गत', 'वयस', 'कहानी', 'होती', 'है', 'तुम', 'सहज', 'लड़कपन', 'खेल',
'रचो', 'साथी']
```

Fig. 3. Sample extracted keywords



```
def get_keywords(self, pos):
KeywordsExtracted.txt -- Edited
['कर', 'चिंता', 'अब', 'अब', 'चिंता', 'चिंता', 'आज', 'ले', 'जा', 'रही', 'आज',
'रहे', 'आज', 'दिया', 'कर', 'आज', 'जा', 'कर', 'बैठक', 'कर', 'हुआ', 'जब',
'बन', 'गया', 'हुआ', 'शहर', 'बन', 'गया', 'आया', 'था', 'थाने', 'चिंता', 'सामने',
'अब', 'अब', 'कभी', 'अब', 'कभी', 'कर', 'सामने', 'करना', 'नौजवान', 'सामने',
'क्रम', 'आगे', 'पीछे', 'किया', 'है', 'जाना', 'नाम', 'संगीत', 'होता', 'है', 'बाहर',
'चले', 'देश', 'है']
```

Fig. 4. Sample extracted keywords ignoring 'Unk' tagged words

The extracted keywords were having certain very common words, which was removed by creating stop word list. After removing stop words the features were converted into their numeric representation, a sample features along with their numeric form is shown in Fig. 5.

```

{'क': 327, 'शत': 1079, 'गय': 392, 'आन': 158, 'मन': 695, 'बर': 815, 'पह': 766, 'अभय': 105, 'मा': 840, 'दद': 638,
'अर': 135, 'बद': 797, 'आग': 148, 'अभ': 104, 'आज': 150, 'इत': 191, 'गव': 400, 'स': 969, 'जय': 497, 'अला': 115,
'तत': 739, 'क्ष': 628, 'मम': 1147, 'रह': 1098, 'वदन': 1057, 'आत': 170, 'उत': 225, 'लपट': 1034, 'सद': 1130, 'फक':
773, 'लात': 1018, 'हत': 1193, 'अस': 126, 'शस': 1097, 'छव': 468, 'बल': 820, 'पहर': 770, 'वध': 1059, 'अवल': 46,
'पक': 715, 'रथम': 975, 'चण': 442, 'नप': 681, 'जन': 479, 'मन': 895, 'आई': 139, 'पहल': 771, 'वन': 1066, 'गत': 384,
'वल': 1069, 'पतव': 745, 'अम': 106, 'मह': 916, 'भम': 1107, 'खल': 366, 'समस': 1155, 'रभ': 986, 'वन': 809, 'चलत': 448,
'इव': 1218, 'श': 995, 'u200dनच': 14, 'खड': 357, 'आई': 371, 'मट': 881, 'सत': 1135, 'रह': 1002, 'उकत': 227, 'इन':
195, 'मश': 913, 'डग': 551, 'हुट': 149, 'उसक': 262, 'इर': 555, 'भण': 1102, 'कमल': 323, 'हम': 1199, 'घर': 415, 'अमर':
107, 'लहर': 1049, 'रथम': 994, 'bud1': 0, 'उन': 237, 'वसन': 1074, 'जल': 507, 'बचपन': 791, 'झर': 524, 'नयन': 698,
'पट': 407, 'सक': 1113, 'पहच': 767, 'कह': 349, 'तन': 589, 'चल': 447, 'जत': 475, 'झस': 528, 'नन': 690, 'कम': 321,
'लक': 1010, 'पख': 754, 'रज': 959, 'सदत': 1131, 'सपन': 1138, 'अपन': 93, 'पन': 749, 'अबाग': 118, 'णलन': 568, 'अपलक':
95, 'आनन': 159, 'नक': 682, 'धपन': 671, 'भत': 857, 'उर': 255, 'उपवन': 245, 'तर': 602, 'बह': 830, 'नवन': 706, 'मय':
901, 'दन': 641, 'लकन': 1012, 'खर': 363, 'लन': 1031, 'जग': 469, 'गन': 386, 'दर': 650, 'अभर': 73, 'झक': 511, 'ला':
1017, 'बह': 796, 'धर': 675, 'खत': 359, 'उनक': 238, 'चह': 424, 'चर': 753, 'कत': 344, 'भर': 855, 'रहक': 1004, 'झकझ':
512, 'वत': 1056, 'लय': 1041, 'बदलन': 806, 'बदलकर': 805, 'सस': 1176, 'छल': 464, 'ओछ': 284, 'मक': 865, 'जयच': 498,
