International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST523103105

Classification of Gujarati Articles using Bernoulli Naïve Bayes Classifier and Extra-trees Classifier

Ravirajsinh Chauhan*, Janvi R Savani, Janvi M Sheta
P P Savani University, Surat, Gujarat, India

ARTICLEINFO

Article History:

Accepted: 05 May 2023
Published: 30 May 2023

Publication Issue
Volume 10, Issue 3
May-June-2023

Page Number
531-540

ABSTRACT

On the internet, information technology generated massive amounts of data. Because this data was initially primarily in English, the majority of data mining research was conducted on English text documents. As internet usage grew, so did data in other languages such as Gujarati, Marathi, Tamil, Telugu, and Punjabi, among others. We present a text categorization method based on artificial text summarization of Gujarati Articles in this paper. For the classification of text documents, various learning techniques such as Naïve Bayes, Support Vector Machines, and Decision Trees are available. We gathered articles from various e-newspaper editorials. This paper focuses on a brief review of the various techniques and methods for Gujarati Articles Classification, so that research in Text Classification can be further explored using various classifier algorithms. The dataset, which contains 1604 documents from 8 different categories, is used by the system. The result shows that Stacking Classifier with Bernoulli Naïve Bayes Classifier and Extra-trees Classifier is efficient for Gujarati Articles.

Keywords: Classification, Gujarati Articles, Natural Language Processing, Classifiers.

I. INTRODUCTION

"Text Classification" is one of the frequently utilized tasks for natural language processing in various commercial circumstances. To systematically classify text documents into one or more predefined categories is the aim of text categorization.

Gujarati is an Indo-Aryan language native to the Indian state of Gujarat and spoken predominantly by the Gujarati people. Gujarati is part of the greater Indo-European language family. The Gujarati language is more than 1000 years old and is spoken by more than 55 million people worldwide.

Moreover, there are numerous advertising and marketing companies which publish articles and e-newspapers in the Gujarati language. Also, there are thousands and a greater number of Gujarati articles or documents available on the internet. So, it is a very

time-consuming process to classify it manually, which is nearly impossible to do.

Text mining in Gujarati language is a very challenging process as there are many researches have done on text classification of Gujarati language. It provides a huge opportunity for research work in various text mining techniques in the Gujarati language. Nowadays, Organizations and enterprises struggle with a serious issue called information overload on the internet. Due to the language barrier and grammatical diversity of the Indian language, sorting out some valuable articles from the web that are published in that language might be difficult. There is no document classifier for Gujarati as of right now.

Search engines, digital library systems, and document management systems have all used automatic text categorization [1]. Electronic email filtering, newsgroup classification, and survey data grouping are examples of such applications. For example, Barq employs automatic categorization to provide a similar documents feature [2].

The following are the fundamentals of Gujarati language and machine learning approach:

- Gujarati Language

Gujarati is the official and regional language of India's Gujarat state. It is the world's 23rd most widely spoken language, with over 46 million people speaking it. Gujarati is spoken by approximately 45.5 million people in India, with an additional half million speakers from Tanzania, Uganda, Pakistan, Kenya, and Zambia. Gujarati is an Indo-Aryan language of the Indo-European language family, and it is closely related to Indian Hindi.

- Naïve Bayes (Supervised Machine Learning Algorithm)

The Naïve Bayes (NB) algorithm is the most widely used statistical machine learning algorithm for text classification. In terms of simplicity, the Naïve Bayes algorithm may be superior to several existing approaches for document classification (such as decision trees, neural networks, and support vector machines). NB performed admirably in a wide range of real-world applications, including document and text classification, but a small amount of training is required to estimate the required parameters.

The organization of this document is as follows. In Section 2 (**Related Work**), I'll give detail of work done till date in this field. In Section 3 (**Proposed work**), present proposed system. In Section 4 (**Experimental results**) present your research findings and your analysis of those findings. Discussed in Section 5(**Conclusion**) a conclusion is the last part of something, its end or result.

II. RELATED WORK

For many years, many machine learning algorithms have been used to categorize text. Early works on decision tree learning and Bayesian learning, nearest neighbour learning, and artificial neural networks can be found in [3], [4], and [5].

