Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

Cyber-Bullying Detection in Hinglish Languages Using Machine Learning

Karan Shah
Electronics Engineering
KJ Somaiya College of EngineeringMumbai, India

Chaitanya Phadtare
Electronics Engineering
KJ Somaiya College of EngineeringMumbai, India

Keval Rajpara
Electronics Engineering
KJ Somaiya College of EngineeringMumbai, India

***Abstract*—Cyberbullying is one of the most recent evils of social media. With a boom in the usage of social media, the freedom of expression is being exploited. Statistics show that overall 36.5 % people think they have been cyberbullied in their lifetime. These numbers are more than double of what they were in 2007, and there is an increase from 2018-19, suggesting we are heading in the wrong direction. Solutions to curtail this issue to a certain extent have already been deployed in the market. However, they possess limitations of usage, or simply do not use efficient algorithms. The main goal of this project is to investigate fundamentally new approaches to understand and automatically detect incidents of cyberbullying over tweets, comments, and messages on various social media network. To this end, we have collected a real time twitter data consisting of headlines, comments and trending post's text messages, and designed a labeling study for cyberbullying. An analysis of the labeled data is then presented, including a study of correlations between different features and cyberbullying as well as cyberaggression. This project aims at identifying cyberbullying at is origin, that is when it is being drafted in real-time. Using Machine Learning and with the support of Natural Language Processing(NLP), better performance of cyberbullying detection is obtained.**

***Index Terms*—Cyberbullying, Machine Learning, Natural language processing, Hinglish Languages, Social media.**

I. INTRODUCTION

The internet is the world's largest platform for communicating, connecting as well as sharing ideas, content, photos, videos, views, and daily updates globally. Social media is multifarious, which makes it extensive and interactive. Twitter, YouTube, Instagram, Linked In, Facebook and Whatsapp are some of the largest platforms. In today's date, almost every person uses social media. According to statistics, in **2020**, nearly **3.6** billion users used social media networking sites, and that number is estimated to become **4.41** billion by **2025**. According to Backlink, **58.11%** of the world population using social media. [1] Although social media has many benefits, it has certain negative aspects. Social media is being misused by some People are being harassed on the basis of caste, color, creed, gender, culture, orientation, and background. Various contents and ideas are neglected and criticized harshly. With so many people criticising and harassing others, bullying is on the rise. This bullying through social media is termed as cyberbullying. It causes damage to the reputation and image of the victim. Some people being bullied fall under depression and inflict self-harm. A few people also commit suicide. Therefore, Cyberbullying is a critical issue which needs to be resolved considering the amount of damage it can cause. Thus, Cyberbullying is a severe problem that needs to be taken care of considering the severity of damage it causes to an individual's mental well-being.

Recent studies show that **36.5%** of people experienced Cyberbullying in their lifetime. **60%** of teenagers have experienced some sort of Cyberbullying. **87%** of young online users have accepted that they are witnesses of some kind of Cyberbullying occurring online. [2] Girls are more likely to become a victim of Cyberbullying as compared to boys.Overall, **36%** of girls have reported being cyberbullied compared to **26%** of boys. [3] The amount of Cyberbullying that now takes place has caused health issues for those targeted. **64%** of Cyberbullying victims are more likely to cause adolescents,depression, anxiety, self-esteem, emotional distress,mental, and behavioral problems.

Various solutions in the form of third-party applications have been deployed but the problem with these applications is that they are based on a simple keyword matching technique which gives less accurate results. Manually adding data in the database could be a subjective point of view of the creator and majority of the drawbacks include lack of labeled database or using a biased data set. Few implementations include lexicons, which is a common way to use databases for the detection of abusive words. However, it limits the scope of the application and disregards the statements which may not use abusive words but have hurtful meanings.

Our proposed idea contributes to solving the problem by identifying and classifying text or messages of an intimidating or threatening nature. Our aim is to build a model to classify or identify cyberbullying in English and Hinglish languages and build a Chat application which can predict whether the text entered in group chats is bullying or non-bullying. Also any hateful, offensive words phrases can prevent the devastating after-effects of cyberbullying by addressing the problem at its root by notifying the user which

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

creates an awareness in the society. It proposes the use of Text Mining to provide a method for feature extraction and Machine Learning for performing classification of the text i.e. whether they have hateful words or meanings. It would also increase the accuracy of our models which were not embedded in the previous models.

Therefore, this paper contributes to solving the problem by developing an efficacious technique which can detect abusive and offensive messages by integrating Machine Learning and Natural Language Processing to develop a model that can detect offensive or hateful words in English and Hinglish language. This study is of practical importance and may serve as a reference for future researchers in the domain of cyberbullying detection.

II. Literature Survey

Cynthia Van Hee , Gilles Jacobs [4], proposed a model for automatic cyberbullying detection in social media text by modelling posts written by bullies, victims by standards of online bullying. two corpora were constructed by collecting data from social networking sites like Ask. fm. developed a model using tokenization, PoS-tagging, and lemmatization for pre-processing. Models were developed for English and Dutch to test for language conversion and subsequent accuracy. ML algorithm SVM gave the accuracy for the English language - **64%** and for the Dutch language - **61%**.