'गद': 385, 'धमक': 674, 'यत': 936, 'गभ': 391, 'सय': 1156, 'मजहब': 879, 'बदन': 800, 'सकत': 1114, 'सदत': 1161, 'इसत':
205, 'जब': 491, 'बदल': 804, 'यम': 933, 'जलन': 506, 'गल': 396, 'वर': 1064, 'तब': 596, 'मर': 902, 'समझ': 1149, 'कड':
301, 'अब': 100, 'नसर': 620, 'भकर': 856, 'धत': 676, 'अध': 68, 'कल': 351, 'रा': 951, 'रा': 278, 'दव': 541, 'हमस':
1202, 'पह': 734, 'वलय': 1070, 'आलम': 172, 'दखल': 635, 'पछत': 728, 'कहल': 352, 'जह': 509, 'जमनन': 494, 'आश': 176,
'दत': 652, 'लत': 1027, 'अम': 251, 'भकन': 848, 'लस': 1046, 'मदम': 883, 'रच': 955, 'सदर': 611, 'जल': 501, 'कट': 298,
'बसनन': 819, 'उत': 230, 'उड': 229, 'पप': 372, 'सतर': 1128, 'फट': 774, 'वह': 1075, 'बसत': 817, 'तह': 585, 'मघल': 875,
'तसत': 609, 'सकर': 1115, 'हर': 1205, 'सतप': 1127, 'घन': 411, 'घत': 410, 'अनमन': 85, 'खड': 370, 'पल': 759, 'सड':
1184, 'पध': 746, 'दल': 657, 'इनक': 196, 'अदपट': 56, 'उलझ': 257, 'पकह': 719, 'अच': 44, 'कप': 313, 'मकड': 867, 'सर':
1157, 'मच': 872, 'छर': 463, 'इनन': 749, 'वहन': 1076, 'वनन': 192, 'सहन': 1180, 'कभ': 345, 'अजानर': 50, 'आमन': 41, 'बड':
837, 'गलज': 394, 'दह': 661, 'कनकन': 311, 'जवनन': 487, 'लाड': 579, 'आत': 152, 'सरसर': 1168, 'लपकत': 1033, 'मत': 885,
'कत': 330, 'गड': 406, 'इत': 199, 'मश': 888, 'जहर': 510, 'शम': 1086, 'मर': 393, 'नत': 688, 'कतम': 341, 'तलय': 615,
'शक': 1077, 'अप': 90, 'कक': 294, 'पकड': 716, 'जलत': 504, 'कम': 320, 'भय': 851, 'नर': 699, 'उगलत': 215, 'अक': 33,
'भर': 1109, 'वनत': 811, 'हरदम': 1207, 'लह': 1048, 'हल': 1214, 'महल': 922, 'रहत': 1006, 'चमकत': 434, 'दरत': 655, 'वत':
798, 'बजत': 793, 'सव': 1174, 'चत': 426, 'यल': 937, 'मध': 894, 'सत': 997, 'कण': 304, 'कमल': 326, 'रच': 958, 'चमक':

```

Fig. 5. Sample extracted features converted to numeric representation

## 4 Performance Tuning

SGDClassifier and Random Forest classification algorithms were used for this experiment, for better results of precision and recall along with accuracy as measure of performance for classification. Hyper parameter tuning was done for each of the algorithm and the model was Grid searched to find the best parameters for precision and recall.

The SGDClassifier used parameters loss, alpha, random\_state, and shuffle. The loss parameter was set to 'hinge' rest of the parameter was changed; alpha was set to 2 values  $1e-3$  and  $1e-4$ , random\_state was set to "1", "10", "100", "500", "1000" and shuffle was set to "True", "False" values. 20 combinations of parameters were used to search the best parameter for precision and 20 combinations for best parameters for recall. Duration of performance tuning with 20 Plus 20 combinations was 0:00:01.310747 h, i.e. 1:31 s.

