Toxic Words Detection in Online Platforms Using Machine Learning

Dr. M. Ayyavaraiah[1]; D. Sreenath[2]

[1]Associate Professor, Department of Computer Science Engineering, Rajeev Gandhi Memorial College of Engineering & Technology, Andhra Pradesh, India.
[2]M. Tech Student, Department of Computer Science Engineering, Rajeev Gandhi Memorial College of Engineering & Technology, Andhra Pradesh, India.

Publication Date: 2025/06/13

Abstract: Harmful comments are insulting, aggressive, or irrational and can interfere with online discussions and frequently cause participants to disengage. The widespread issue of cyberbullying and digital harassment undermines open communication by deterring people from expressing opposing perspectives. Numerous websites encounter difficulties sustaining constructive conversations, prompting some to limit or completely remove commenting. This research intends to investigate the prevalence of online abuse and categorize user input through annotated data to effectively recognize toxicity. To tackle this challenge, we will implement numerous Natural Language Processing (NLP) techniques to handle text categorization, assessing their outcomes to identify the most efficient approach for toxic comment identification. Numerous machine learning methods, including SVM, logistic regression, decision tree and deep Learning Techniques, are used to group the abusive words. Our objective is to attain high precision in detecting toxic behaviour, thus motivating organizations to adopt measures that reduce its negative consequences.

Keywords: *Harmful Comments, Machine Learning, Toxicity, Insulting, Communication, Deep Learning, Logistic Regression and Naive Bayes.*

How to Site: Dr. M. Ayyavaraiah; D. Sreenath; (2025) Toxic Words Detection in Online Platforms Using Machine Learning. *International Journal of Innovative Science and Research Technology,* 10(6), 308-313.
https://doi.org/10.38124/ijisrt/25jun516

I. INTRODUCTION

The unforeseen growth of the internet has significantly changed the way druggies communicate, interact, and exchange information encyclopaedically. Social networking Apps and Video blogs allow druggies to engage in real time and perform several conditionings. This has led to a rise in dangerous words and unhappy commentary that include hate speech, oppression, and other types of virtual incivility.

Toxin in virtual speech not only affects druggies but also erodes the quality of news, affects structure in exchanges, and encourages aggressive speech. The need for methodical change is less than ever because mass-stoner-generated matter is uploaded every alternate day to the media. Traditional alteration methods, such as reviewing and filtering, are ineffective due to their lack of accuracy and contextual understanding.

Machine learning takes a favourable approach to discuss this provocation. By using large datasets in models using six algorithms, it can identify the harmful and harmless words. Using KNN, SVM, Logistic Regression, Random Forest, and Decision Tree can classify datasets and categorise the words according to them. It also works on natural language processing (NLP), like transformer-based systems like BERT, to perform tasks in a modernised learning way to identify foul words. It is a key part of the process, allowing dataset to enrich the model and its context of comments rather than depending on traditional features.

This paper inquires into the use of Machine Learning to group the harmful words and process the six algorithm techniques to improve the classification and precision. Comparing different approaches of Deep Learning and ML makes more safer online platforms and user content.

- *Objectives*

The core aim of this study was to perform accurate values and classify the toxic comments using six algorithms of machine learning. this requires numerous machine learning techniques to detect the most harmful words. This project aimed to rate the performance of algorithms based on user applications. With the rapid prevalence of hate speech, abusive words, and threatening language in online spaces, there is a need to detect this harmful content and promote secure online interactions.

II. LITERATURE REVIEW

Offensive commentary on social media platforms which is increasingly common, has sparked major conflicts between individuals and groups. Commenting toxically not only generates verbal violence but and often conveys offensive, impolite, or socially harmful communication patterns that discourage further participation in discussions. Therefore, identifying offensive comments on social platforms should be treated as an important task to Keep its operations free from disruptions and animosity. As a result, a diverse range of methodologies for identifying toxic comments have been proposed. These methodologies are evaluated based on three principal criteria: classification effectiveness, feature dimension reduction, and feature importance assessment.

In recent years research on toxic and offensive comment classification has grown significantly because of the rise in concerns about online hate speech. For late speech classification Davidson et al. (2017), created a labelled dataset of tweets and deployed support vector machines. Their study also emphasized the subjectivity and bias inherent in human annotation, which has remained a critical challenge in subsequent analyses. Warner and Hirschberg (2012) explored hate speech detection by avoiding simple keyword filters. While achieving modest performance (F1 score of 0.63), their study was significant for conceptualizing hate speech in algorithmic terms.