Mohammed Ali Al-Garadi, et al. [5], implemented a model to reduce textual cyberbullying because it has become the dominant aggressive behaviour in social media sites.They extracted data from Wikipedia, you- tube Twitter, Instagram and developed a model using tokenization lemmatization and N-gram was used up to 5 levels to calculate TF IDF and count vector for pre-processing. They gave a comparative analysis of ML algorithms using SVM , K clustering, Random forest, Decision Trees and concluded that SVM worked best amongst the four machine learning models. Kshitiz Sahay,et al. [6], Their focus was to identify and classify bullying in the text by analyzing and studying the properties of bullies and aggressors and what features distinguish them from regular users. The dataset they used was obtained from Wikipedia, YouTube, Twitter. In preprocessing removed URL and tags from dataset and performed Count Vectors and TF- IDF vectors. For classification they used Logistic Regression, SVM, Random Forest and Gradient Boosting.

Michele Di Capua [7], implemented a model inspired by Growing Hierarchical SOMs, which are able to efficiently cluster documents containing bully traces, built upon semantic and syntactic features of textual sentences. They followed an Unsupervised approach with Syntactic, Semantic, Sentiment analysis. In Pre-processing stop word removal, punctuation removal was done to generate word clusters. Social features were extracted. Convolutional neural networks were applied using Kohonen map (or GHSOM).Homa Hosseinmardi, et al. [8], They proposed a model to automatically detect cyberbullying text in Instagram by modelling posts written by bullies. developed a system for deciding posts based on shortlisting words of caption. The paper suggested using image processing on Instagram posts for deciding emotional response or test response in case of text pictures. Vijay Banerjee Jui Telavane et al. [9] developed the cyberbullying detection model using Convolution Neural Network and compared the accuracy with previous models. They used the twitter dataset which consists of **69874** tweets which converted to vectors. The accuracy of this model was **93.97%** which was greater than other models.

Noviantho, S. M. Isa and L. Ashianti [3] created a classification model for cyberbullying using Naive Bayes method and Support Vector Machine (SVM).The dataset they used was collected from Kaggle which provides **1600** conversations in Formspring.me in which question and answer are used as labels. This consists of **12729** data of which **11661** data is labeled non-cyberbullying and **1068** is labeled cyberbullying.In data cleaning they removed the words like ‘haha’, ‘hehe’ , ‘umm’ etc. For balancing dataset they formed classification: 2 classes(cyberbullying and non -cyberbullying), 4 classes (non-cyberbullying, cyberbullying with low,middle and high severity level), 11 classes (non-cyberbullying, cyberbullying with 1-10 severity level). In preprocessing they used tokenizations, Transfer case, stop word removal, filter token, stemming, and generating n-gram. For classification they used Naive Bayes and SVM with linear,poly and sigmoid kernels. The SVM kernel with poly kernel gave most average accuracy **97.11%**.

H. Watanabe, M. Bouazizi and T. Ohtsuki [10] their aim was to detect hate speech on Twitter. Their technique is based on unigram and patterns that are automatically collected from the dataset. Their aim was to classify tweets as clean, offensive and hateful. They used 3 types of datasets the first dataset was from crowdflower contains **14000** tweets are classified into clean,offensive and hateful; second was also from crowdflower tweets classified into offensive,hateful and neither; third dataset was from github in which tweets were classified into sexism, racism and neither. They combined 3 datasets to make a bigger dataset. In preprocessing removed URL and tags from tweets also they did tokenization, Part of Speech Tagging, and lemmatization.They used binary classification and ternary classification to identify sentiment-based features, semantic features, Unigram features and pattern feature.Their proposed model gave accuracy of **87.4%** for binary classification to classify tweets into offensive and non-offensive and **78.4%** for ternary classification to classify tweets into hateful, offensive and clean.

J. Yadav, D. Kumar and D. Chauhan [11] developed a model to classify cyberbullying using a pre-trained BERT model. BERT model is a recently developed learning

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

model by Google researchers. In this they use publicly available Formspring (a QA forum) and Wikipedia talk pages (collaborative knowledge repository) datasets and both datasets were manually labelled and also pre-processed . The Formspring dataset contains **12773** question-answer pair comments of which **776** are bully posts and Wikipedia dataset contains **115864** discussion comments which are manually annotated by ten persons of which **13590** comments which are labelled as bully. Their model gave accuracy of **94%** for formspring dataset which is oversampled 3 times and **81%** for wikipedia dataset.

S.E.Vishwapriya, Ajay Gour et al. [12], implemented a model for detecting hate speech and offensive language on twitter using machine learning. Datasets were taken from crowd flower and GitHub. Crowd flower dataset had tweets with labels Hateful, offensive and clean whereas GitHub dataset had columns tweet id and class such as sexism, racism and neither. Tweets were fetched by the tweet id using twitter API. These datasets were then combined. Tweets were converted to lowercase and Space Pattern, URLs, Twitter Mentions, Retweet Symbols and Stop words were removed. To reduce inflectional forms of words, stemming was applied. The dataset was then split into **70%** training and **30%** test samples. N-gram features from the tweets were extracted and were weighed according to their TF IDF values. Unigram, Bigram and Trigram features along with L1 and L2 normalization of TF IDF were considered. Logistic Regression, Na¨ıve Bayes and Support vector machine algorithms were compared. **95%** accuracy was obtained using Logistic Regression with L2 Normalization and n=3.

