



Detection And Comparative Analysis Of Cyberbullying Behavior In Twitter Data Using Multiclass Classification Algorithms

¹Shreedhar Chavan, ²Nishad Marathe, ³Prasad Thorat, and ⁴ K.S. Charumathi

^{1,2,3}Student,, ⁴Professor at Department of Computer Engineering

Pillai College of Engineering, University of Mumbai, New Panvel, India

Abstract: Cyberbullying or cyber harassment is a form of bullying or harassment using electronic means. Cyberbullying and cyber harassment are also known as online bullying. It has become increasingly common, especially among teenagers, as the digital sphere has expanded and technology has advanced. In this research, we have addressed the problem of cyberbullying detection on Twitter Data Set. Various inbuilt Multiclass Classification Algorithms in Python such as Naive Bayes, Decision tree, Random Forest, K Nearest Neighbor(KNN), Support Vector Machine (SVM), and Natural Language Processing(NLP) techniques are used to classify bullying and non-bullying tweets and performance of these algorithms is compared and model with highest accuracy is selected for the prediction. Bullying tweets will be reported to the nearest Cyber Crime Branch.

Index Terms - Cyberbullying, Machine Learning, Twitter, Multiclass Classification

I.INTRODUCTION

Cyber-bullying is an aggressive act that can be carried out by a single person or a group of aggressive people, using online platforms, repeatedly against a person who is not capable of defending himself. Twitter is one of the social networking services on which users communicate using tweets. Cyber-bullying is especially present on Twitter according to the survey done by Pew Center, and Twitter users face many forms of bullying such as death threats, stalking and sexual abusive threats.

What is Multi-Class Classification?

Multi-class classification refers to those classification tasks that have more than two class labels.

Examples include:

- Face classification.
- Plant species classification.
- Optical character recognition.

Unlike binary classification, multi-class classification does not have the notion of normal and abnormal outcomes. Instead, examples are classified as belonging to one among a range of known classes.

The number of class labels may be very large on some problems. For example, a model may predict a photo as belonging to one among thousands or tens of thousands of faces in a face recognition system.

Problems that involve predicting a sequence of words, such as text translation models, may also be considered a special type of multi-class classification. Each word in the sequence of words to be predicted involves a multi-class classification where the size of the vocabulary defines the number of possible classes that may be predicted and could be tens or hundreds of thousands of words in size.

In this research, we have proposed the model for cyberbullying detection on Twitter Platform using various Multiclass Classification Algorithms such as Naive Bayes, Decision tree, Random Forest, K Nearest Neighbor (KNN), Support Vector Machine (SVM), and Natural Language Processing (NLP) techniques to classify bullying and non-bullying tweets and report them to nearest cybercrime branch.

II. RELATED WORK

Cyberbullying Detection: A Comparative Analysis of Twitter Data [1]:

In this paper, authors have attempted to conduct analysis on "tweets" using various inbuilt machine learning algorithms in python. Author had attempted to classify the tweets as bullying and non-bullying. Supervised machine learning techniques like Linear regression, logistic regression, Naive Bayes, SVM, Decision trees and neural networks are used.

The author has used feature extraction techniques such as Bag of Words (BOW) and TF-IDF. Then the word cloud is used to visualize the most frequent words in tweets. The dataset consists of 18,000 tweets out of which 75% were used for training the model and the rest 25% were used for testing the model. The confusion matrix and bar plot were used to visualize the performance of the various algorithms. When recall is high and precision is low, most of the bullying tweets were correctly identified as bullying tweets and few non-bullying tweets were identified as bullying tweets. Out of all algorithms the SVM has performed better with TF-IDF and Bag of Words as a feature extraction method.

Cyber-Bullying Detection using Machine Learning Algorithms [2]

Authors have selected two distinct datasets, which are recently published, related to the social network FormSpring.me and YouTube.

In this work authors have checked the density of "bad" words as a single feature. This feature is equivalent to the number of bad words that appear in a sentence, for each severity level, divided by the words in the same sentence.