The best parameter set for precision and recall was found to be same and is { 'alpha': 0.001, 'loss': 'hinge', 'random\_state': 1, 'shuffle': False }. A Subset of combinations of parameters along with accuracy is represented in Table 3 for precision and recall.

The random Forest algorithm used parameters n\_estimators, which decides number of decision tree to create the forest, random\_state with 4 different values and bootstrap to "True" and "False", there were 50 combinations of parameters for searching the best parameters for good precision and 50 combinations for recall. Duration of performance tuning with 50 Plus 50 combinations was 0:04:43.029051 h, i.e. 4 min and 43 s. The experiment shows that the best parameter combination found for precision score

**Table 3.** Subset of parameters used for parameter tuning for precision and recall

Parameters for SGDClassifier with loss = 'hinge'			
Alpha	Shuffle	Random state	Accuracy
0.001	False	1	56.06%
		10	56.06%
	True	100	56.03%
		10	52.38%
0.0001	False	1	55.30%
		1000	55.30%
	True	10	53.99%
		100	54.42%

is {'bootstrap': False, 'n\_estimators': 500, 'random\_state': None} and best parameter combination for recall score is {'bootstrap': False, 'n\_estimators': 500, 'random\_state': 42}. A subset of parameters combinations for precision along with accuracy achieved is shown in Table 4 and a subset of all the combinations of parameters used for recall is shown in Table 5.

**Table 4.** Subset of Parameters used for parameter tuning for precision

Parameters for Random Forest			
Bootstrap	Estimator	Random state	Accuracy
True	50	42	64.81%
		20	65.16%
	100	None	65.32%
		21	62.23%
True	250	21	64.52%
		20	62.89%
	500	None	63.61%
		42	62.13%

Random Forest is taking longer time for performance tuning than SGDClassifier.



**Table 5.** Subset of parameters used for parameter tuning for recall

Parameters for Random Forest			
Bootstrap	Estimator	Random state	Accuracy
True	500	42	57.02%
		None	56.51%
	1000	42	57.02%
		20	55.68%
False	50	21	59.25%
		20	59.16%
	500	None	59.49%
		42	60.03%

## 5 Experimentation Results

The performance tuning provided us with best parameters combinations, which can be used to train the classifier. The model was trained with 134 Poems and tested with 46 poems using SGDClassifier and Random Forest algorithms. Model was trained and tested using both the feature Set; the results were better when reduced feature set was used. The Feature Set used along with its statistics and results of the model in terms of accuracy in shown in Table 6.

**Table 6.** Results in accuracy

Classifier	Feature set with 8 keywords Unk	Accuracy	Feature set with 7 keywords	Accuracy
Random Forest	22,096	51.42	5,352	58.69%
SGDClassifier	22,096	40.00	5,352	50.00%

The results of classification is measured in terms of accuracy, but to know the details about how each class was classified precision and recall of each of the class is monitored. The Classification Report is shown in Table 7 for Random Forest and SGDClassifier in Table 8. Shringar and Shanta are having most overlapping features; bringing accuracy of the entire model down. In future fuzzy logic can be used to deal with the problem of overlapping. Karuna poems are the most correctly classified class of Poetries.