Lida Ketsbaia, Biju Issac et al. [13], proposed a model To detect hateful and offensive tweets using Machine Learning and Deep Learning. A data set created by the University of Maryland and Cornell University of about **35000** and **24000** tweets respectively was used with tweets labelled as Hate and Non hate. Tweets were converted into lowercase, numbers, URLs and user mentions, punctuation's, special characters and stop words were removed and contradictions were replaced. Data set was then balanced. Logistic Regression, Linear SVC, Multinomial and Bernoulli classifiers were applied in unigrams, bigrams and trigrams. Word2Vec technique was used to improve accuracy. Accuracy of **95%** and **96%** was achieved for the datasets.

Sindhu Abro, Sarang Shaikh et al. [14], implemented a model to detect cyberbullying via text using Machine Learning. CrowdFlower dataset was used. Tweets were converted into lowercase and URLs, usernames, white spaces, hashtags, punctuations and stop-words were removed. Tokenization and lemmatization was applied. Naives Bayes, Support Vector Machine, K Nearest Neighbour,Random forest and Logistic Regression were applied. N-gram with TFIDF, Word2vec and Doc2vec feature techniques were applied. SVM with a combination of bigram and TFIDF technique showed the best results.

A field survey conducted in the community is analysed in Fig. 1. It gathered responses of over **300** users, to give a brief of the public opinion.

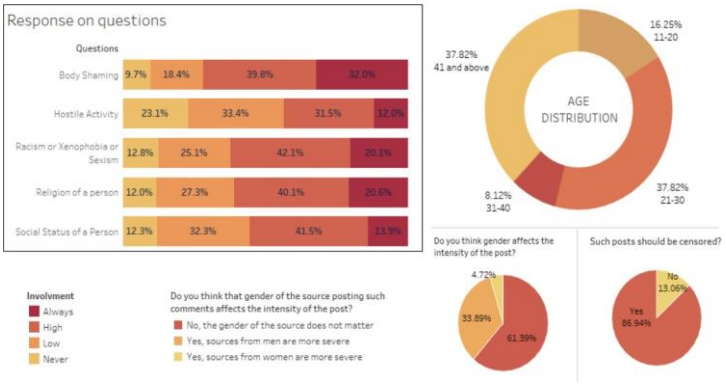


Fig. 1. Summary of field survey

The research conducted for the effects of cyber-bullying was based on a public poll. We conducted an online survey conducted with over 350 respondants with the aim to understand the current understanding of cyber-bullying among people and majorly covering all possible reasons of cyber-bullying or basis for targeting certain sections of the society. The survey covered age distribution, gender discrimination,racism, hostile activity, xenophobia, body shaming, religion or social status of a a person. It was concluded that nearly **61%** users agreed on women being subject to cyber-bullying more than men. Body shaming ans sexual orientation were the top concerns for cyber-bullying. Twitter was selected as the ideal platform to extract raw data. Twitter is one of the leading platforms for discussing all kinds of societal issues, and hence gives a large amount of views of people on various topics.

III. Research Methodology

This paper suggests solutions for cyberbullying detection using various social media site like whatsapp, twitter and you-tube.

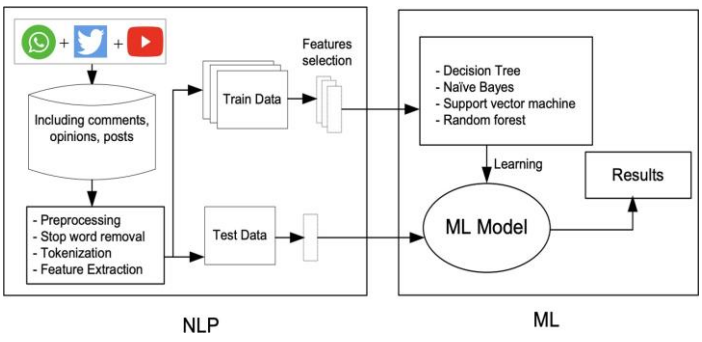


Fig. 2. Functional Block Diagram

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

In the above functional block diagram, we describe the cyberbullying detection framework which consist of two major part as follows :-

1. Natural Language Processing
2. Machine Learning.

In the first phase, we have collected real time tweets from Twitter, extracted Whatsapp chats and Youtube comments in English and Hinglish language. This real time data contains various unnecessary characters, so before applying the machine learning algorithms to our data, we need to clean and prepare the data for detection phase. In the pre-processing stage we remove hashtags, stopwords, numeric data, hexadecimal patterns and convert the text into lower case. It is done by using numpy with the help of vectorize functions. We manually created a list of stopwords for English and Hinglish language and applied it to remove these words from the clean data because the presence of these unnecessary words adversely affects the accuracy and predictions of the model. We then applied NLP techniques like Tokenization to break raw text into words called as tokens, Lemmatization to remove a given word to its root word and vectorization for converting raw text into vectors or a number.