Badness of the word: Authors also have added a feature to their work in order to measure the overall "badness" of a text. This feature is computed by taking a weighted average of the "bad" words (weighted by a severity assigned).

Density of upper case letters: This feature is based on Dadvar et al. [4] results. The presence of capital letters in a text message is selected as a feature, considering it as possible 'shouting' at someone behavior, as commonly treated in social networks netiquette. This feature is given by the ratio between the number of uppercase letters and the length (number of chars) of the whole sentence.

Exclamations and questions marks:

Just like capital letters, also exclamation points and question marks can be considered as emotional comments. Authors just stated that cyber bullying is related to an extreme case of sentiment analysis and so it can be connected to the strong (usually bad) emotions. With this premise, authors consider it helpful to introduce the number of exclamation points and question marks as a feature in their work.

Comparative Study of Cyberbullying Detection using different Machine Learning Algorithms [3]

In this research, authors have focused on the detection of cyberbullying using the Instagram data set, with help of four different machine learning algorithms viz. Naive Bayes, SVM, Decision tree and logistic regression. Authors have found that comparatively Naive Bayes algorithms with trigram have the highest performance of 79% accuracy than other three algorithms.

The data collected was based on a set of keywords which includes Muslim, god, tattoo, pray, Hindu, bjp, etc. Authors have chosen these keywords by analyzing different profiles and found that these are the most commonly used keywords. The extracted fields of a profile from Instagram include the entire posts of the profile along with captions, hashtags, created time and the biography of the user and comments related with each post. The biography has an important role in ease filtering of data i.e., if bio contains words like quotes, memes, awareness, motivation etc. then we can avoid those profiles in the initial

Authors found out that these kinds of public pages will have more followers than followed by count. These points were used to filter the public pages like motivation pages or awareness pages etc. The authors took the comments of each post and labeled it manually as either bullying or non-bullying. A total of 1065 comments were considered and in which 636 were non-bullying comments and the following were bullying one. Authors divided the entire data set as training as well as test data and each contains 746 and 319 respectively. During each algorithm call for every n-gram combination the authors have shuffled the dataset.

III. PROPOSED METHODOLOGY

The working of the proposed system is the same as the existing system[1] except that for collection of twitter data we are using OSINT (open source intelligence) project “twint”.

```
In [27]: # Data fetching
#!pip3 install git+https://github.com/Museum-Barberini/twint.git@fix/RefreshTokenException#egg=twint
import nest_asyncio
nest_asyncio.apply()
import twint, time

try:
    c = twint.Config()
    # c.Username = input("Enter the username : ")
    c.Search = input("Enter the topic you want to search : ")
    c.Lang = "en"

    # c.Near = "India"
    c.Limit = 100
    c.Min_Likes = 100
    c.Store_csv = True # store tweets in a csv file
    c.Output = "D:/Programming/BE PROJECT/test_tweets"+c.Search+".csv" # path to csv file
    twint.run.Search(c)

except Exception as e:
    print(e)
```

```
Enter the topic you want to search : nigga
1507228663250661405 2022-03-25 11:01:30 +0530 <catacombreaper> @snoopyoop7 I would've even punch that nigga ew 🤔
1507228655478616074 2022-03-25 11:01:29 +0530 <ksogranddd> a bitch add me to they cfl and my nigga posted, i just feel like they wanna fight
1507228640140046344 2022-03-25 11:01:25 +0530 <durshon258> I ain't ever been the nigga get money 🤔 mistreat mf go after bitches who wouldn't fuck
with me if I ain't have shit 🤔 if a mf knew me
1507228636683939858 2022-03-25 11:01:24 +0530 <TgotTorch> Whatever nigga I WAS THE ONE SHE WAS TALKING ABOUT
1507228634393849930 2022-03-25 11:01:24 +0530 <Nnando_barbs> @sparkybwoi @ChineEvans Ontop you? 🤔🤔🤔Nigga you got to be kidding 🤔 h
https://t.co/z51z5DKAc7
1507228632661602323 2022-03-25 11:01:23 +0530 <6godwoi> Niggas brought in Power like she's not a fucking idiot
1507228626005241921 2022-03-25 11:01:22 +0530 <crooksdscentence> @jorgemikell nigga lts way too late for that lol
1507228623031480320 2022-03-25 11:01:21 +0530 <NOHEZZO> Ion big bro no nigga and ion let no nigga lll bro me
1507228621777383436 2022-03-25 11:01:21 +0530 <arrogantave> She swipe. She ain't impressed by yo bread nigga lol
1507228620753973272 2022-03-25 11:01:20 +0530 <nino_dior> Esa rola de nigga, "Luna" siempre será de mis favoritas.
1507228615001001778 2022-03-25 11:01:19 +0530 <NassahaB007> @Mls_halle Of 1989 rafinator shared see 🤔
```