Georgeakopoulos et al. (2018) evaluated convolutional neural networks (CNNs) with bag-of-words models, highlighting that deep learning models are superior to conventional techniques in capturing contextual meaning in texts. Studies by Schmidt and Wiegand (2017) and Ross et al. (2017) investigated the subjectivity of annotations. They underscored the need for uniform annotation methods, as viewpoints of toxicity vary significantly between individuals. Moreover, Pioneering research in machine learning models such as LSTM (Hoch Reiter & Schmidhuber, 1997), GRU (Cho et al., 2014), and Transformers (Vaswani et al., 2017) inform this thesis's strategy. Pre-developed models via Hugging face Transformers (Wolf et al., 2019) offered flexible resources for customizing the Jigsaw dataset (CJ Adams et al., 2017), the core dataset used here.

III. METHODOLOGY

To implement the toxic grouping system, the project plans a system architecture that contains six individual modules to run the data pipeline of the project.

- ➢ *Upload Dataset:*

Allow users to upload the content data of virtual words for testing and using in Machine learning models.

- ➢ *Process Data:*

Scans and use the raw words text by making punctuation and symbols and converting text to lowercase and eliminate the stop words.

- ➢ *Data visualization:*

It provides graphical data from datasets, different frequencies, and label variance using Matplotlib and Seaborn, etc.

- ➢ *Run ML Algorithms:*

It makes data into training and testing subsets and applies several Deep learning Methods and Machine Learning methods. It evaluates and differentiates the prognosis of precision of the set.

- ➢ *Accuracy Graph:*

It compares accuracy, hamming loss, and Log loss for all inserted ML algorithms to analyse the best outcome.

- ➢ *Predict toxic comments from test data:*

Using methods, we can assume harmful comments and classifies them, and user is allowed to upload new harmful words to prediction.

IV. SYSTEM ARCHITECTURE

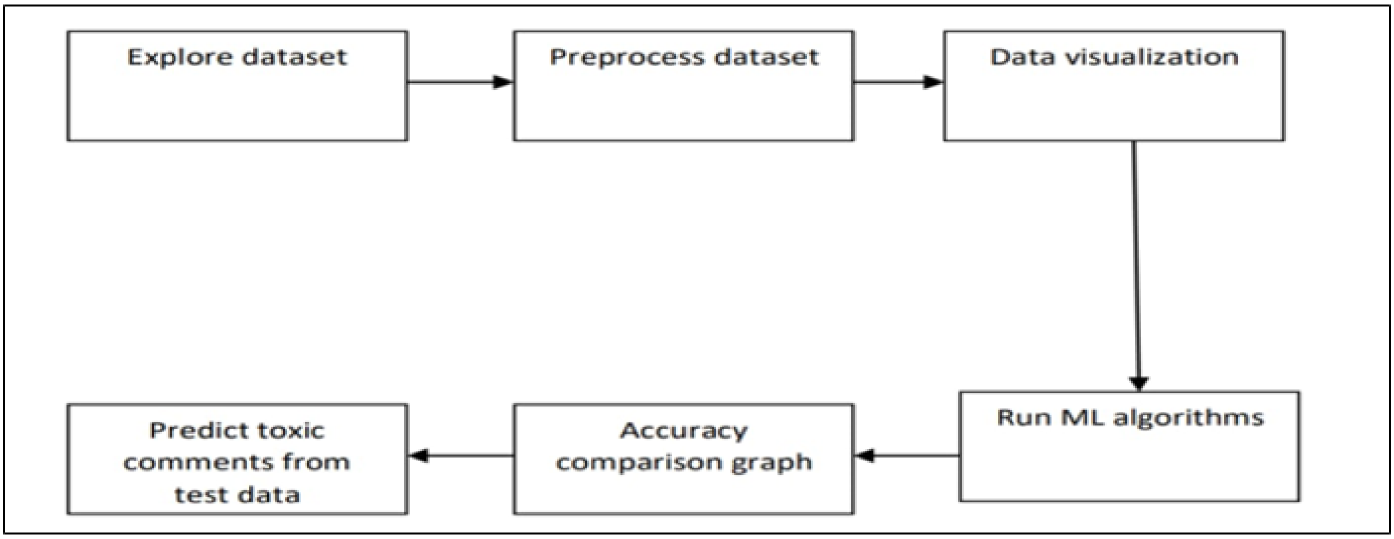


Fig 1 System Architecture

V. FLOW CHART

The flowchart describes the complete key-to-key process of the system classifier in machine learning. It begins with importing packages of necessary conditions, followed by data exploration using sets of harmful words. After that, the data processing can be visualized based on groups created. The text processing ways, like stemming and tokenization, are applied and lead to storeing the data for future access and deeper insight into words.