After pre-processing we split the dataset into two parts i.e training data and testing data. Next, we applied the two important features selection of text, which are:

1. Count Vectorizer
2. Term frequency- Inverse frequency.

In this second phase, we applied various machine learning approaches like Linear SVC, Decisison Tree, Naive Bayes, Bagging classifier, Logistic Regression, Random Forest, MultinomailNB, K Neighbours Classifier and Adaboost classifier to train the model and find the accuracy for each model based on the literature survey we conducted. We also calculated F1 score for evaluation purposes and improved accuracy by repeating the stages again. We wanted to select best pair between feature selection like TF-IDF and count vectorizer and machine learning model. For this we have done a comparative analysis between count vectorizer and TF-IDF, from this comparative analysis we found out the best pair which has higher accuracy and less prediction time and made its pickle file. After that we passed the testing data to the models to compare the accuracy of various algorithms with each other. After following these stages, our model is able to predict whether the text enter is toxic i.e bullying and harmful for the society or non - toxic i.e non-bullying in Hinglish language.

A. Data extraction

Our data set consists of text in English and Hinglish language. For the English data set we scraped real time tweets from twitter and also took dataset from Kaggle. The dataset consists of actual tweets and messages which are extracted from various social media networking platforms. It has about **15,307 rows**.

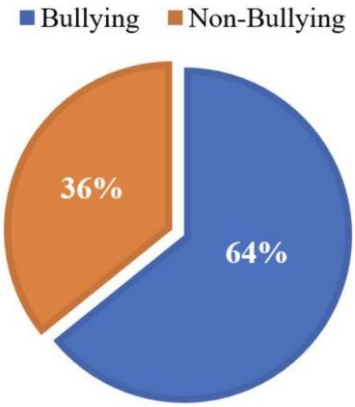


Fig. 3. Data Classification

For the Hinglish dataset we have extracted tweets from Twitter, extracted chats from whatsapp and Youtube comments. It has around **3000 rows**. We then merged them together to get a larger dataset.The dataset has two columns namely Tweets and Label. The label consists of -1 and 0 which indicates toxic i.e offensive and non-toxic i.e non-offensive sentences respectively.Our dataset has real world examples in which tweets and messages are scrapped from social media networking websites. It also has a diverse collection of negative words which are most commonly used by people in their day to day life. This would help us to detect almost every negative comment or tweet. After extracting the data the next step is preprocessing the data. It is done because real world data contains a lot of unnecessary characters, so data cleaning is required to prepare the data for the detection phase. This is a tedious but a very important task.

B. Data Cleaning

The data is required to clean before passing through multiple ML models. as shown in Fig:4, this steps are necessary to removed from the data because they do not contribute for classification phase.

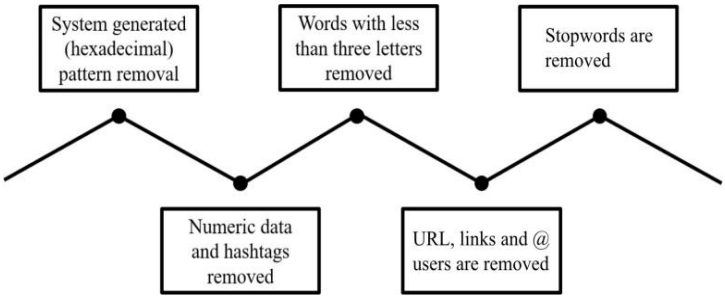


Fig. 4. process of data cleaning

When raw data of various users are imported from social media sites, it is collected with multiple characters and encoding.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

In this stage, we cleaned the data by removing punctuation marks, special characters, retweet symbols, hashtags, numeric values, hexadecimal values and URLs as they do not influence the meaning of the sentence. Words smaller than three letters long were eliminated. Also, the sentences are converted into lowercase to avoid duplication. We also manually created a list of stopwords for English and Hinglish language and applied them to remove these words from the clean data because presence of these unnecessary words adversely affects the accuracy and predictions of the model.

C. Preprocessing techniques

After cleaning the data we have applied Natural language processing techniques because the machine learning algorithm cannot work directly with the raw text that is they cannot understand the whole sentences given to it, so we transform these sentences into understandable format by using pre-processing techniques. This was followed by 3 key processes as shown in Fig:5 -

- **Tokenization** - Tokenization was used to each phrase throughout the tweet. Tokenization is the process of splitting a text sequence in smaller chunks, including sentences, words, terms, symbols that are called tokens.
- **Lemmatization** - Lemmatization is implemented after tokenization.this process is applied to reduce the inflectional forms of each word into a root word.
- **Vectorization** - Finally, Vectorization is methodology of NLP which is used to assigned weight i.e. probability to each words in a document which can be used to find word predictions and sematics.

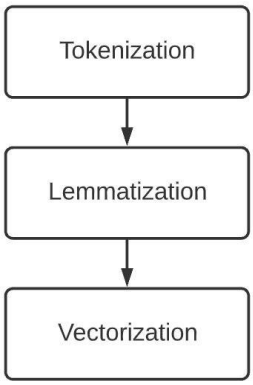


Fig. 5. Natural Language Processing Techniques

After data cleaning and applying pre-processing techniques as shown in Fig.3 and Fig.4, we split the data into training and testing.

D. Splitting the data

The datasets are divided into two types, i.e. training data set and testing data set. The testing datasets needs to be extracted from the platforms via text mining for a real time usage of the system. Both datasets pass through preprocessing techniques and various ML models.