Fig 3.1 tweets scraping using twint tool

Twint is an advanced Twitter scraping tool written in Python that allows for scraping Tweets from Twitter profiles without using Twitter's API.

1. Some of the benefits of using twint vs Twitter API:
2. Can fetch almost all Tweets (Twitter API limits to last 3200 Tweets only)
3. Fast initial setup
4. Can be used anonymously and without Twitter sign up
5. No rate limitations

Also in the proposed system we are adding functionality of extracting text from images

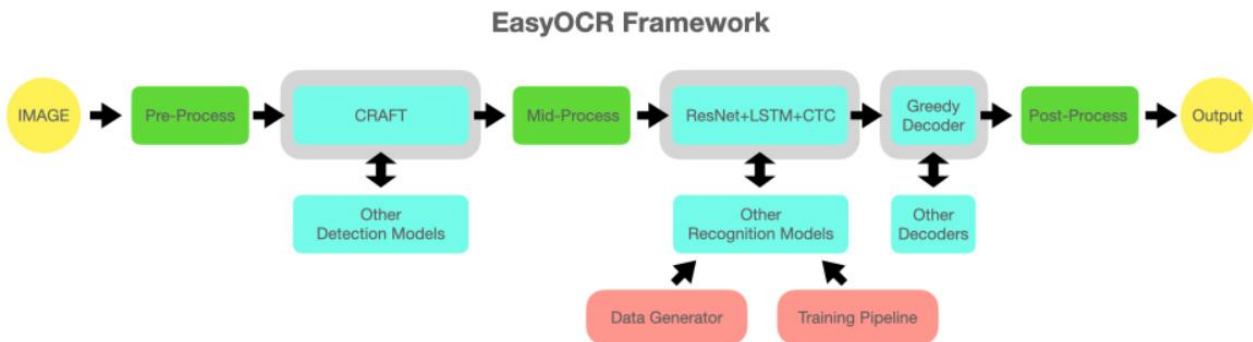


Fig. 3.2 Easy OCR workflow (Image Source: [Github](#))

EasyOCR is actually a python package that holds PyTorch as a backend handler. EasyOCR like any other OCR(tesseract of Google or any other) detects the text from images but in my reference, while using it I found that it is the most straightforward way to detect text from images also when high end deep learning library(PyTorch) is supporting it in the backend which makes it accuracy more credible. EasyOCR supports 42+ languages for detection purposes. EasyOCR is created by the company named Jaided AI.

