Multi-Class Classification of Hindi Text using Machine Learning

G. Bharath Teja[1], E.Bharghav Rao[2], P.Bhargavi[3], M.Bhargavi[4], G.Vijay Bhaskar Reddy[5], Prof. Sameera Sultana [6]

[1]Department of Artificial Intelligence and Machine Learning, Malla Reddy University
2111cs020085@mallareddyuniversity.ac.in ,
[2]Department of Artificial Intelligence and Machine Learning, Malla Reddy University
2111cs020086@mallareddyuniversity.ac.in,
[3]Department of Artificial Intelligence and Machine Learning, Malla Reddy University
2111cs020087@mallareddyuniversity.ac.in ,
[4]Department of Artificial Intelligence and Machine Learning, Malla Reddy University
2111cs020088@mallareddyuniversity.ac.in ,
[5]Department of Artificial Intelligence and Machine Learning, Malla Reddy University
2111cs020090@mallareddyuniversity.ac.in ,
[6]Department of Artificial Intelligence and Machine Learning, Malla Reddy University
Sameera_sultana@mallareddyuniversity.ac.in

ABSTRACT

This study focuses on the multi-class classification of Hindi texts using ML techniques. The goal is to develop an efficient model that can accurately classify Hindi text documents into various predefined categories. We explore the preprocessing steps necessary for cleaning and preparing the text data, tokenization, stop-word removal, and stemming. Feature extraction methods such as TF-IDF and word embeddings are employed to represent the text data numerically. It includes ML algorithm like Supervised Learning algorithm (Naïve Bayes, SVM, Decision tree and Random Forest) are experimented with to determine the best-performing model. The dataset used for training and testing consists of a diverse range of Hindi texts from different domains. The experimental results provide insights into the effectiveness of different approaches and algorithms, shedding light on the challenges and opportunities of Hindi texts.

Keywords: Multi-Class Classification, Hindi Texts, Machine Learning Algorithms, Natural Language Processing, Text Classification, Feature Extraction, Evaluation Metrics.

I. INTRODUCTION

This study focuses on the multi-class classification of Hindi texts, leveraging machine learning techniques to address the critical need for tasks such as sentiment analysis, topic categorization, and content recommendation. The crucial preprocessing steps necessary to prepare Hindi text data for classification. These steps involve tokenization, stop-word removal, stemming, and addressing challenges unique to the Hindi script. The study delves into feature extraction methods, considering techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings such as Word2Vec and FastText. The traditional models like Naïve Bayes, Support Vector Machines and advanced approaches including Random Forests. To assess the models, a comprehensive dataset of labeled Hindi textsspanning multiple categories is utilized for training, validation, and testing. Evaluation metrics encompass accuracy, precision, recall, F1-score, and confusion matrices, providing a holistic view of the models' capabilities.Beyond contributing to the advancement of natural language processing techniques for Hindi, this research offers valuable insights into the suitability of different algorithms for similar tasks in other languages.

II. PROBLEM STATEMENT

Develop a machine learning model for the multi-class classification of Hindi texts into predefined categories. The goal is to create a robust system capable of accurately assigning one or more labels to a given Hindi text, where each label represents a specific class or category. The focus is on genres or topics relevant to the context of the texts, such as sentiment analysis, genre identification, or topic categorization.

III. LITERATURE REVIEW

The multi-class classification of Hindi text using machine learning has gained substantial attention in recent literature due to the escalating volume of digital content in the Hindi language. Scholars underscore the significance of robust preprocessing steps for effective text classification in Hindi. Tokenization, stop-word removal, stemming, and addressing script-specific challenges have been identified as crucial tasks, impacting the accuracy and efficiency of classification models for Hindi textFeature extraction plays a pivotal role in capturing the semantic nuances of Hindi text. Studies have explored the effectiveness of traditional techniques such as TF-IDF, as well as embeddings like Word2Vec and FastText. Traditional models like Naïve Bayes and Support Vector Machines have demonstrated competitive performance in handling diverse Hindi textual data. Recent research has introduced novel approaches using ensemble methods such as Random Forests and Gradient Boosting, revealing promising results.Some studies have explored the transferability of models trained on English or other languages to Hindi text classification tasks. Transfer learning approaches, where pre-trained models are fine-tuned on Hindi datasets, have demonstrated promising resultsModels need to be culturally sensitive, taking into account linguistic variations and cultural idiosyncrasies in the Hindi script.