E. Feature selection

After the splitting the data, we prepare the important features of the text, This technique helps to measure the quality of the resulting vector representations. This works with similar words that tend to close with words that can have multiple degrees of similarity. Vectorization is performed prior to sending the training and testing data set through the ML models.

1) Count Vectorization :-

Count Vectorization is used to convert the collection of words within a corpus into a vector of terms/term counts. The model will fit and learn the words from vocabulary and then try to make a word matrix in which the individual cells show the frequency of that word in a particular document, this is known as term frequency, and the columns are dedicated to each word in the corpus.

2) TF-IDF :-

TF-IDF means Term Frequency and Inverse Document Frequency, is a scoring measure which will evaluate how relevant the word in the document. This is done by multiplying two terms as follows:
1) The term frequency of a word in a document/text file.
2)The inverse document frequency of the word within a document/text file. It shows how rare or common the word is in the document. If the value is closer to 0 the more common the word is and vice versa.
Multiplying these two terms will gives the TF-IDF score of a word in the document.

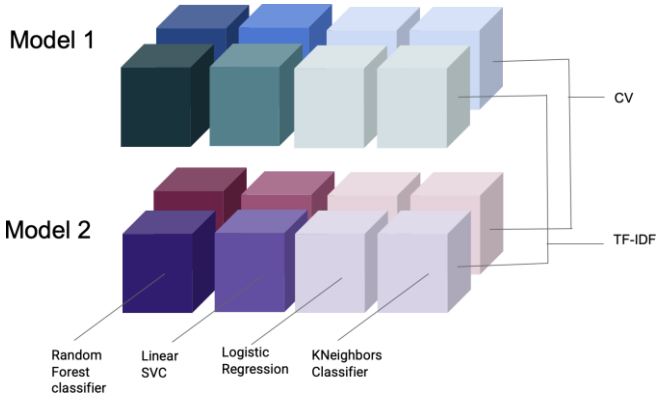


Fig. 6. Comparision between CV and TF-IDF

We have done a comparative analysis between this two feature extraction techniques with few algorithmsm through this we have observed that CV i.e. Count Vectorizer gives slightly better accuracy then TF-IDF i.e., Term Frequency- Inverse Document Frequency which is shown in the results section below in Fig.6. Hence, for predicting bullying messages,comments,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

chats and tweets in Hinglish language and to build machine learning models for classification, we selected TF-IDF as our feature selection model.

F. Machine Learning Algorithms

After these steps of preprocessing and feature selection, various machine learning models were studied and identify 8 machine learning models to compare for functionality. These models were chosen on the basis of popularity, ease of use, training and prediction time. Following are different classifiers used in study:

1) Support Vector Machine (SVM) :-

It is a classification algorithm whose objective is to fit data, and return the best fitting hyperplane which categorizes or divides data into different classes. After obtaining a hyper-plane, class can be predicted by some features of the classifier. SVM chooses extreme vectors or points which helps in creating hyperplanes called as support vectors. These vectors are nearer to the hyperplane and influence the orientation and position of the hyperplane. Hyperplanes can be drawn in infinite numbers but there is no guarantee that they all perform well which is why decision boundaries are created that are parallel to the hyperplane and touch a few supporting class vectors on one side of the hyperplane. The distance between the two decision limits of the hyperplane is called the margin and that means an error in the divider. When the margin is too high, there is less classification error. These points help us to build SVM models. SVM works well with higher dimension data and thus avoids dimensionality problems. Also it is less prone to overfitting.

2) K-Nearest Neighbors (KNN) :-

The K-Nearest Neighbors (KNN) is a simple text classification algorithm, which categorizes the new data using some similarity measure by comparing it with all available data. It generates the classification rules in the tree-shaped form, where each internal node denotes attribute conditions, each branch denotes conditions for outcome and leaf node represents the class label. In this algorithm, distance is used to classify a new sample from its neighbor. Thus, it finds the K-nearest neighbors among the training set and places an object into the class that is most frequent among its k nearest neighbors.

3) Logistic Regression :-

Logistic regression is a supervised machine learning algorithm, which is used to predict categorical dependent variables by using a set of independent variables. It is a statistical approach with which we can easily foretell a data input based on previous examinations of the data set in use. It can classify the data in 0 or 1, Yes or No, true or false and so on but instead of giving exact values it gives the probability that the given data belongs to class '1'.

4) Random Forest classifier :-

Random Forest classifier consists of large numbers of decision trees and each splits out a class prediction. The class who has higher votes becomes the model's prediction. Since it consists of multiple decision trees it is possible that some of them give wrong predictions but many others will be right. They are fast, scalable, robust to noise, do not over-fit, easy to interpret and visualize with no parameters to manage. However as the number of trees increases the algorithm becomes slow for real time prediction.

5) Bagging Classifier :-

Bagging classifier is a widely used ensemble machine learning algorithm. In ensemble algorithms a group models work together to make predictions. Advantage of this is that several different methods counteract each model's weakness resulting in less error. Each model is trained individually and combined in an averaging process. Bagging algorithm reduces model overfitting.