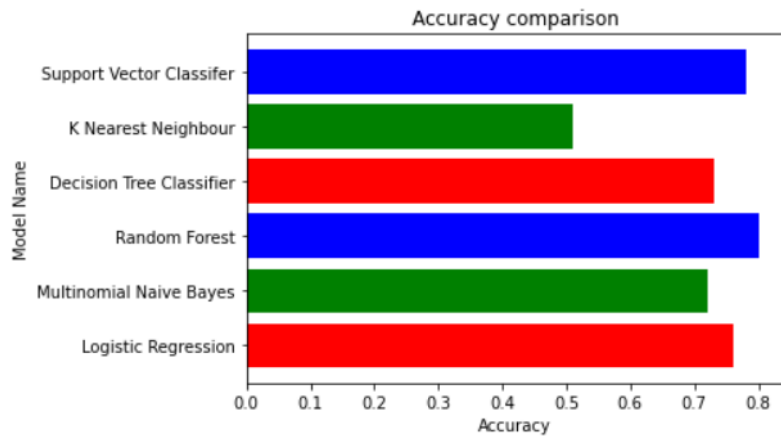


Fig 3.8 Accuracy comparison of Models

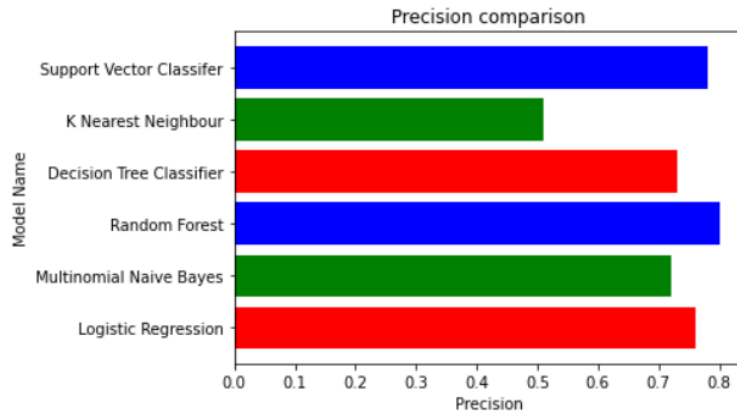


Fig 3.9 Precision Comparison of Models

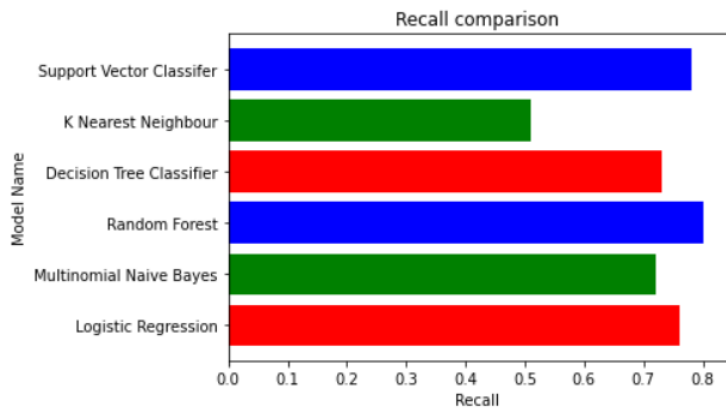


Fig 3.10 Recall Comparison of Models

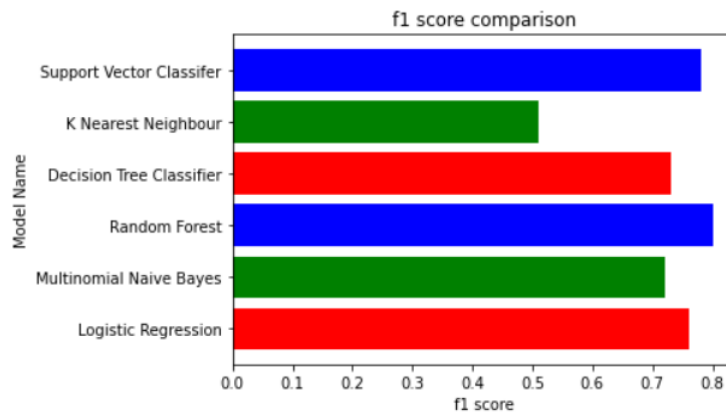


Fig 3.11 F1 Score Comparison of Models

Prediction on real time data

To test the model the tweets are scraped in real time using Twint. Once the data is scraped it goes into preprocessing where we allow only tweets with english language. The NLP techniques such as tokenization, lemmatization are used to build the corpus.

To predict the text based tweet we are using two conditions:

1. Check whether text tweet without hashtags is classified as bullying tweets
2. If not proceed for the hashtags and check whether the hashtags are classified as bullying tweets.

