# Machine Learning Models for Hate Speech Identification in Marathi Language

Disha Gajbhiye[1], Swapnil Deshpande[1], Prerna Ghante[1], Abhijeet Kale[1] and Deptii Chaudhari[1]

[1]*Hope Foundation's International Institute of Information Technology, Hinjawadi, Pune*

### Abstract

Hate speech content has become a significant issue in today's world. Hate speech detection is an automated task of detecting textual content that contains discriminatory language regarding a person or group based on who they are, their race, gender, caste, etc. In this paper, we discuss the models submitted by our team, Mind Benders, for Marathi subtask A, for "Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (HASOC)" at Forum for Information Retrieval Evaluation. A training and test dataset in Marathi language containing 1874 and 625 tweets, respectively, were shared by the HASOC organizers. Using these datasets, we propose an approach to automatically classify the tweets into two categories: "NOT" (Non-Hate-Offensive) and "HOF" (Hate and Offensive). The classification models developed are applied to the test dataset. They are experimented with to predict the categories of respective test data.

### Keywords

Logistic Regression, Random Forest Classifier, TF-IDF Vectorizer, Text Classification

## 1. Introduction

The use of social media has increased in recent years. It plays a significant role in forming and shaping views of people on various issues. Users tend to send hateful and offensive messages to a person or community on social media platforms, leading to heated debates.

To make social networking sites a friendly knowledge-sharing environment, there is an acute need for an automated hate speech detection system that will automate making decisions.

Hate speech classifies tweets into two categories, hate speech or non-hate speech. The number of hate and non-hate tweets had to be balanced as the initial stage in developing our model. Our data preprocessing step involved two approaches, Random forest, and Logistic Regression.

Random forest is a supervised learning technique used for both classification and regression problems in ML. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

*Forum for Information Retrieval Evaluation, December 13-17, 2021, India*

dishagajbhiye14@gmail.com (D. Gajbhiye); swapnildeshpande412@gmail.com (S. Deshpande); ghanteperi6@gmail.com (P. Ghante); abhijeetkale459@gmail.com (A. Kale); deptiic@isquareit.edu.in (D. Chaudhari)

0000-0003-3632-3832 (D. Chaudhari)

A machine learning approach called logistic regression is used to forecast the likelihood of a target variable. It's a method for predicting a categorical dependent variable from a set of independent variables.

## 2. Related Work

Several studies on the automatic detection of hate speech and offensive and non-offensive content have been published. Kulkarni, et al.[1] has adopted the best accuracy using IndicBERT and CNN with Indic fastText word embeddings. This dataset will play a crucial role in advancing NLP research for the Marathi language.

Aluru et al.[2] worked on using classification techniques for hate speech detection like CNN-GRU, BERT, mBERT, translation. Pathak et al.[3] applied Support Vector Classifier, Multinomial Bayes, LR, Random Forest Classifier, n-gram model, Text Classification. Founta et al.[4] worked on Deep Learning Architectures such as text classification network, metadata network, combining two classification paths, and trained combined networks. The related study shows that significant work has been done on detecting hate speech in many Indian languages.

The approach of a system developed by Khandelwal et al.[5] is based on N-gram, CBOW, and reference tokens. This system detects abusive language in English from social media. Another work done by Lakshmi BS et al.[6] detects the offensive content from English and Kannada social media text. Sutejo et al.[7] used word n-gram, Long short-term memory (LSTM) in deep learning to determine the sentiments from the Indonesian language. Jiang et al.[8] used two datasets of the Hate speech dataset published on Kaggle that contain 1000 unique labeled values (tweets data). They have used multiple classifiers such as Logistic Regression and Support Vector Machines (SVMs) for classification.

Kovács et al.[9] worked on the text preprocessing methods and the cross-validation method used to train and evaluate models. Working on Natural Language Toolkit, Word2vec, a combination of Bag of-word (CBOW) and Skip-Gram algorithm, was done by Chaitanya et al.[10]. Gaydhani et al.[11] employed several techniques such as SVM, Logistic regression, and Naive Bayes to classify tweets into offensive and non-offensive. Mandl et al.[12] presented an overview of the tasks and the results of the HASOC track at FIRE 2020.