6) Stochastic Gradient Descent(SGD) Classifier :-

Stochastic Gradient Descent(SGD) is an optimization algorithm used to find parameters that will reduce a cost function. In other words, it is used for discriminative learning of linear classifiers under convex loss functions such as SVM and Logistic regression. SGD Classifier is a linear classifier like SVM, logistic regression but it is optimized by the SGD. in other words SGD is an optimization method, Logistic Regression or linear Support Vector Machine is a machine learning algorithm/model.

7) Adaboost Classifier :-

Adaboost is one of the ensemble boosting classifiers. Boosting algorithms try to build a strong learner or model from mistakes of other weaker models. It tries to reduce errors which arise when it is not able to identify trends in data. The AdaBoost algorithm uses one-level decision trees as weak learners then those are added sequentially to the ensemble.At each subsequent step, the model attempts to correct the predictions made by the model before it in the sequence. This is achieved by weighing the training dataset to put more focus on training examples on where previous models make predictable errors.

8) Multinomial NB Classifier :-

Multinomial NB classifier is a probabilistic machine learning algorithm which is mostly used for Natural Language Processing (NLP). The algorithm is based on Bayes theorem and predicts the tag of a text for example newspaper article or piece of mail. It calculates the probability of a given sample and returns a tag with high probability. It follows the principle that each feature being classified is not related to any other feature. The presence or absence of one feature does not affect the presence or absence of the other feature.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

G. Evaluation Phase

The bullying detection algorithms are implemented using python machine learning packages. The performances are analyzed by calculating True Negatives (TN), False Positives (FP), False Negatives (FN) and True Positives (TP). These four numbers can be shown as a confusion matrix. Different performance metrics are used to assess the performance of the constructed classifier.Some common performance measures performances are analyzed with respect to the following metrics in text categorization are

- **Precision:** - Precision is also known as the positive predicted value. It is the proportion of predictive positives which are actually positive.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall** - Recall is the proportion of actual positives which are predicted positive

$$Recall = \frac{TP}{TP + FN}$$

- **F-Measure** - F-Measure is the harmonic mean of precision and recall. The standard F-measure (F1) gives equal importance to precision and recall.

$$F\_measure = \frac{2 * precision * recall}{precision + recall}$$

- **Accuracy** - Accuracy is the number of correctly classified instances (true positives and true negatives).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

IV. Experiment And Result

We have used the four machine learning algorithms like Support Vector Machines (SVM), Random Forest (RF), K-Nearest Neighbors (KNN) and Logistic Regression(LR) to choose best feature extraction model between count vectorizer and term frequency-inverse document frequency.
In this section, we describe:

A. Comparative Analysis between 2 feature extraction models.

From the above plot Fig.7 we can say that Logistic regression and Random forest classifier gives the highest accuracy than the other two algorithms. Best accuracy and F1 score is given by Random Forest with **96.5%** and **97.0%** respectively. Linear SVC gives best recall score with **96.37%** and also takes very less time for training and predicting the output.

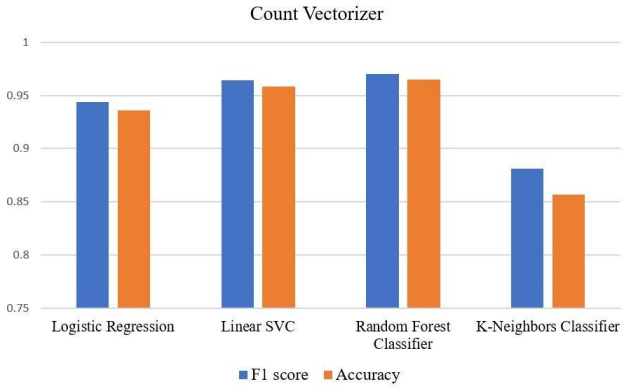


Fig. 7. Comparison of Algorithms with Count Vectorizer

Although Random Forest provides best accuracy with **96.5%**, it takes more time for training and predicting the output.

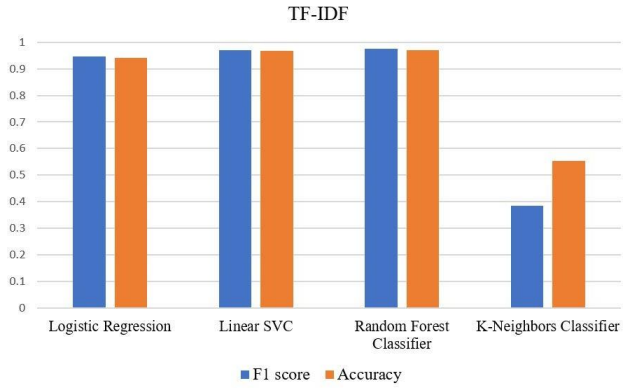


Fig. 8. Comparison of Algorithms with Term Frequency-Inverse Document Frequency

From the about plot Fig.7 we can say that Linear SVC and Random forest classifier gives the highest accuracy than the other two algorithms. Best accuracy and F1 score is given by Random Forest Classifier with **97.1%** and **97.2%** respectively. Linear SVC gives best recall score with **97.13%** and also takes very less time for training and predicting the output. Although Random Forest provides best accuracy with **97.1%**, it takes more time for training and predicting the output.