The authors of [6] proposed a hybrid Associative Classification (AC) with the Naïve Bayes algorithm (NB). The AC model suffers from a large number of classification rules and the use of various pruning methods, which remove some important information and thus influence the correct decision. These disadvantages, according to the authors, were addressed by using NB. By integrating mining association rules with the classification task, the proposed mechanism improved the efficiency of Arabic text classification.

The authors of [7] used DL to classify Arabic text. They extracted, selected, and reduced the collected features using stemming. As a feature weighting technique, the TF-IDF scheme was applied to the documents. Finally, CNN classification was applied to a variety of benchmarks with positive results.

Sci-kit learn is a machine learning library for the Python programming language. This library includes

various classification, regression, and clustering algorithms such as SVM (support vector machine), Random Forest, and k-means clustering. There are three types of Naïve Bayes models in this library [8]. These are:

- Gaussian: It ensures that the features are following normal.
- Multinomial: It is used when we have discrete count.
- Bernoulli: it is useful when we have binary feature vectors.

The authors of [9] discussed the computational linguistics implementations they used to perform linguistic analysis on tweets in order to observe patterns exhibited by legitimate and fake or ambiguous news. They deconstructed the tweets' grammar for in-depth analysis and built a comprehensive BoW model based on the categorised labelled tweets. They compared how polarity and subjectivity differ between legitimate and polarised tweets while keeping the topics of the tweets constant.

In [15] they used two different classifiers to identify improved performance in a language like English. The use of a word-level N-Gram feature for word vectorization in conjunction with logistic regression (LR) and NB improved performance.

The authors of [16] used the Naïve Bayes, k-NN, and Centroid Based Classification methods. They extracted features from a Marathi word dictionary. Marathi documents were classified into five categories: literature, economy, botany, geography, and history. They produced over 800 documents in each category. They achieved higher accuracy by using the Naïve Bayes classifier, whereas the k-NN classifier produced the lowest accuracy.

In the paper [17], they used the Naïve Bayes, SVM, and k-NN classification methods. They used the TF-IDF method for feature selection and collected data from 800 documents on the web (Telugu newspapers) in the fields of science, economics, sports, politics, culture, and health. Their findings revealed that SVM outperformed Naïve Bayes and k-NN.

In [18], authors employed a Naïve Bayes, Centroid-based, Ontology-based, and Hybrid-based Classification approach. They chose features using the TF-IDF method. They generated 180 documents from the internet (Panjabi newspapers) For data collection, the sports categories of Cricket, Football, Kabaddi, Tennis, Hockey, Badminton, and Olympics were used. In their results, Naïve Bayes provided 64% accuracy. The centroid-based classification approach provided 71% accuracy. The ontology-based classification approach achieved 85% accuracy. The hybrid-based classification approach achieved 85% accuracy. They achieved the same results in both the hybrid and ontology-based classification approaches.

In [19], authors proposes a complex structure for detecting fake news. The technique uses a machine learning model to classify incidents. The model includes five NLP features and three knowledge verification features in the form of questions about the source's scope, spread, and consistency. Limitation: The concept of similarity is essential for automated knowledge verification. It is especially dependent on establishing semantic similarity.

In [12] they propose two empirical heuristics in this paper: per-document text normalization and the feature weighting method. While these are somewhat haphazard methods, their proposed naïve Bayes text classifier outperforms state-of-the-art text classifiers based on a highly complex learning method such as SVM in standard benchmark collections.

III.PROPOSED WORK

The proposed work focuses on automating the time-consuming manual categorization of Gujarati documents using a text classification model. The model's goal is to improve previous work on Gujarati language text classification and increase scalability for article categorization. As a result, there is a lot of room for research in this field to improve the scalability, accuracy, and quality of Text Classification Model.

The document classification feature allows the user to upload multiple documents at once and categorize them. It facilitates the processing of various document types and assigns them to the appropriate team member for review, processing, and analysis. Correctly classifying a business is an important step in providing risk coverage because it supports the rating structure and allows an insurance carrier to charge a rate commensurate with business exposures. The answer to manual document classification is automatic document classification, which is much faster and more accurate. Documents are identified, classified, sorted, split, assembled, and processed as they are ingested into an IDP system.