## 3. Problem Definition

We propose a coarse-grained binary classification to classify tweets into two classes: Hate and Offensive (HOF) and Non- Hate and offensive (NOT).

Non-Hate-Offensive (NOT) - Post does not contain any Hate speech, profane, offensive content. Hate and Offensive (HOF) - Post contains Hate, offensive, and profane content.

Best resulting features are used by extracting language-specific and language-independent characteristics of the given dataset. The approach applied for the classification of this text data is explained below.

**Table 1**
Training and Test Dataset Statistics for the Marathi language

| Language used | Type of dataset | Type of tweet | % | Total |
|---|---|---|---|---|
| Marathi | Training | HOF | 1204 (64.27%) | 1874 |
| Marathi | Training | NOT | 670 (35.73%) | |
| Marathi | Test | HOF | Not known | 625 |
| Marathi | Test | NOT | Not known | |

## 3.1. Datasets

We have chosen the task to identify offensive and non-offensive content in the Marathi dataset released for the HASOC shared task as discussed above, consisting of CSV files of comments. All given comments are in Marathi. This training dataset has columns with column names as Text ID, text, and label, respectively. The Label column has values either HOF, indicating offensive text, or NOT, indicating a non-offensive text. The number of comments in the file is around 1874. This training dataset is given to carry the experimental work of training the machine by applying appropriate machine learning algorithms. The test dataset has only two columns, text id, and text. The third column i.e. Label, is missing. After the training in the first phase, the machine learning algorithms have to predict the labels of the respective tweets. Approximately 625 comments are available in this dataset for both languages. Gaikwad et al.[13] worked on a model for Marathi language that described the task's data. Modha et al.[14] have given an overview of the results and findings of HASOC 2021. Table 1 represents the statistical data about this Training and Test Dataset for the Marathi language.

## 3.2. Methodology

A supervised machine learning approach is used in the experimental work. While building the model, data preprocessing is a vital step. In NLP, the first step is to preprocess the data, i.e., removal of unnecessary noise from the textual content. This is followed by encoding the text into numeric vectors as Machine Learning needs data in the numeric form. This is done using encoding techniques such as BagOfWords, n-gram, TF-IDF, Word2Vec, etc. In our analysis, we have implemented the TF-IDF feature extraction technique.

### 3.2.1. Data preprocessing

The data is usually in the natural human format, which is in sentences or paragraphs. Hence, before analyzing it, the information needs to be transformed and cleaned up so that the computer in the desired language can understand it. Following are the steps of the preprocessing data phase:

- Removal of Leading and Trailing spaces: They are unnecessary whitespaces located at both ends of the line, removed using the python strip() method.