Algorithms	Logistic Regression	Linear SVC	Random Forest Classifier	K-Neighbors Classifier
CV	0.936	0.958	0.965	0.857
TF-IDF	0.940	0.967	0.971	0.554

TF-IDF gives slightly better accuracy then CV because it not only aims on the frequency of tokens present in the corpus, but also provides the importance on the tokens. We can remove the tokens that are less important for analysis, hence

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

it makes the our training model efficient and less complex by reducing the our dimensions of input.

B. Hinglish result using TF-IDF

So now we have move ahead to classify and predict bullying messages in comments, chats, tweets on various social media platforms in Hinglish language and apply more ML algorithms like Multinomial NB, Decision Tree Classifier, Ada-boost classifier and Bagging classifier with TF-IDF as feature extraction model.

Algorithms	CV Accuracy	TF-IDF Accuracy	CV F1 Score	TF-IDF F1 score
Decision Tree Classifier	0.955	0.962	0.965	0.968
Linear SVC	0.94	0.958	0.954	0.966
Bagging classifier	0.955	0.956	0.961	0.965
Logistic regression	0.935	0.944	0.949	0.949
Stochastic Gradient classifier	0.933	0.943	0.942	0.947
Multinomial NB	0.890	0.907	0.903	0.918
Ada boost classifier	0.827	0.830	0.832	0.851

From the above table, we observed that Decision Tree classifier provides highest accuracy among all the algorithms but has worst training and prediction time which is similar to random forest classifier. Linear SVC, Logistic Regression and SGDC Classifier have more or less similar performance in terms of accuracy and F1 score but among them Linear SVC and logistic regression performs faster that is provides best training and prediction time. Adaboost Classifier provides less accuracy among all the algorithms. Linear SVC and SGD (stochastic gradient classifier) is able to give a comparatively better output when adjusted with parameters on the larger dataset because it takes less time for training the algorithms and provides better accuracy then the rest.

V. Architecture of Chat Prediction Service

We have made a service wrapper using flask for our prediction model. Now whenever the group of users write or post the messages in format of text, it will request our service wrapper and our service wrapper will load the Machine learning model which is in pickle file. this ML model will predict whether the given message is bullying or non-bullying i.e either 1 or 0 and will return to the service wrapper. Later our service wrapper will respond to the users, whether the message enter is bullying or non-bullying.

VI. User Interface Design

We have created a Multi Group chat application using python sockets and Tkinder GUI. It has the functionalities to create

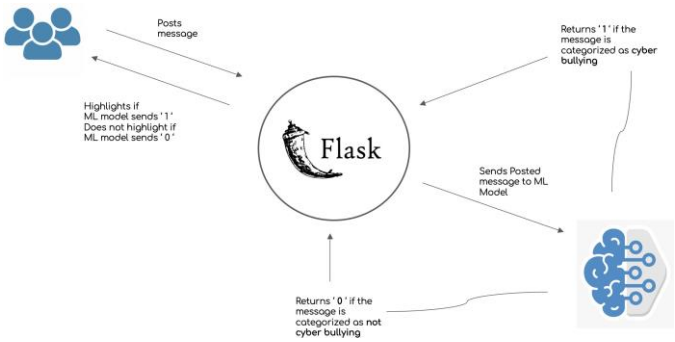
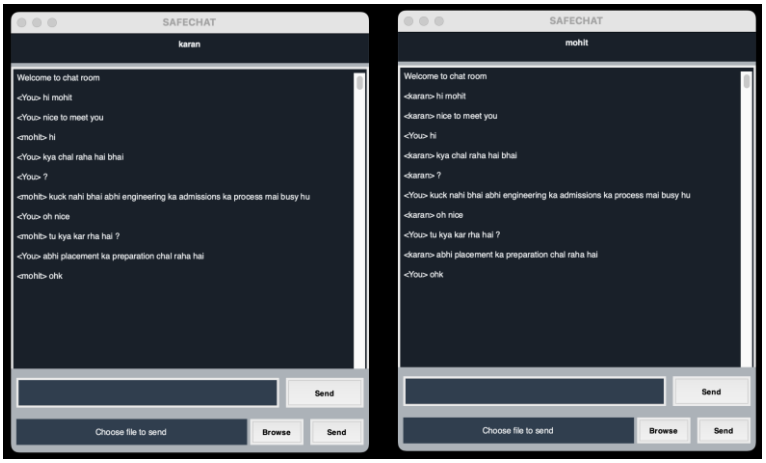


Fig. 9. Architecture of Service Wrapper

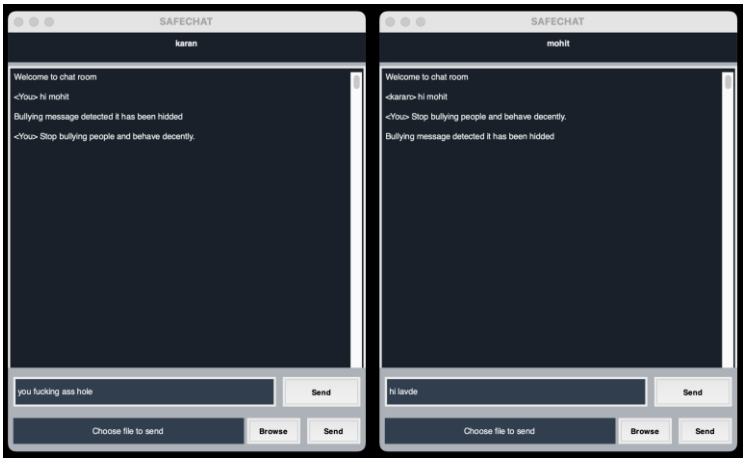
room or join room using room Id and send messages within a room.