Next, the selection of words is carried out, and data gets split into training data, where it applies six machine learning techinques to classify or group the harassing and abusive words. The final part involves taking the best -trained model using F1 score and accuracy. The intake of words is tokenized and vectorized , and assumptions are made upon these. After that output is shown, making the workflow to end.

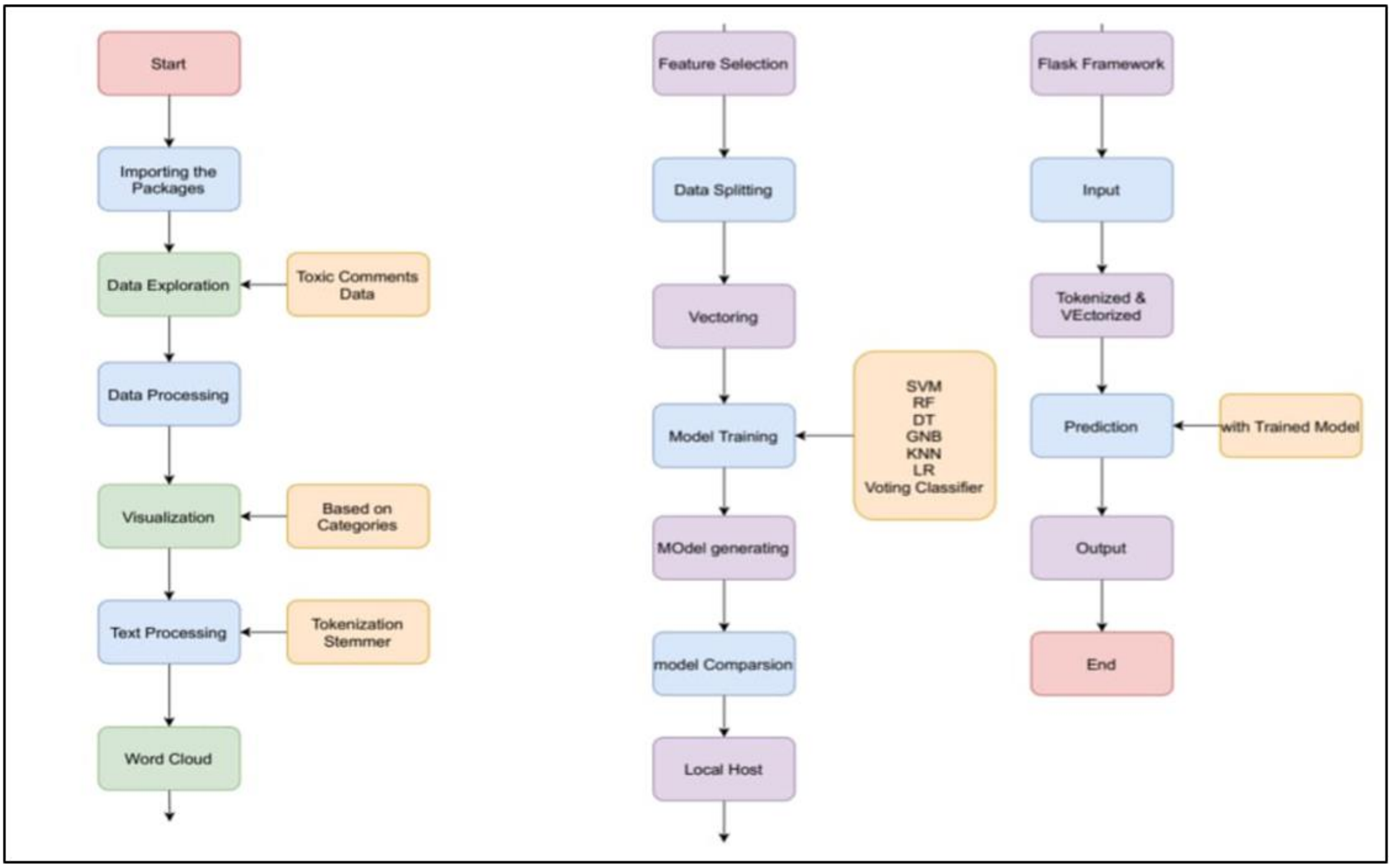


Fig 2 System Architecture Flowchart for Toxic Word Detection Using Machine Learning

VI. IMPLEMENTATION

To run this project, we are using six different ML learning techniques such as Support Vector Machines, Naive Bayes, Random Forest, KNN, Decision Tree, and Logistic Regression, and then calculating their production in terms of accuracy and loss. The more accurate and the less loss will show the better ML prediction algorithm.