The model is trained with set of 134 poetries, and tested with 46 poetries, for a robust classifier it is better to train them with different set of data and test the performance, k – fold cross validation is done with k = 4 and k = 8 for both the classifiers, the results of each fold is shown in Table 9 for 4 folds and results are shown in Table 10 for 8 folds. In k –fold cross validation the data is divided into k equal portions called folds, 1 fold

**Table 7.** Classification report of Random Forest with reduced feature set

Class	Precision	Recall	F1 - score	Support
Karuna	0.92	0.85	0.88	13
Shanta	0.55	0.50	0.52	12
Shringar	0.37	0.64	0.47	11
Veera	0.75	0.30	0.43	10
Accuracy			0.59	46
Macro avg.	0.65	0.57	0.57	46
Weighted avg.	0.65	0.59	0.59	46

**Table 8.** Classification report of SGDClassifier with reduced feature set

Class	Precision	Recall	F1 - score	Support
Karuna	1.00	0.77	0.87	13
Shanta	0.42	0.42	0.42	12
Shringar	0.42	0.45	0.43	11
Veera	0.25	0.30	0.27	10
Accuracy			0.50	46
Macro avg.	0.52	0.49	0.50	46
Weighted avg.	0.55	0.50	0.52	46

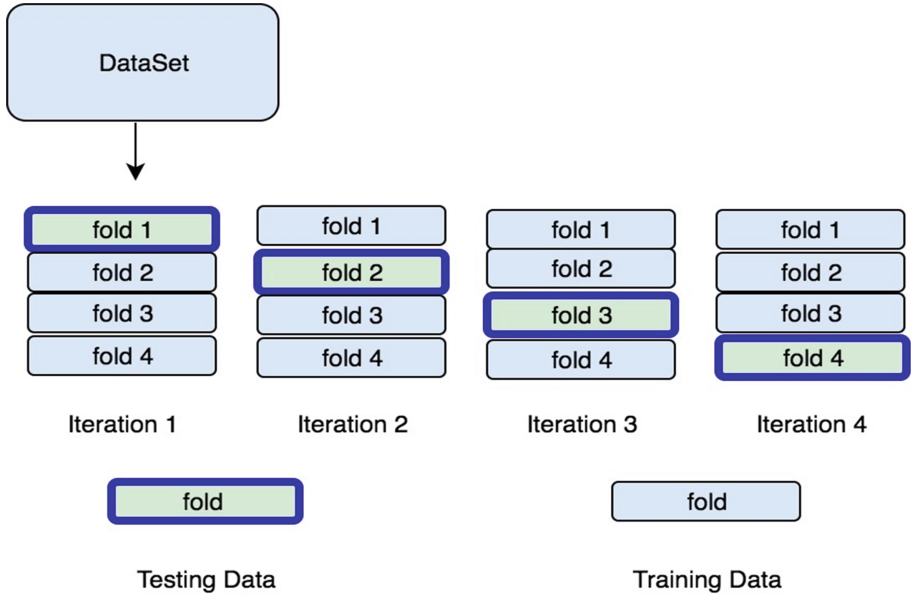
is used for testing and rest portions are used for training, the model is trained and tested for  $k$  times. Visualization of how  $k$ -fold works is shown in Fig. 6.

**Table 9.** Results of  $k$  - folds cross validation with  $k = 4$ 

Classifier/folds	Fold 1	Fold 2	Fold 3	Fold 4	Average accuracy
Random Forest	70.27%	64.86%	59.45%	55.55%	62.53%
SGDClassifier	62.16%	43.24%	56.75%	55.55%	54.42%

**Table 10.** Results of k - folds cross validation with k = 8

Classifier/folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Average Accuracy
Random Forest	63.15%	68.42%	63.15%	61.11%	50.00%	61.11%	66.66%	50.00%	60.45%
SGDClassifier	57.89%	47.36%	42.10%	27.77%	55.55%	61.11%	55.55%	38.88%	48.28%



**Data set division for k- fold cross validation for k = 4**

**Fig. 6.** K-fold cross validation data set division

The results of k – fold cross validation consistently shows good results, except 1 or 2 fold with poor accuracy. The range of results in accuracy using box plot is shown in Fig. 7 (a) and average accuracy using bar plot is shown in Fig. 7 (b) for k = 4. For k = 8 the results are visualized using box plot for showing range in Fig. 8 (a) and average results are shown in Fig. 8 (b) using bar plot.