A. Non-Bullying Flow

Whenever the user posts a message in the chat, our prediction service will the load the model and if the text enter is categorized as non-bullying then text or messages will be displayed on the chat screen as shown in the fig below.



B. Bullying Flow



International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

Whenever the user posts a message in the chat, our prediction service will load the model and if the text enter is categorized as bullying, then the message will be not displayed on the chat screen, the sender will get the warning as Stop bullying people and behave decently and the receiver will not receive the bullying message. Instead, they will be informed that a bullying message has been detected it and it is hidden as shown in the above figure

VII. Conclusion

Thus we have successfully been able to extract the data , clean it, and visualize it using various python libraries. We also implemented various natural language processing techniques like tokenization, lemmatization and vectorization i.e. feature extraction. After reading various research papers published in this field we analyzed that in feature extraction, count vectorizer and TF-IDF are the two methods which are giving very good accuracy compare to word2vec and bag of words. So for selecting best feature extraction between count Vectorizer and TF-IDF, we have done comparative analysis between this two feature extraction models and observed that count Vectorizer slightly provides better accuracy then TF-IDF. We identified various algorithms and try to apply some of them in our project like Support Vector Machine, Logistic Regression, K Nearest Neighbor and Random Forest, Bagging Classifier,Decisison Tree Classifier,SGDC classifier, Multinomial Classifier, and AdaBoost Classifier. We then trained our models and obtained good accuracy as well as speed while applying these algorithms with count vectorizer as feature selection model.After training we summarized all the Algorithms in one plot with Accuracy and F1 score. After observing the results we noted that Linear SVC and SGD (stochastic gradient classifier) is able to give a comparatively better results in classifying and predicting bullying messages in Hinglish languages and takes less time to train and predict then other algorithms

VIII. Future Scope

This research work can be improved in the future by doing the following work.Firstly, the accuracy of the models can further be increased to get better results by using deep learning. Next, detection can be done in more languages such as Gujarati, Marathi, Tamil, Telugu, Kannada etc. Inshort, the project can be made more diverse to make it applicable for multiple applications. For this diverse datasets for the required languages are required and a list of stopwords is required. Then a similar procedure can be implemented. We have developed a model which classifies cyberbullying through text only. In the future, we can develop a model which can classify cyberbullying through images and videos.

References

[1] B. Dean, “How many people use social media in 2021? (65+ statistics),” Sep 2021. [Online]. Available: https://backlinko.com/social-media-users

[2] J. W. Patchin, “Summary of our cyberbullying research (2004-2016),” Jul 2019. [Online]. Available: https://cyberbullying.org/summary-of-our-cyberbullying-research

[3] Noviantho, S. M. Isa, and L. Ashianti, “Cyberbullying classification using text mining,” in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, 2017, pp. 241–246.

[4] C. Van Hee, G. Jacobs, C. Emmery, B. Desmet, E. Lefever, B. Verhoeven, G. De Pauw, W. Daelemans, and V. Hoste, “Automatic detection of cyberbullying in social media text,” *PloS one*, vol. 13, no. 10, p. e0203794, 2018.

[5] M. A. Al-Garadi, M. R. Hussain, N. Khan, G. Murtaza, H. F. Nweke, I. Ali, G. Mujtaba, H. Chiroma, H. A. Khattak, and A. Gani, “Predicting cyberbullying on social media in the big data era using machine learning algorithms: Review of literature and open challenges,” *IEEE Access*, vol. 7, pp. 70 701–70 718, 2019.

[6] K. Sahay, H. S. Khaira, P. Kukreja, and N. Shukla, “Detecting cyberbullying and aggression in social commentary using nlp and machine learning,” *International Journal of Engineering Technology Science and Research*, vol. 5, no. 1, 2018.

[7] M. Di Capua, E. Di Nardo, and A. Petrosino, “Unsupervised cyber bullying detection in social networks,” in *2016 23rd International conference on pattern recognition (ICPR)*. IEEE, 2016, pp. 432–437.

[8] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra, “Detection of cyberbullying incidents on the instagram social network,” *arXiv preprint arXiv:1503.03909*, 2015.

[9] V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak, “Detection of cyberbullying using deep neural network,” in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. IEEE, 2019, pp. 604–607.

[10] H. Watanabe, M. Bouazizi, and T. Ohtsuki, “Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection,” *IEEE access*, vol. 6, pp. 13 825–13 835, 2018.

[11] J. Yadav, D. Kumar, and D. Chauhan, “Cyberbullying detection using pre-trained bert model,” in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2020, pp. 1096–1100.

[12] A. Gaydhani, V. Doma, S. Kendre, and 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.

[13] L. Ketsbaia, B. Issac, and X. Chen, “Detection of hate tweets using machine learning and deep learning,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 751–758.

[14] S. Abro, Z. S. Shaikh, S. Khan, G. Mujtaba, and Z. H. Khand, “Automatic hate speech detection using machine learning: A comparative study,” *Machine Learning*, vol. 10, no. 6, 2020.