VII. ALGORITHMS

- *SVM:*

Support Vector Machine or SVM, is a popular Machine learning method, which is applied for grouping and regressions queries. The main objective of SVM is to create the most appropriate decision that can differs a dataset into toxic and harmless using a boundary called a hyper-plane and put a new dataset separately in the plane.

- *KNN:*

The K-Nearest Neighbour algorithm proposes to make grouping or choices by using proximity about the categorising of an individual dataset. With the help of KNN, we can use Speech Recognition, Image detection and word classifiers.

- *Decision Tree:*

As a supervised learning method, decision trees can handle both categorizing and regression works. It has a typical tree structure and a type of supervised learning which splits data into small sets based on input by creating a tree-like model of decisions.

- *Logistic Regression:*

It is a fundamental ML method applied to assume the chance of assured data influenced by variables and values. It is evaluated using a loss function.

$$L=-1/N\sum [y\log(p)+(1-y)\log(1-p)]$$

- ➢ *Naive Bayes:*

Naive Bayes is one of the easiest and most useful grouping algorithms that helps in constructing speed learning models that can make swift decisions. It works on large-scale datasets with high valued accuracy and speed. It works on the Bayes principle known as conditional probability and classifies works on spam detection and sentiment analysis.

- ➢ *Random Forest:*

Random Forest is a powerful and well-known supervised learning algorithm used to categorizing and returning tasks. It has more accuracy, speed and adaptable and produce stable and reliable outputs. It handles large datasets with more scope and categorize numerical data to process exact predictions.

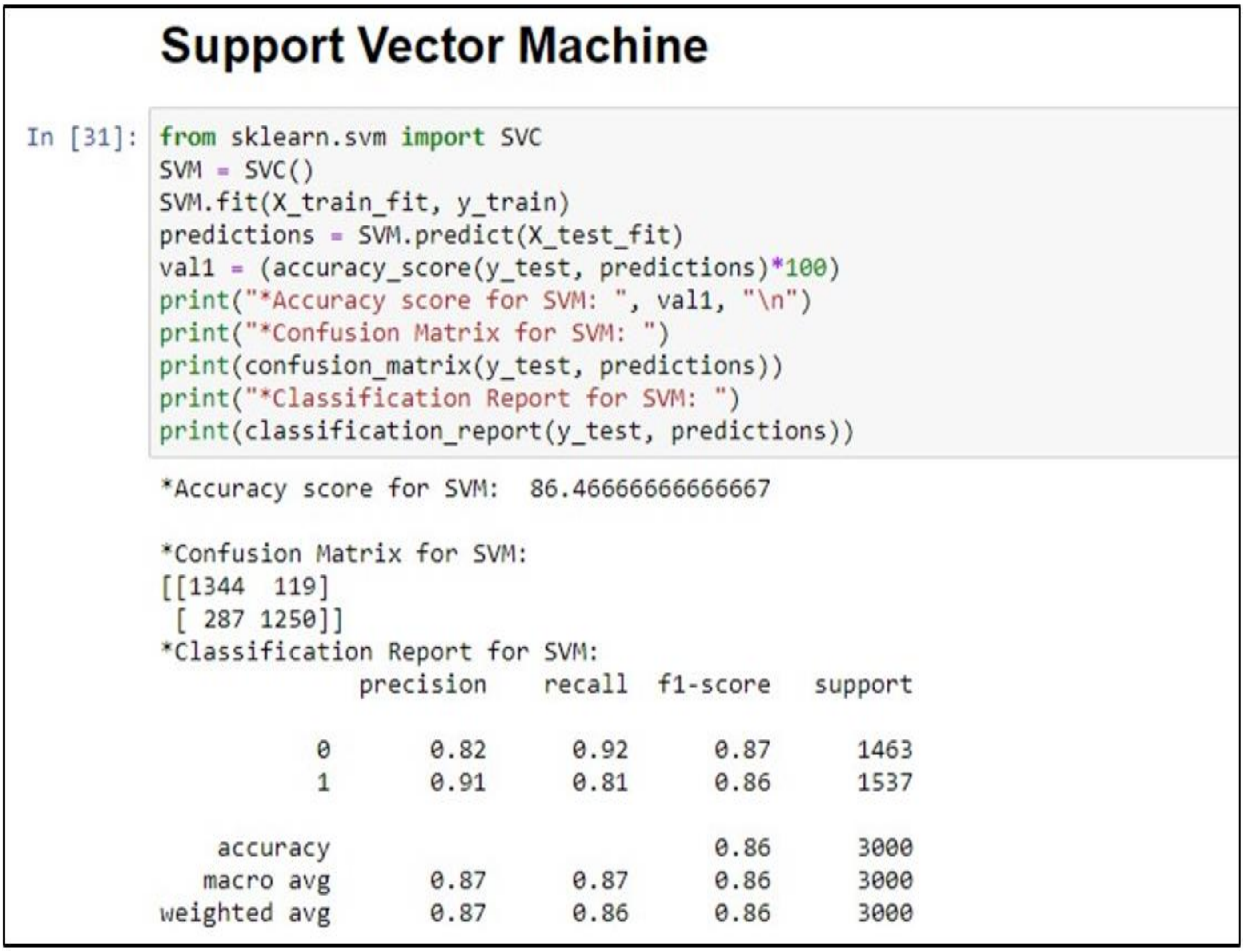
In this project, we implement a machine learning-based approach to detect toxic comments in online platforms. The primary goal is to classify user comments as either toxic or non-toxic, thereby aiding in the moderation of harmful content on websites and social media. The implementation begins with the collection of a labelled dataset that includes various comments along with corresponding labels—1 for toxic and 0 for non-toxic.