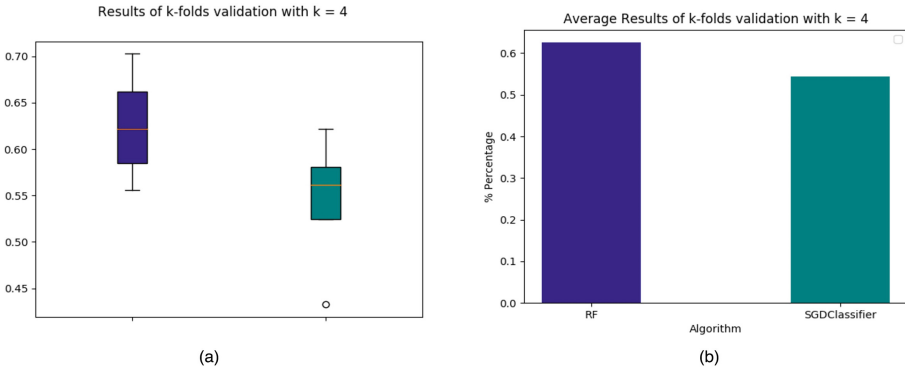


Fig. 7. K-fold cross validation with 4 folds (a) range of results (b) average accuracy

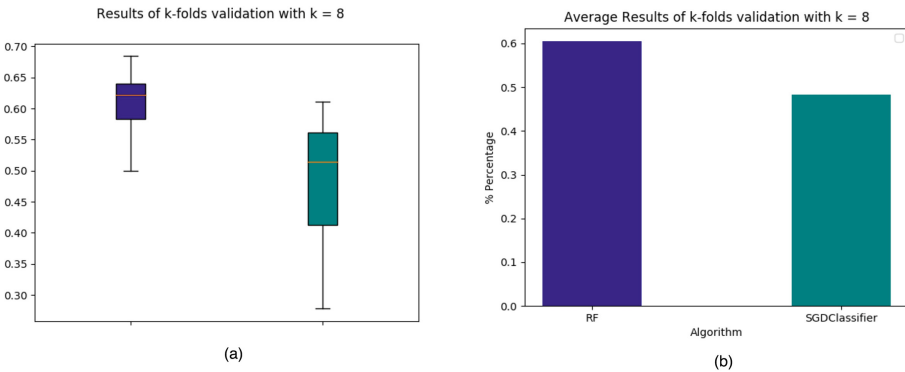


Fig. 8. K-fold cross validation with 8 folds (a) range of results (b) average accuracy

## 6 Conclusion

Emotion Classification in any form is challenging, this research work used 180 poetries and tagged using part-of-the-Speech tagging, and the data set was then partitioned into 134 poetries for training and 46 poetries for testing. The feature was extracted on keywords basis by manually monitoring the tagged files, two set of Feature set was prepared one using 8- keywords with 22,096 features and another reduced the feature set using 7 keywords with 5352 features. The experiment used Grid Search for performance tuning and experimented with 50 combinations of parameters using Random Forest for score precision and 50 combinations for score recall, to find the best parameter set for the dataset using Random Forest. To find best parameters set for SGDClassifier 20 combinations for precision and 20 combinations for recall were used. Both the algorithms trained the model one at a time using their best parameters found using performance tuning. The accuracy achieved with 8-keyword feature set was found to be 51.42% for Random Forest and 40% for SGDClassifier. Using reduced Feature set to train classifier; the classification accuracy was better with 58.69% accuracy for random forest and 50:00% accuracy for SGDClassifier. The results were also validated using k – fold cross

validation giving average results of 62.53% for 4 folds and 60.45% for 8 folds using Random Forest and 54.42% for 4 folds and 48.28% for 8 folds using SGDClassifier. The results of Random Forest are better compared to SGDClassifier in all scenarios.

## 7 Limitation and Future Work

The Classes Shanta and Shringar is having overlapping features which is troubling the performance of the model developed in this research work. In future fuzzy logic will be used to solve the overlapping feature problem.

Secondly POS tagger available for Hindi language available in NLTK is used which is tagging a lot of words in the poetry as 'Unk' meaning unknown, but observing the tagged poems shows that there are important words related to Hindi poetry which are tagged as 'Unk' but certain garbage values are also tagged as 'Unk'. Currently all those visible and not visible garbage values are cleaned with script in python. In future algorithm will be developed to extract important features from 'Unk' keywords extracted from tagged poems to make feature set rich for better emotion Classification.