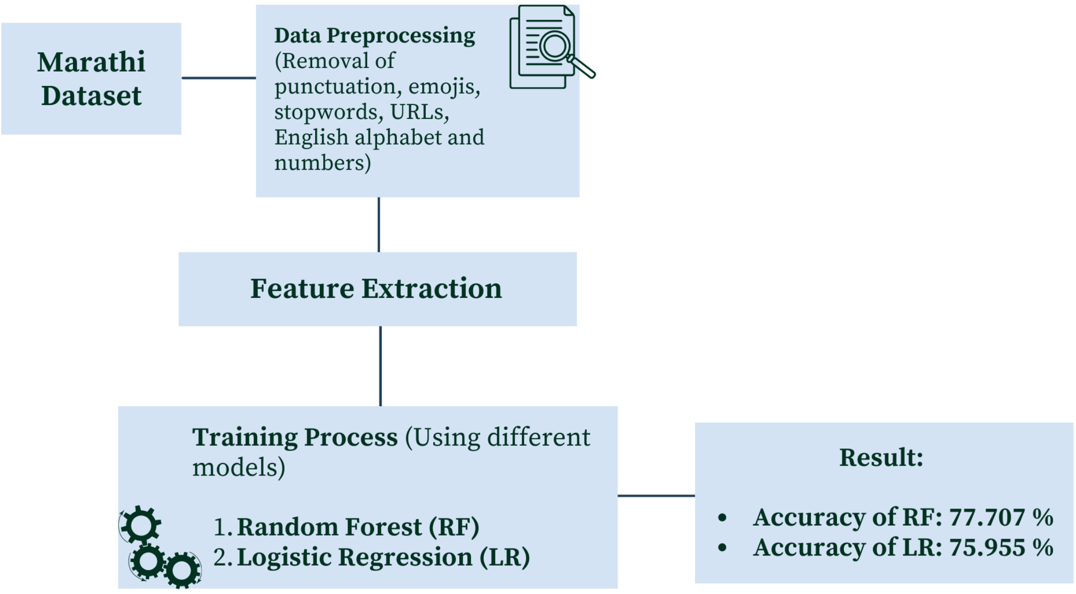
Marathi Dataset
Data Preprocessing (Removal of punctuation, emojis, stopwords, URLs, English alphabet and numbers)
Feature Extraction
Training Process (Using different models)
1. Random Forest (RF)
2. Logistic Regression (LR)
Result:
• Accuracy of RF: 77.707 %
• Accuracy of LR: 75.955 %

**Figure 1:** Architectural View

- Removal of irrelevant characters (numbers and punctuation): In our analysis, the English alphabet and numbers, Marathi numbers, and punctuation are irrelevant. Thus, they are removed to simplify the text content.
- Removal of URLs and emojis: URLs and emojis are also needless in our analysis; hence they are removed from the text using regex expression.
- Removal of stopwords: A custom-made Marathi stopwords list is defined for removing stopwords, which are commonly used words that have no real value in the analysis.

### 3.2.2. Features Extraction

For feature extraction, we applied the TF-IDF technique, which is used to get the most important words. TF and IDF measure the frequency of the word in a document and the uniqueness of the word, respectively. To convert the sentences into vectors, multiply the word frequency by the inverse document frequency. This is done with sci-kit-learn, and the TF-IDF vectorizer technique is used to extract features from the document of words. Thus, it provides a matrix of numeric values of the entire document.

### 3.2.3. Classifier Models

Implementation of two classifier models was carried out, namely, Logistic Regression and Random Forest Classifier. The extracted feature set is used in the training phase. Around 70% of the observations from the training dataset are used for fitting the model. In contrast, the remaining portion is used for testing to make predictions to test the model's accuracy. We have used the accuracy of the results of classification to evaluate the performance of these classifiers. For this purpose, the parametric values with the best performance are found by varying the parameters of each classifier.

## 4. Experimental Work

We have developed a model for the HASOC Marathi subtask A using a Machine Learning approach. Experiments were done on various classifier algorithms by the feature extraction set. The classifier algorithms that we used for our experiments are as follows:

- **Logistic Regression (LR):** As it is known, Linear regression uses a linear function to map input values to continuous values. The data is modeled using a straight line to predict the output of a variable. Logistic regression is similar to linear regression, except logistic regression predicts whether something is true or false instead of predicting continuous values. It is a Supervised Machine Learning algorithm used to predict the probability of target variables. The probability of some obtained event is represented as a linear function of a combination of predictor variables. It is used when data is linearly separable and output is binary or dichotomous in nature. So, it is used for binary classification problems. The target variable is divided into two classes' 1' for success/YES and' 0' for failure/NO. Logistic regression's ability to provide probabilities and classify new samples using continuous and discrete measurements makes it a popular machine learning method. One big difference between linear regression and logistic regression is how the line is fit to the data. With linear regression, we fit the line using the least-squares method, i.e., we find the line that minimizes the sum of the squares of the residuals. We also use the residuals to calculate $R^2$ and to compare simple models to complicated models. Logistic regression doesn't have the same concept of a residual, so it can't use the least-squares method. Instead, it uses the concept of maximum likelihood. The goal of maximum likelihood is to find the optimal way to fit a distribution to the data.
  Instead of fitting a line to the data, logistic regression fits an "S" shaped logistic function called the Sigmoid function, which is used for classification. It is helpful to map any predicted values into values between 0 and 1. The concept of the threshold value is used in LR. If the expected value is above the threshold, it tends to be one, and below the threshold, it is 0. There are two hypotheses in logistic regression: a null hypothesis and the other is an alternative hypothesis. We used an alternative where the model predicts the accurate values and differs significantly from null or zero. From the analysis of this hypothesis, the output from the hypothesis depends on estimated probability.
  $log\frac{p}{1-p}$ is a link function used in logistic regression where p is the probability of success and 1-p is the probability of failure. Here p must always be positive and less than equal to 1. $\frac{p}{1-p}$ is an odds ratio. If the odds ratio comes out positive then the probability of success is always more than 50%. If it comes out negative, then it is the probability of failure.
- **Random Forest (RF):** It is a Machine Learning algorithm used for classification and regression problems. Random forests are made out of decision trees. Decision trees work great with the data used to create them, but they are not flexible when it comes to classifying new samples. Random forests combine the simplicity of decision trees with

flexibility resulting in a vast improvement in accuracy. A random forest contains several decision trees on various subsets of a given training dataset. It provides output based on a majority vote. The decision tree consists of three components, a decision node, a leaf node, and a root node. This tree will divide the training dataset into branches and further separate it into other branches.

The variation between a decision tree and a random forest is that the earlier combines certain decisions while the latter does not. A random forest, on the other hand, combines many decision trees. We have used a bagging method for prediction known as Bootstrap aggregation, the ensemble technique used in random forests. It involves using different samples of data rather than one. The training dataset consists of observation and features that are used for prediction. Now the tree will produce different outputs depending upon training data. The final output obtained is based on majority voting, and the collection of this output is called aggregation.

Also, we used hyper-parameters which are helpful to increase the prediction power of the model. n_estimators is one of the hyper-parameters. n_estimator is many trees that the algorithm builds before taking majority voting or average predictions. If the number of trees increases, the model's performance will improve, and prediction will be stable.

## 5. Results

For better model performance, we have used 70 percent of the training data for training the model, and the remaining is used for testing. Table 2 and Table 3 show the Precision, Recall, F1, and Accuracy scores for Logistic Regression and Random Forest.

Precision is defined as the ratio $\frac{tp}{tp+fp}$ where $tp$ is the number of true positives and $fp$, the number of false positives. Precision is the ability of the classifier not to label a sample as positive, that is negative. The F1 score is also known as balanced F-score or F-measure. It is the weighted average of Precision and Recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of Precision and Recall to the F1 score is equal. The formula for the F1 score is defined as $F1 = \frac{precision \cdot recall}{precision + recall}$. Recall score helps when the cost of false negatives is high. Recall is the ratio $\frac{tp}{tp+fn}$ where $tp$ is the number of true positives and $fn$, the number of false negatives. Recall is the ability of the classifier to find all the positive samples. Accuracy score can tell us immediately whether a model is being trained correctly and how it may perform generally. It is simply a ratio of correctly predicted observations to the total observations.

Logistic Regression has an F1 score of 0.84 for the non-offensive text and 0.54 for the hate-offensive. Accuracy of 0.7595 is obtained from Logistic Regression. Random Forest has an F1 score of 0.83 for the non-offensive text and 0.67 for the hate-offensive text. This classifier gives an accuracy of 0.7770.