Publicly available datasets such as the Jigsaw Toxic Comment dataset or a custom CSV file can be used. The next step involves preprocessing the text data, which includes converting all text to lowercase, removing special characters, and stop words, and normalizing the spacing. Key evaluation metrics such as accuracy, precision, recall, and F1-score are used to assess the model’s ability to correctly classify comments. Finally, the trained model can be used to predict the toxicity of new comments entered by users. This implementation demonstrates that machine learning can be effectively applied to detect toxic content, providing a valuable tool for maintaining safe and respectful online environments.

- ➢ *Outcomes*

- *Upload toxic words to the Dataset and run the code According to it.*

To implement this proposal, we are importing python packages and using KAGGLE TOXIC Comments data which contains words and class label as average or harm contents.



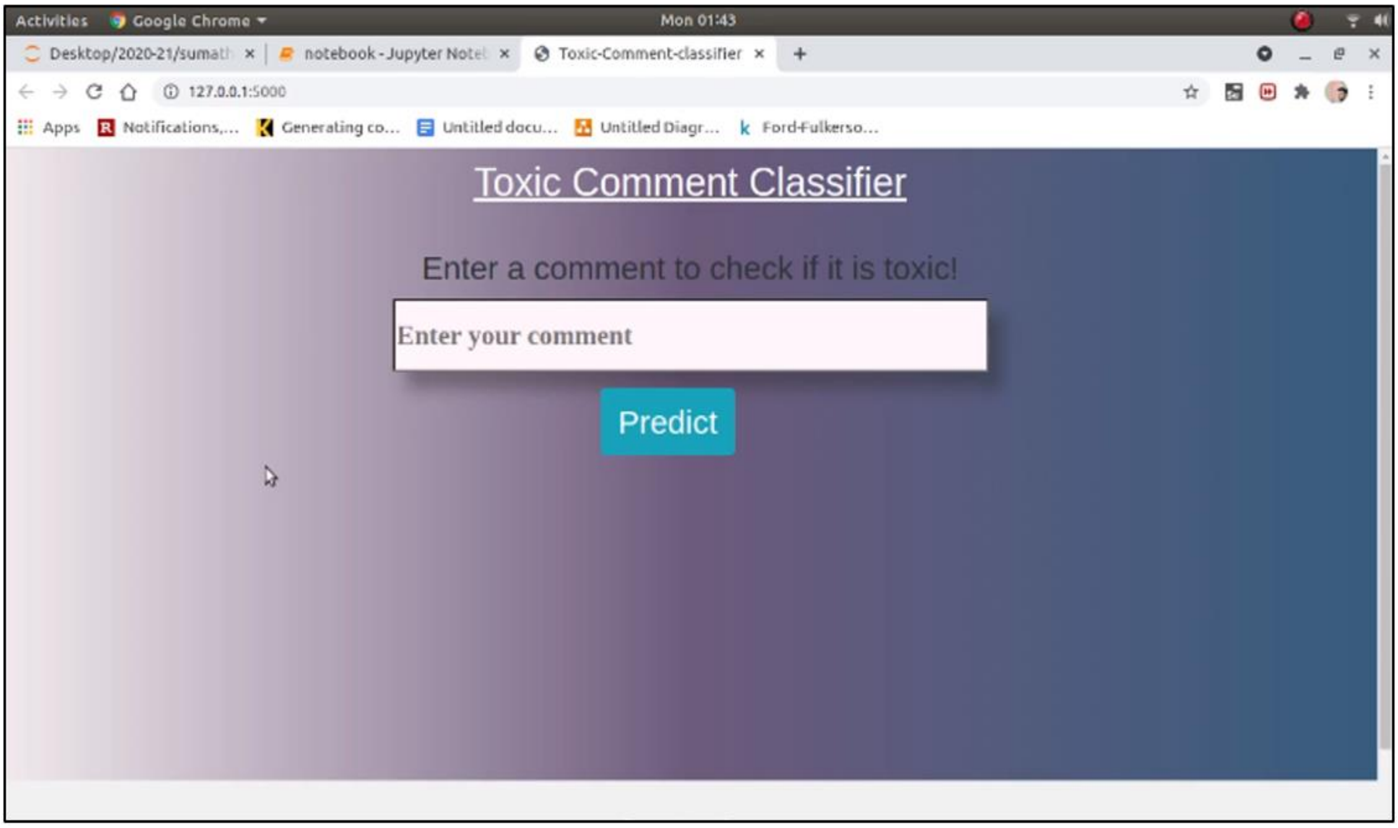
Support Vector Machine

```
In [31]: from sklearn.svm import SVC
         SVM = SVC()
         SVM.fit(X_train_fit, y_train)
         predictions = SVM.predict(X_test_fit)
         val1 = (accuracy_score(y_test, predictions)*100)
         print("*Accuracy score for SVM: ", val1, "\n")
         print("*Confusion Matrix for SVM: ")
         print(confusion_matrix(y_test, predictions))
         print("*Classification Report for SVM: ")
         print(classification_report(y_test, predictions))

         *Accuracy score for SVM:  86.46666666666667

         *Confusion Matrix for SVM:
         [[1344  119]
          [ 287 1250]]
         *Classification Report for SVM:
                       precision    recall  f1-score   support

                    0       0.82      0.92      0.87      1463
                    1       0.91      0.81      0.86      1537

             accuracy                           0.86      3000
            macro avg       0.87      0.87      0.86      3000
         weighted avg       0.87      0.86      0.86      3000
```

Fig 3 Importing packages and Kaggle dataset in SVM classifier

- *Upload Comments to check if it is Abusive or not.*

  Enter a word to check it is harm or not by using classifiers and algorithms.



Toxic Comment Classifier

Enter a comment to check if it is toxic!