## References

1. Shalini, M., Indira, B.: Implementation of Hindi word recognition and classification of system using artificial neural network. *Int. J. Pure Appl. Math.* **117**(15), 557–564 (2017)
2. Shalini, P., Satya Prakash, S.: A hybrid Hindi printed document classification system using SVM an Fuzzy: an advancement. *J. Inf. Technol. Res.* **12**(4), 107–131 (2019)
3. Jasleen, K., JatinderKumar, S.: PuPoCl: development of Punjabi poetry classifier using linguistic features and weighting. *INFOCOMP J. Comput. Sci.* **16**(1–2), 1–7 (2017)
4. Mandal, A.K.: Supervised learning method for bangla web document categorization. *Int. J. Artif. Intell. Appl.* **5**(5), 93–105 (2014)
5. Vandana, J., Manjunath, N.: Sentiment analysis in a resource scarce language: Hindi. *Int. J. Sci. Eng. Res.* **7**(6), 968–980 (2016)
6. Jasleen, K., JatinderKumar, S.: Emotion detection and sentiment analysis in text corpus: a differential study with informal and formal writing styles. *Int. J. Comput. Appl.* **101**(9), 1–9 (2014)
7. Noraini, J., Masnizah, M., Shahrul, A.: Poetry classification using support vector machines. *Int. J. Comput. Sci.* **8**(9), 1441–1446 (2012)
8. Hamid, R.: Poetic features for poem recognition: a comparative study. *J. Pattern Recognit. Res.* **3**, 24–39 (2008)
9. Kamath, C.N., Bukhari, S.S., Dengel, A.: Comparative study between traditional machine learning and deep learning approaches for text classification. In: *Proceedings of the ACM Symposium on Document Engineering 2018 DocEng 2018*, pp. 1–11 (2018). Article No.: 14
10. Bounabi, M., El Moutaouakil, K., Satori, K.: Text classification using fuzzy TF-IDF and machine learning models In: *BDIoT 2019: Proceedings of the 4th International Conference on Big Data and Internet of Things*, pp. 1–6 (2019). Article No.: 18
11. Puri, S., Singh, S.P.: An efficient Devanagari character classification in printed and handwritten documents using SVM. *Procedia Comput. Sci.* **152**, 111–121 (2019). <https://doi.org/10.1016/j.procs.2019.05.033>
12. Ishaan, T., Ashyush, C.: Classification of spam categorization on Hindi documents using Bayesian Classifier. *IOSR J. Comput. Eng.* **20**(6), 53–58 (2018)

13. Surkova, A., Skorynin, S., Chernobaev, I.: Word embedding and cognitive linguistic models in text classification tasks. In: Proceedings of the XI International Scientific Conference Communicative Strategies of the Information Society CSIS 2019, pp. 1–6 (2019). Article No.: 12
14. Anne, C., Mishra, A., Hoque, M.T., Tu, S.: Multiclass patent document classification. *Artif. Intell. Res.* **7**(1), 1–14 (2018)
15. Meng, Y., Shen, J., Zhang, C., Han, J.: Weakly-supervised neural text classification In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management CIKM 2018, pp. 983–992 (2018)
16. Liang, Q., Wu, P., Huang, C.: An efficient method for text classification task In: BDE 2019: Proceedings of the 2019 International Conference on Big Data Engineering pp. 92–97 (2019)
17. Tran, T.C.T., Huynh, H.X., Tran, P.Q., Truong, D.Q.: Text classification based on keywords with different thresholds In: ICIIT 2019: Proceedings of the 2019 4th International Conference on Intelligent Information Technology, pp. 101–106 (2019)
18. Islam, M.Z., Liu, J., Li, J., Liu, L., Kang, W.: A semantics aware random forest for text classification In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management CIKM 2019, pp. 1061–1070 (2019)
19. Zheng, W., Jin, M.: Do we need more training samples for text classification? In: Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference AICCC 2018, pp. 121–128 (2018)
20. Yao, R., Cao, Y., Ding, Z., Guo, L.: A sensitive words filtering model based on web text features In: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence CSAI 2018, pp. 516–520 (2018)
21. Pal, K., Patel, B.V.: A study of current state of work done for classification in Indian languages. *Int. J. Sci. Res. Sci. Technol.* **3**(7), 403–407 (2017)