In this paper we have used Stacking. Stacking is the process of combining multiple classifiers produced by different machine learning algorithms. It is a two-phase process that generates a set of base classifiers in the first phase and then combines these base classifiers in the second phase to generate a meta-classifier [10]. Stacked generalization involves stacking individual estimator output and using a classifier to compute the final prediction. Stacking allows you to capitalize on the strengths of each individual estimator by feeding their output into a final estimator. The final estimator is a classifier that will combine the base estimators. In this paper Bernoulli Naïve Bayes (NB) classifier and Extra Tree Classifier are Base estimators and Random Forest classifier is Final estimator.

The proposed work employs the Bernoulli Naïve Bayes (NB) classifier and Extra Tree Classifier.

- Bernoulli Naïve Bayes (NB) classifier

The Bernoulli Naïve Bayes is a variation of the Naïve Bayes algorithm used in machine learning that is particularly useful in binary distributions where the output label can be present or absent. This algorithm's main advantage is that it only accepts features in the form of binary values such as True or False, Spam or Ham, Yes or No, 0 or 1.

- Extra Tree Classifier

It is an estimator that attempts to fit randomized decision trees on different sub-samples of the dataset and employs the concept of averaging to improve accuracy and control over data fitting. They differ from traditional decision trees in the way they are built. Rather than looking for the best split to divide a node's samples into two groups, random splits are drawn for each of the max_features randomly selected features, and the best split among those is chosen [11].

3.1 SYSTEM DESIGN

Text Classification is a supervised machine learning task because it uses a labelled dataset containing text documents and their labels to train a classifier. An end-to-end text classification pipeline is made up of four major parts:

1. Dataset Preparation: The first step is Dataset Preparation, which includes loading a dataset and performing basic pre-processing. After that, the dataset is divided into train and validation sets.
2. Feature Engineering: The next step is Feature Engineering, which involves transforming the raw dataset into flat features that can be used in a machine learning model. This step also includes the creation of new features from existing data.
3. Model Training: The final step is Model Building, which involves training a machine learning model on a labelled dataset.
4. Improve Text Classifier Performance: In this Paper, we will look at various methods for improving text classifier performance.

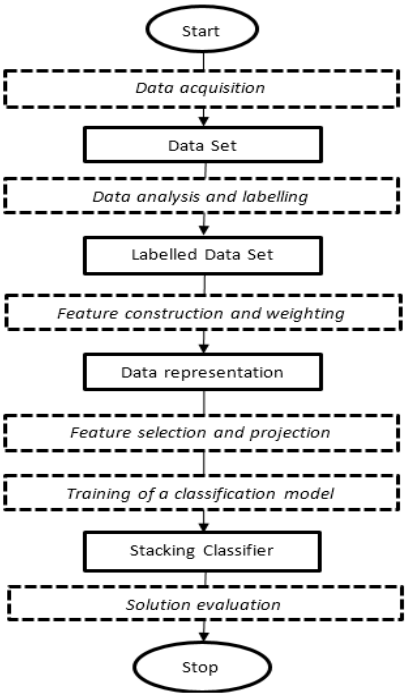

Figure 2. UML Diagram for Model Development

3.2 DATASET

To assist with the proposed work, we have created a dataset containing Gujarati articles. We collected 1604 Gujarati articles from the internet (from various news publications/newspapers). We collected all of the categories separately so that we could easily label each article. We initially saved collected articles in a notepad file for easy access, then converted these notepad files into a python-compatible format. These articles are divided into the following eight categories:

Table 1. Category wise Number of articles in Dataset

Sr.No.	Category	No. of Articles
1	Astrology	200
2	Business	207
3	Entertainment	200
4	International News	200
5	Lifestyle	213
6	Politics	180
7	Science and Technology	200
8	Sports	204

3.3 EXPLORATORY DATA ANALYSIS (EDA)

By extracting patterns and testing hypotheses to identify anomalies, exploratory data analysis plays a significant role in learning the hidden structures that encompass the significant features of the data in an ordered manner. The graph below shows that the data is balanced, and there were no missing values in this dataset because it was created by hand.