IV. METHODS AND MATERIAL

Dataset:

The study utilizes a comprehensive dataset comprising labeled Hindi texts across multiple categories. The dataset is carefully curated to ensure diversity in topics and genres, enabling the models to generalize well. A substantial volume of text data is essential to train, validate, and test the classification models effectively.

Preprocessing Steps:

Tokenization: The Hindi text data undergoes tokenization to break down sentences into individual tokens or words. Special consideration is given to the script-specific challenges of the Hindi language during this process.

Stop-word Removal: Common stop words in Hindi are removed to focus on meaningful content.

Handling Hindi Script Challenges: Specific preprocessing steps are implemented to address challenges unique to the Hindi script, such as variations in compound words and characters.

Feature Extraction Methods:

TF-IDF (Term Frequency-Inverse Document Frequency): This traditional technique is employed to weigh the importance of words in the corpus, considering both their frequency in a document and their rarity across the entire dataset.

Machine Learning Algorithms:

Naïve Bayes: A probabilistic model based on Bayes' theorem, well-suited for text classification tasks.

Support Vector Machines (SVM): Effective for high-dimensional data, SVM is applied to find optimal hyperplanes for class separation.

Random Forests : Ensemble methods are employed to combine the strengths of multiple models and enhance overall performance.

Evaluation Metrics:

Accuracy: Measures the overall correctness of the classification.

Precision: Assesses the accuracy of positive predictions.

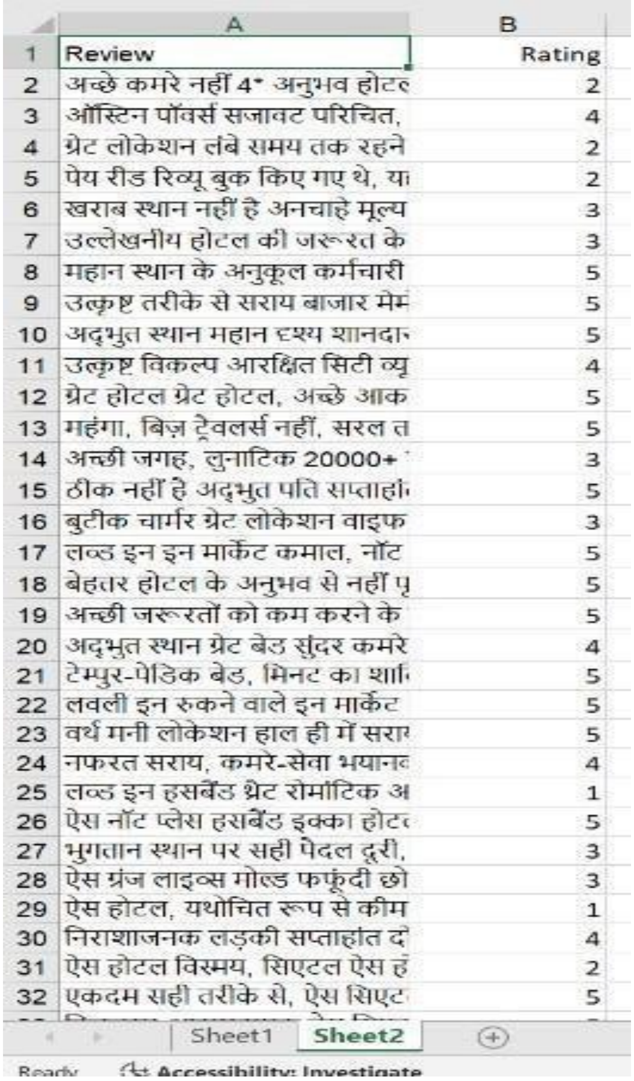
F1-score: Balances precision and recall, providinga harmonic mean.