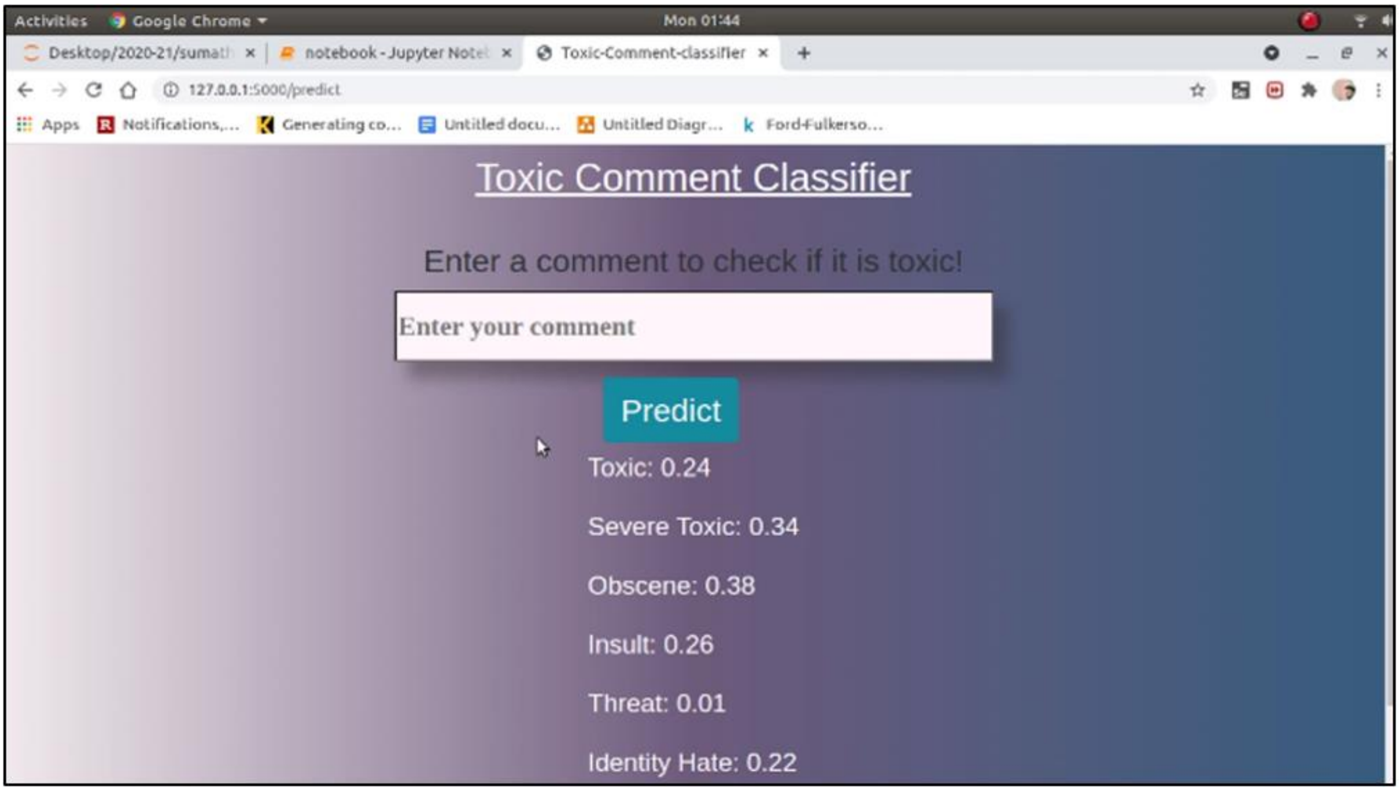
Enter your comment

Predict

Fig 4 Enter Comments to check the toxic range

- *Result of word frequency in toxic Range*

  By entering a word, the frequency can be calculated and displays the toxic range with machine learning algorithms.



Toxic Comment Classifier

Enter a comment to check if it is toxic!

Enter your comment

Predict

Toxic: 0.24

Severe Toxic: 0.34

Obscene: 0.38

Insult: 0.26

Threat: 0.01

Identity Hate: 0.22

Fig 5 Toxic range Calculation

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun516

- *Table of six algorithm classifiers*

Various algorithms are used to check the toxic range in words, and when a word is entered, the frequency can be calculated and displayed in a table, showing the toxic range.