## 6. Conclusion

Hate speech continues to be a social media problem. This paper presents the experimental work and results of HASOC Marathi subtask-A by Team Mind Benders. This paper proposes a

**Table 2**
Results of Logistic Regression

| | Precision | Recall | F1 score | Support |
|---|---|---|---|---|
| HOF | 0.85 | 0.39 | 0.54 | 224 |
| NOT | 0.74 | 0.96 | 0.84 | 404 |
| Accuracy | | | 0.76 | 628 |
| Macro avg | 0.80 | 0.68 | 0.69 | 628 |
| Weighted avg | 0.78 | 0.76 | 0.73 | 628 |

Logistic Regression, Accuracy Score: 75.955%

**Table 3**
Results of Random Forest

| | Precision | Recall | F1 score | Support |
|---|---|---|---|---|
| HOF | 0.70 | 0.65 | 0.67 | 224 |
| NOT | 0.81 | 0.85 | 0.83 | 404 |
| Accuracy | | | 0.78 | 628 |
| Macro avg | 0.76 | 0.75 | 0.75 | 628 |
| Weighted avg | 0.77 | 0.78 | 0.77 | 628 |

Random Forest, Accuracy Score: 77.707%

solution for detecting Marathi hate speech and offensive content on the Twitter dataset through supervised machine learning approaches like Logistic Regression (LR) and Random Forest (RF).

We performed an analysis of LR and RF on various sets of feature values and model parameters. For the identification of critical features from data, we used the TF-IDF feature extraction technique. The results showed that Random Forest performs comparatively better than the Logistic Regression approach. We achieved a reasonable accuracy of 0.77 using the Random Forest classifier. Given all the challenges that remain, there is a need for more research on this problem statement.

## References

[1] A. Kulkarni, M. Mandhane, M. Likhitkar, G. Kshirsagar, R. Joshi, L3cubemahasent: A marathi tweet-based sentiment analysis dataset, arXiv preprint arXiv:2103.11408 (2021).

[2] S. S. Aluru, B. Mathew, P. Saha, A. Mukherjee, Deep learning models for multilingual hate speech detection, arXiv preprint arXiv:2004.06465 (2020).

[3] V. Pathak, M. Joshi, P. Joshi, M. Mundada, T. Joshi, Kbcnmujal@ hasoc-dravidian-codemix-fire2020: Using machine learning for detection of hate speech and offensive code-mixed social media text, arXiv preprint arXiv:2102.09866 (2021).

[4] A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, I. Leontiadis, A unified deep learning architecture for abuse detection, in: Proceedings of the 10th ACM conference on web science, 2019, pp. 105–114.

[5] A. Khandelwal, S. Swami, S. S. Akhtar, M. Shrivastava, Gender prediction in english-hindi

code-mixed social media content: Corpus and baseline system, Computación y Sistemas 22 (2018) 1241–1247.

[6] B. S. Lakshmi, B. Shambhavi, An automatic language identification system for code-mixed english-kannada social media text, in: 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), IEEE, 2017, pp. 1–5.

[7] T. L. Sutejo, D. P. Lestari, Indonesia hate speech detection using deep learning, in: 2018 International Conference on Asian Language Processing (IALP), IEEE, 2018, pp. 39–43.

[8] L. Jiang, Y. Suzuki, Detecting hate speech from tweets for sentiment analysis, in: 2019 6th International Conference on Systems and Informatics (ICSAI), IEEE, 2019, pp. 671–676.

[9] G. Kovács, P. Alonso, R. Saini, Challenges of hate speech detection in social media, SN Computer Science 2 (2021) 1–15.

[10] I. Chaitanya, I. Madapakula, S. K. Gupta, S. Thara, Word level language identification in code-mixed data using word embedding methods for indian languages, in: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2018, pp. 1137–1141.

[11] A. Gaydhani, V. Doma, S. Kendre, L. Bhagwat, Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach, arXiv preprint arXiv:1809.08651 (2018).

[12] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german, in: Forum for Information Retrieval Evaluation, 2020, pp. 29–32.

[13] S. Gaikwad, T. Ranasinghe, M. Zampieri, C. M. Homan, Cross-lingual offensive language identification for low resource languages: The case of marathi, in: Proceedings of RANLP, 2021.

[14] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in english and indo-aryan languages and conversational hate speech, in: FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, 13th-17th December 2021, ACM, 2021.