V. RESULTS AND DISCUSSION

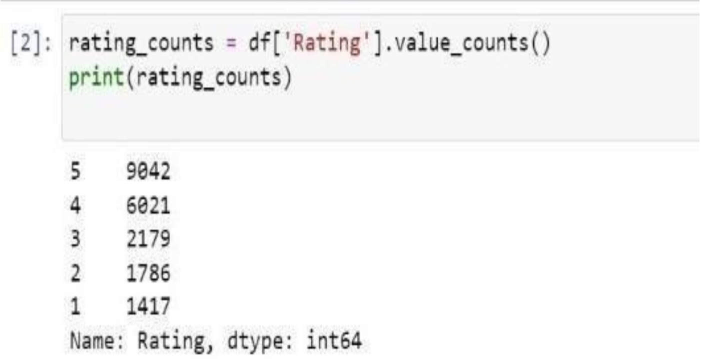
Accuracy Score: Measure of how accurately the models predicted the validation set.

Classification Report: Provides precision, recall, F1-score, and support for each class. Helps to understand the model's performance for individual classes.

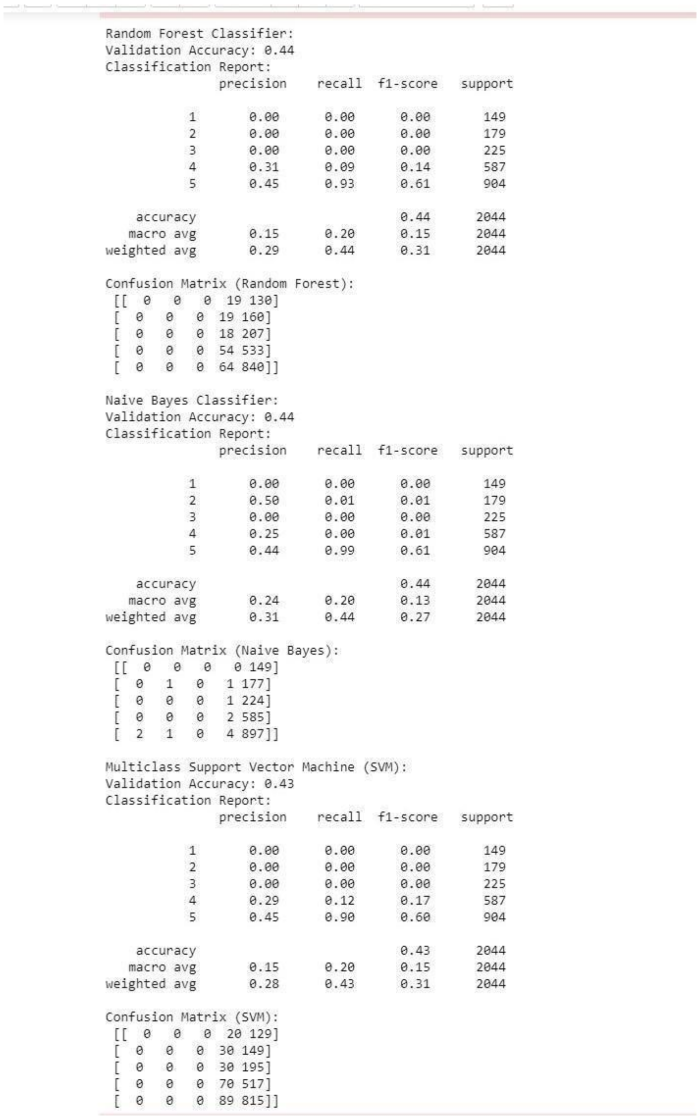
Confussion Matrix: Shows the count of true positive, true negative, false positive, and false negative values. Useful to understand the model's misclassifications.