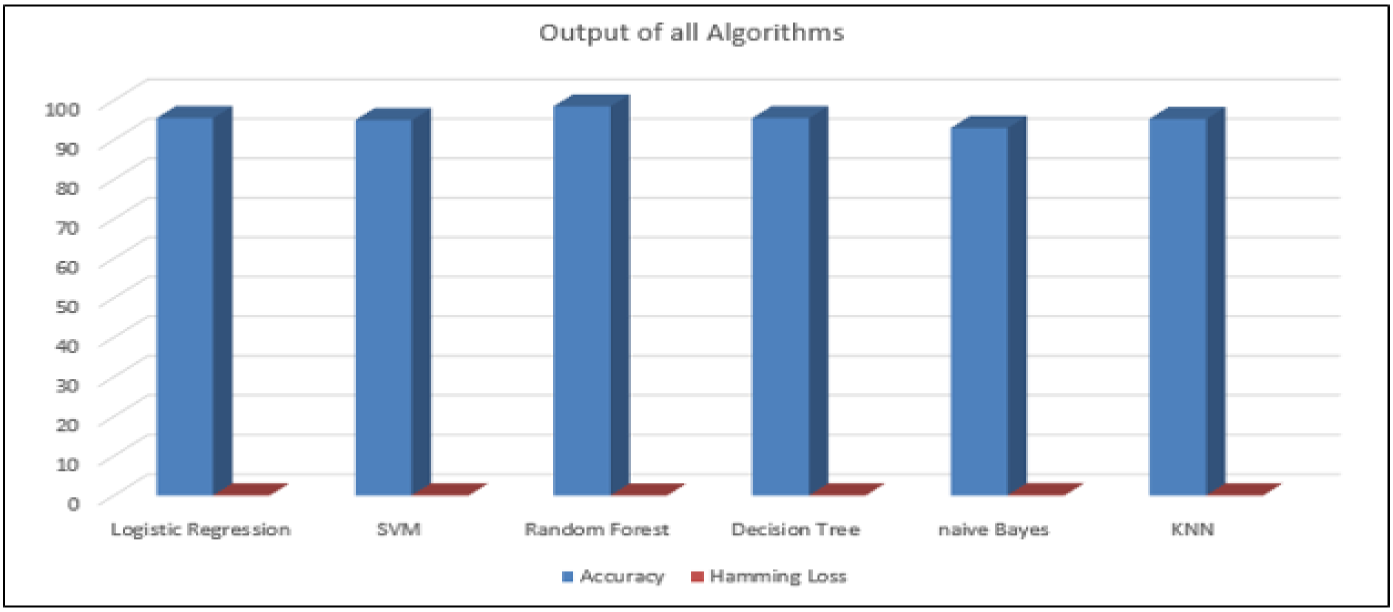


Fig 6 Accuracy and Hamming Loss Comparison of ML Algorithms

VIII. CONCLUSION

As the magnitude of online interactions continues to grow, the toxic and abusive language in media platforms are rapidly increasing. Machine learning, particularly works in natural processing language, has more need for this project to detect the harmful words.

This project proposes six machine learning methods such as Naive Bayes, random forest, KNN grouping, decision tree, support vector machine, and logistic regression, and contrasts their Hamming loss, precision and loss of log. After proper study, logistic regression handles advanced than Hamming loss, and accuracy is better in logistic regression and using log loss. Random Forest works better than others. It also proposes a route for grouping by using deep learning models like BERT and RNNs, which perform in terms of accuracy and return values. SMOTE and class-labelled loss function methods are used to group the polarity and multi-label datasets to identify the abuse across word categories.

The project focus on accuracy and hamming loss. We achieved the highest precision at a rate of 87.32%, and the least feasible Hamming loss is 2.51% of the model. By this we opt for the logistic regression method as our work for data analysis.

REFERENCES

[1]. H. M. Saleem, K. P. Dillon, S. Benesch, and D. Ruths, "A Web of Hate: Tackling Hateful Speech in Online Social Spaces," 2017, [Online]. Available: http://arxiv.org/abs/1709.10159.