Figure 3. Balanced Data

In Exploratory Data Analysis pair plots are used to determine the best set of features to explain a relationship between two variables or to form the most distinct clusters. It also helps to form some simple classification models in our data-set by drawing some simple lines or making linear separation. The figure below shows that the number of characters, words, and sentences are distributed equally across each category.

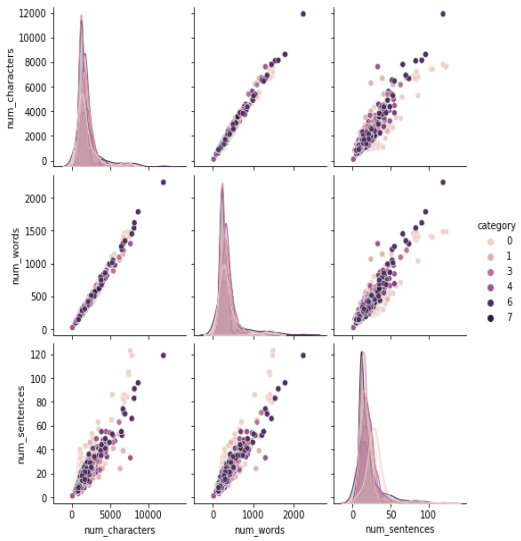

Figure 4. Relation between characters, words and sentences of each category

3.4 PRE-PROCESSING

Data pre-processing is absolutely essential in many research areas, including NLP, DM, and ML. It allows for the enhancement of the raw experimental data's quality. The performance of supervised learning models is significantly influenced by data pre-processing [13]. The primary goal of pre-processing is to reduce the test space and error rate. The next step in text classification is appropriate data pre-processing and data analysis [14]. The following stages are included in data pre-processing:

3.4.1 Tokenization

The tokenization process divides the dataset into individual words/ tokens. Multiple delimiters, such as white spaces, tabs, and punctuation marks, are used to separate the words. The tokenization process produces two types of output: tokens that correspond to recognizable units such as punctuation marks, numeric data, dates, and so on, and tokens that require further morphological analysis. Tokens of one or two characters in length, non-Gujarati characters, or numerical values are ignored and removed from the dataset because they degrade the classifier's performance [7].

3.4.2 Removing Stop-words

Typically, stop-words are functional words. They include things like conjunctions and prepositions. They appear frequently in a text and have little influence on the classification process. We have compiled a list of all the Gujarati stop words.

We removed punctuation, English words, and numbers from Gujarati text after removing stop-words to improve the classifier's performance.

IV. EXPERIMENTAL RESULTS

In this section, we discuss model construction and compare our model to other classifiers. The model is evaluated using standard metrics such as precision, recall, F1-Score and accuracy. A Confusion Matrix, which is a visual summary of the classification prediction results, is generated to evaluate the accuracy of classification models. The number of correct and incorrect predictions for each class is counted and totalled. The confusion matrix provides a detailed breakdown of the classifier's mistakes and misclassified instances. As mentioned, model evaluation is done using standard metrics which are calculated by,

$$Precision = \frac{TP}{TP + FP} = \frac{true\ positive}{total\ predicted\ positive} \quad (1)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} = \frac{true\ positive}{total\ actual\ positive} \quad (3)$$

$$F1\_Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (4)$$

We used Python 3.10.1 to implement the classification models, along with Jupyter Notebook 6.4.5 and Scikit-Learn 1.1.1., Stacking Classifier with Bernoulli Naïve Bayes Classifier and Extra-trees Classifier put to the test. With precision value of 1.00, recall value of 1.00, and F1-score value of 1.00, the “Entertainment” class displayed the model's best

performance followed by "Astrology" with precision value of 1.00, recall value of 0.95, and F1-score value of 0.97. Overall, an accuracy score of 0.94 was obtained using stacking classifier. Table 2 shows detailed precision value, recall value and f1-score of all the categories. Fig. 5 shows confusion matrix. The support values in the confusion matrix represent the frequency with which each specific category appears in the actual responses (responses in our test dataset).