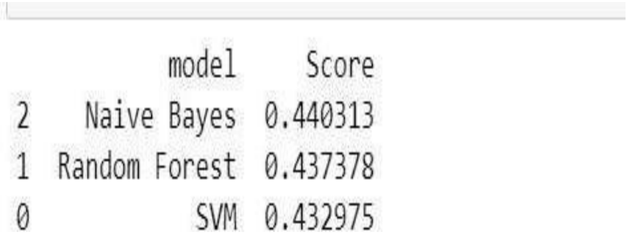
	A	B
1	Review	Rating
2	अच्छे कमरे नहीं 4* अनुभव होटल	2
3	ऑस्टिन पॉवर्स सजावट परिचित,	4
4	ग्रेट लोकेशन लंबे समय तक रहने	2
5	पेय रीड रिव्यू बुक किए गए थे, या	2
6	खराब स्थान नहीं है अनचाहे मूल्य	3
7	उल्लेखनीय होटल की जरूरत के	3
8	महान स्थान के अनुकूल कर्मचारी	5
9	उत्कृष्ट तरीके से सराय बाजार मेम	5
10	अद्भुत स्थान महान दृश्य शानदा	5
11	उत्कृष्ट विकल्प आरक्षित सिटी व्यू	4
12	ग्रेट होटल ग्रेट होटल, अच्छे आक	5
13	महंगा, बिज़ ट्रैवलर्स नहीं, सरल त	5
14	अच्छी जगह, लुनाटिक 20000+	3
15	ठीक नहीं है अद्भुत पति सप्ताहां	5
16	बुटीक चार्मर ग्रेट लोकेशन वाइफ	3
17	लव्ड इन इन मार्केट कमाल, नॉट	5
18	बेहतर होटल के अनुभव से नहीं पृ	5
19	अच्छी जरूरतों को कम करने के	5
20	अद्भुत स्थान ग्रेट बेड सुंदर कमरे	4
21	टेम्पुर-पेडिक बेड, मिनट का शां	5
22	लवली इन रुकने वाले इन मार्केट	5
23	वर्थ मनी लोकेशन हाल ही में सरा	5
24	नफरत सराय, कमरे-सेवा भयानक	4
25	लव्ड इन हसबैंड ग्रेट रोमांटिक अ	1
26	ऐस नॉट प्लेस हसबैंड इक्का होटल	5
27	भुगतान स्थान पर सही पैदल दूरी,	3
28	ऐस ग्रंज लाइव्स मोल्ड फफूंदी छो	3
29	ऐस होटल, यथोचित रूप से कीम	1
30	निराशाजनक लड़की सप्ताहांत दे	4
31	ऐस होटल विस्मय, सिएटल ऐस हे	2
32	एकदम सही तरीके से, ऐस सिएट	5



```
[2]: rating_counts = df['Rating'].value_counts()
     print(rating_counts)

5    9042
4    6021
3    2179
2    1786
1    1417
Name: Rating, dtype: int64
```



```
Random Forest Classifier:
Validation Accuracy: 0.44
Classification Report:
              precision    recall  f1-score   support

           1       0.00      0.00      0.00       149
           2       0.00      0.00      0.00       179
           3       0.00      0.00      0.00       225
           4       0.31      0.09      0.14       587
           5       0.45      0.93      0.61       904

    accuracy                           0.44      2044
   macro avg       0.15      0.20      0.15      2044
weighted avg       0.29      0.44      0.31      2044

Confusion Matrix (Random Forest):
 [[  0   0   0  19 130]
 [  0   0   0  19 160]
 [  0   0   0  18 207]
 [  0   0   0  54 533]
 [  0   0   0  64 840]]

Naive Bayes Classifier:
Validation Accuracy: 0.44
Classification Report:
              precision    recall  f1-score   support

           1       0.00      0.00      0.00       149
           2       0.50      0.01      0.01       179
           3       0.00      0.00      0.00       225
           4       0.25      0.00      0.01       587
           5       0.44      0.99      0.61       904

    accuracy                           0.44      2044
   macro avg       0.24      0.20      0.13      2044
weighted avg       0.31      0.44      0.27      2044

Confusion Matrix (Naive Bayes):
 [[  0   0   0   0 149]
 [  0   1   0   1 177]
 [  0   0   0   1 224]
 [  0   0   0   2 585]
 [  2   1   0   4 897]]

Multiclass Support Vector Machine (SVM):
Validation Accuracy: 0.43
Classification Report:
              precision    recall  f1-score   support

           1       0.00      0.00      0.00       149
           2       0.00      0.00      0.00       179
           3       0.00      0.00      0.00       225
           4       0.29      0.12      0.17       587
           5       0.45      0.90      0.60       904

    accuracy                           0.43      2044
   macro avg       0.15      0.20      0.15      2044
weighted avg       0.28      0.43      0.31      2044

Confusion Matrix (SVM):
 [[  0   0   0  20 129]
 [  0   0   0  30 149]
 [  0   0   0  30 195]
 [  0   0   0  70 517]
 [  0   0   0  89 815]]
```



```
           model     Score
2    Naive Bayes  0.440313
1  Random Forest  0.437378
0            SVM  0.432975
```