[2]. M. Duggan, "Online harassment 2017," Pew Res., pp. 1–85, 2017, doi: 202.419.4372.

[3]. M. A. Walker, P. Anand, J. E. F. Tree, R. Abbott, and J. King, "A corpus for research on deliberation and debate," Proc. 8th Int. Conf. Lang. Resour. Eval. Lr. 2012, pp. 812–817, 2012.

[4]. J. Cheng, C. Danescu-Niculescu-Mizel, and J. Leskovec, "Antisocial behaviour in online discussion communities," Proc. 9th Int. Conf. Web Soc. Media, ICWSM 2015, pp. 61–70, 2015.

[5]. B. Mathew et al., "Thou shalt not hate: Countering online hate speech," Proc. 13th Int. Conf. Web Soc. Media, ICWSM 2019, no. August, pp. 369–380, 2019.

[6]. C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," 25th Int. World Wide Web Conf. WWW 2016, pp. 145–153, 2016, Doi: 10.1145/2872427.2883062.

[7]. E. K. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text Classification Using Machine Learning Techniques," no. August, 2005.

[8]. M. R. Murty, J. V. . Murthy, and P. Reddy P.V.G.D, " Text Document Classification based on Least Square Support Vector Machines with Singular Value Decomposition," Int. J. Comput. Appl., vol. 27, no. 7, pp. 21–26, 2011, doi: 10.5120/3312-4540.

[9]. E. Wulczyn, N. Thain, and L. Dixon, "Ex machina: Personal attacks seen at scale," 26th Int. World Wide Web Conf. WWW 2017, pp. 1391– 1399, 2017, doi: 10.1145/3038912.3052591.

[10]. H. Hosseini, S. Kannan, B. Zhang, and R. Poovendran, "Deceiving Google's Perspective API Built for Detecting Toxic Comments," 2017, [Online]. Available: http://arxiv.org/abs/1702.08138.