In Fig. 5, 0 = "Astrology", 1 = "Business", 2 = Entertainment, 3 = International News, 4 = Lifestyle, 5 = Politics, 6 = Science and Technology and 7 = Sports.

Table 2. Classification report

Sr. No.	Category	Precision
1	Astrology	1.00
2	Business	0.88
3	Entertainment	1.00
4	International News	0.98
5	Lifestyle	0.76
6	Politics	0.94
7	Science and Technology	0.97
8	Sports	0.97

Sr.No.	Category	Recall
1	Astrology	0.95
2	Business	1.00
3	Entertainment	1.00
4	International News	0.92
5	Lifestyle	0.83
6	Politics	0.84
7	Science and Technology	0.95
8	Sports	1.00

Sr.No.	Category	F1-Score
1	Astrology	0.97
2	Business	0.93
3	Entertainment	1.00
4	International News	0.95
5	Lifestyle	0.79
6	Politics	0.89
7	Science and Technology	0.96
8	Sports	0.99

Sr.No.	Category	Support
1	Astrology	37
2	Business	43
3	Entertainment	48
4	International News	50
5	Lifestyle	30
6	Politics	38
7	Science and Technology	38
8	Sports	37

We employed numerous classifiers in an effort to increase accuracy, but the stacking classifier provided the best accuracy and precision. Table 3 lists the classifiers we used to test our dataset, along with their calculated accuracy and precision. We used Bernoulli (Naïve Bayes classifier), Extra Trees classifier as Base estimators and Random Forest classifier as Final estimators in our Stacking Classifier because they show the best accuracy and precision score. We made an effort to improve accuracy and precision score by changing the maximum number of features (max_features) in the TF-IDF vectorizer, providing 4000 and 5000 max features at a time but the outcome was not what we had hoped for. With the help of max features, we can restrict the number of features (words) from the dataset that we want to use to determine the TF-IDF scores. We also used count vectorizer instead of TF-IDF, but TF-IDF produced more accurate results. Table 4 and Table 5 shows accuracy and precision we obtained by changing maximum number of features.

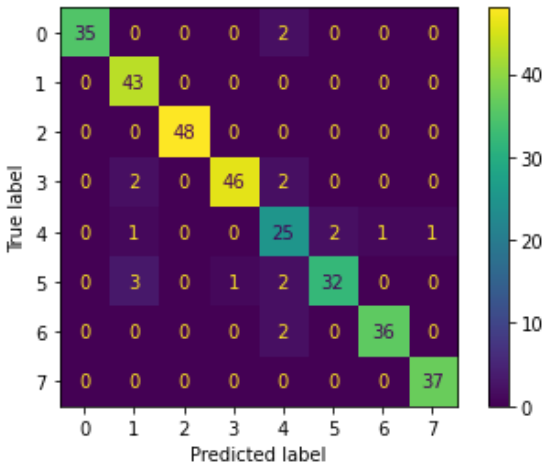

Figure 5. Confusion Matrix

Table 3. Accuracy and Precision report of all tested classifiers

Sr.No.	Algorithm	Accuracy	Precision
1	Bernoulli (Naïve Bayes classifier)	0.912773	0.915583
2	Extra Trees Classifier	0.890966	0.898153
3	Random Forest	0.884735	0.891777
4	Multinomial (Naïve Bayes classifier)	0.841121	0.886874
5	Gradient Boosting Classifier	0.875389	0.879121
6	XGB Classifier	0.878505	0.878757
7	Support Vector Machine	0.862928	0.873974
8	Bagging classifier	0.872274	0.873224
9	Gaussian (Naive Bayes classifier)	0.82866	0.83622
10	K-Nearest Neighbours	0.788162	0.792388
11	Logistic Regression	0.747664	0.769356
12	Decision Tree	0.389408	0.576217
13	AdaBoost	0.277259	0.352205

All of the tested classifiers' accuracy and precision are displayed in a horizontal bar chart format in Figure 6. Bernoulli (Naive Bayes Classifier) is the best performing classifier, as shown in Table 3 and Figure 6.