VI. CONCLUSION

In conclusion, the multi-class classification of Hindi text using machine learning techniques presents a significant advancement in the field of natural language processing(NLP). The effectiveness of various machine learning algorithms, from traditional models like Naïve Bayes and Support Vector Machines(SVM) to advanced techniques such as Random Forests, was demonstrated. Feature extraction methods, including TF-IDF and word embeddings (Word2Vec, FastText), played a critical role in enhancing the semantic understanding of the Hindi language. The practical implications of this research are evident in applications such as sentiment analysis, topic categorization, and content recommendation for the Hindi language.

VII. FUTURE WORK

Future research should focus on apable of handling the richness of the Hindi language. Additionally, exploring transfer learning from pre-trained models on larger datasets, potentially in multiple languages, holds promise for further improving classificationperformance.

VIII. REFERENCES

1. Sinha RMK (2009) A journey from Indian scripts processing to Indian language processing. IEEE Ann Hist Comput 31(1):8–31. https://doi.org/10.1109/MAHC.2009.1

2. Mishra G, Nitharwal SL, Kaur S (2010) Languageidentification using Fuzzy-SVM technique. In: 2nd International conference on computing, communication and networking technologies, https://doi.org/10.1109/icccnt.2010.5592553

3. Sreejith C, Indu M, Raj PCR (2013) N-gram basedalgorithm for distinguishing between Hindi and Sanskrit texts. In: 4th International conference on computing, communications and networkingtechnologies. https://doi.org/10.1109/icccnt.2013.6726777

4. Kumar R, Singh P (2017) Bilingual code-mixing inIndian social media texts for Hindi and English. In: Singh D, Raman B, Luhach A, Lingras P (eds) Advanced informatics for computingresearch. communications in computer and information science, vol 712. Springer, Singapore,pp 121–129. https://doi.org/10.1007/978-981-10-5780-9_11

5. Kumar R, Dua M, Jindal S (2014) D-HIRD Domain-independent Hindi language interface to relational database. In: IEEE international conferenceon computation of power, energy, information and communication. IEEE Press, pp 81–86. https://doi.org/10.1109/iccpeic.2014.

6. Prasad G, Fousiya KK (2015) Named entity recognition approaches: a study applied to English and Hindi language. In: International conference oncircuits, power and computing technologies. https://doi.org/10.1109/iccpct.2015.7159443

7. Gupta S, Bhattacharyya P (2010) Think globally, apply locally: using distributional characteristics for

developing culturally sensitive models c

Hindi named entity identification. In: Proceedings of the named entities work-shop. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 116–125

8. Jain A, Yadav D, Tayal DK (2014) NER for Hindi language using association rules. In: International conference on data mining and intelligent computing. . https://doi.org/10.1109/icdmic.2014.6954253