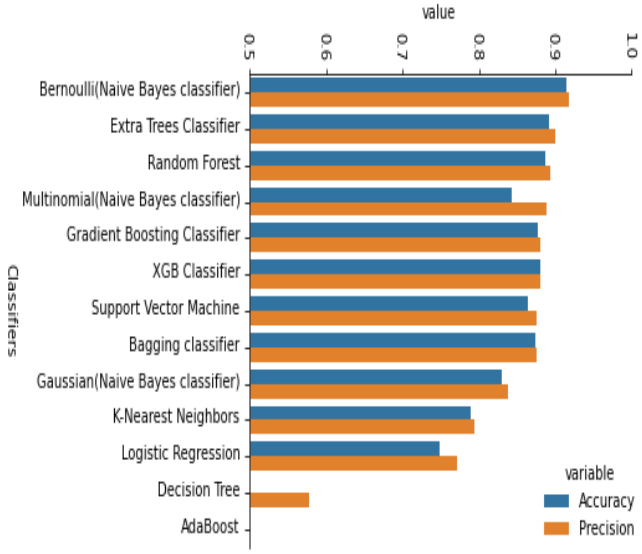

Figure 6. Accuracy and Precision chart of tested classifiers

Table 4. Classification Report with 4000 Maximum Features of TF-IDF

Sr. No.	Algorithm	Accuracy	Precision
1	Bernoulli (Naïve Bayes classifier)	0.915888	0.91972
2	Extra Trees Classifier	0.88785	0.897406
3	Random Forest	0.894081	0.89017
4	XGB Classifier	0.88785	0.888883
5	Multinomial (Naïve Bayes classifier)	0.841121	0.885545
6	Support Vector Machine	0.875389	0.883074
7	Gradient Boosting Classifier	0.88162	0.880754
8	Bagging classifier	0.884735	0.880095
9	Gaussian (Naive Bayes classifier)	0.82866	0.83622
10	K-Nearest Neighbours	0.775701	0.779936

11	Logistic Regression	0.747664	0.769356
12	Decision Tree	0.389408	0.576217
13	AdaBoost	0.29595	0.341138

Table 5. Classification Report with 5000 Maximum Features of TF-IDF

Sr. No.	Algorithm	Accuracy	Precision
1	Bernoulli (Naïve Bayes classifier)	0.915888	0.91972
2	Extra Trees Classifier	0.890966	0.901924
3	Random Forest	0.88785	0.889033
4	XGB Classifier	0.878505	0.878041
5	Multinomial (Naïve Bayes classifier)	0.841121	0.885545
6	Support Vector Machine	0.866044	0.876073
7	Gradient Boosting Classifier	0.869159	0.871132
8	Bagging classifier	0.872274	0.87245
9	Gaussian (Naïve Bayes classifier)	0.82866	0.83622
10	K-Nearest Neighbours	0.778816	0.782112
11	Logistic Regression	0.747664	0.769356
12	Decision Tree	0.389408	0.576217
13	AdaBoost	0.274143	0.351038

V. CONCLUSION

In this paper, we show how a supervised Classification model built with a stacking classifier can be used to solve the problem of multiclass, single-label Gujarati article classification. We created our own dataset of articles divided into eight categories. We used the Bernoulli Naïve Bayes Classifier, Extra-trees Classifier (Base estimators), and Random Forest Classifier (Final estimator) to implement the stacking classifier. We used a variety of classifiers to improve the accuracy and precision of our model, with the stacking classifier achieving the highest accuracy score of 0.94. The concepts presented in this paper can also be applied to the broader domain of text classification.

VI. REFERENCES

[1] Y. Yang, “An evaluation of statistical approaches to text categorization,” Journal of Information Retrieval, Vol. 1, Number 1-2, pp. 69--90, 1999.

[2] Rachidi, Tajje-eddine & Iraqi, Omar & Bouzoubaa, M. & Khattab, A.B.E. & Kourdi, M.E. & Zahi, Abdelali & Bensaid, A. (2003). Barq: distributed multilingual internet search engine with focus on Arabic language. 1. 428 - 435 vol.1. 10.1109/ICSMC.2003.1243853..

[3] D. Lewis, M. Ringnette, “Comparison of two learning algorithms for text categorization,” Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94), 1994.

[4] R. H. Creecy, B. M. Masand, S. J. Smith, and D. L. Waltz, “Trading mips and memory for knowledge engineering,” Communication of the ACM, Vol. 35, No. 8, pp. 48--64, August 1992.

[5] (Wiene and Pedersen, 1995) E. Wiener, J. O. Pedersen, and A. S. Zeigend, “A neural network approach to topic spotting,” Proceedings of the Fourth Symposium on Document Analysis and Information Retrieval (SDAIR'95), 1995.

[6] W. Hadi, Q. A. Al-Radaideh, S. Alhawari, “Integrating Associative Rule-based Classification with Naïve Bayes for Text Classification,” Applied Soft Computing, 69, pp. 344-356, 2018.

[7] S. Boukil, M. Biniz, F. El Adnani, L. Cherrat, and A. E. El Moutaouakkil, “Arabic Text Classification Using Deep Learning Technics,” International Journal of Grid and Distributed Computing,11(9), pp. 103-114, 2018.

[8] Singh, Mandeep, et al. "Performance of bernoulli's naive bayes classifier in the detection of fake news." Materials Today: Proceedings (2020).

[9] Dey, A., Rafi, R. Z., Parash, S. H., Arko, S. K., & Chakrabarty, A. (2018, June). Fake news pattern recognition using linguistic analysis. In 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR) (pp. 305-309). IEEE.

[10] Chand, N., Mishra, P., Krishna, C. R., Pilli, E. S., & Govil, M. C. (2016, April). A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection. In 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring) (pp. 1-6). IEEE.

[11] L. Abhishek, "Optical Character Recognition using Ensemble of SVM, MLP and Extra Trees Classifier," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154050.

[12] Kim, Sang-Bum, et al. "Some effective techniques for naive bayes text classification." IEEE transactions on knowledge and data engineering 18.11 (2006): 1457-1466.

[13] Kotsiantis, Sotiris B., Dimitris Kanellopoulos, and Panagiotis E. Pintelas. "Data preprocessing for supervised leaning." International journal of computer science 1.2 (2006): 111-117.

[14] Sundus, Katrina, Fatima Al-Haj, and Bassam Hammo. "A deep learning approach for arabic text classification." 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS). IEEE, 2019.

[15] Narayanan, Vivek, Ishan Arora, and Arjun Bhatia. "Fast and accurate sentiment classification using an enhanced Naive Bayes model." Intelligent Data Engineering and Automated Learning–IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, October 20-23, 2013. Proceedings 14. Springer Berlin Heidelberg, 2013.

[16] Patil, Meera, and Pravin Game. "Comparison of Marathi text classifiers." International Journal on Information Technology 4.1 (2014): 11.

[17] Murthy, Vishnu G., et al. "A comparative study on term weighting methods for automated Telugu text categorization with effective classifiers." International Journal of Data Mining & Knowledge Management Process 3.6 (2013): 95.

[18] Krail, Nidhi, and Vishal Gupta. "Domain based classification of Punjabi text documents using ontology and hybrid-based approach." Proceedings of the 3rd Workshop on south and Southeast Asian natural language processing. 2012.

[19] Ibrishimova, Marina Danchovsky, and Kin Fun Li. "A machine learning approach to fake news detection using knowledge verification and natural language processing." Advances in Intelligent Networking and Collaborative Systems: The 11th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2019). Springer International Publishing, 2020.

Cite this article as :

Ravirajsinh Chauhan, Janvi R Savani, Janvi M Sheta, "Classification of Gujarati Articles using Bernoulli Naïve Bayes Classifier and Extra-trees Classifier", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 10 Issue 3, pp. 531-540, May-June 2023. Available at doi : https://doi.org/10.32628/IJSRST523103105
Journal URL : https://ijsrst.com/IJSRST523103105