To predict the text based images in tweet we are using three conditions:

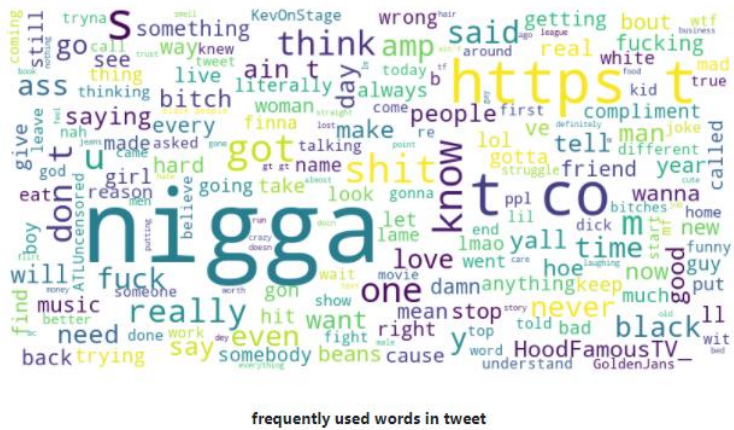
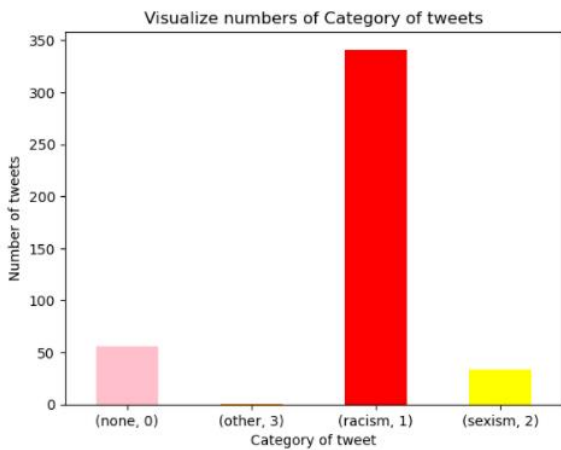
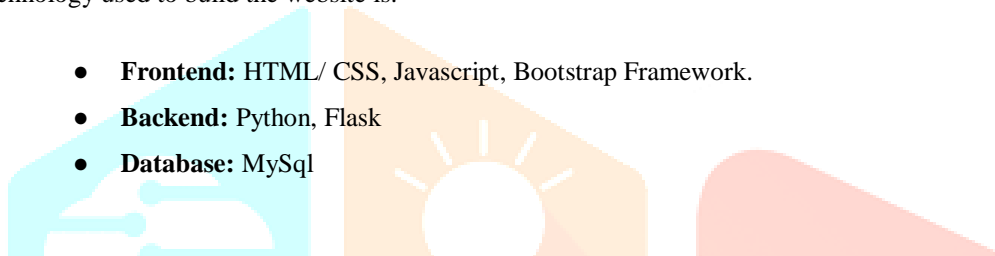
1. Check whether text tweet without hashtags is classified as bullying tweets
2. If not proceed for the hashtags and check whether the hashtags are classified as bullying tweets.
3. If not proceed to extract the text from the image and classify the text in image.

IV. OUTPUT

We have been building a website where users can create an account on our website and search the bullying tweets and finally report them to the nearest Cyber Crime Branch.

The technology used to build the website is:

- **Frontend:** HTML/ CSS, Javascript, Bootstrap Framework.
- **Backend:** Python, Flask
- **Database:** MySql



Result:

SNo	Tweet	Prediction	Report
1	My nigga, he a stepper, he gon die bout his bitch 🤔🤔🤔 #777TheAlbum	Sexism Tweet	Report
2	I just want these niggas to keep they distance. I don't even wanna be bothered honestly.	Racism Tweet	Report
3	Any nigga with money can take your girl work hard brooh 🤔	Racism Tweet	Report
4	trap music so crazy bc niggas b algorithmed up with they opps 🤔 it b streaming codependence till somebody die	Racism Tweet	Report

Fig 4.1 Real Time Prediction for the text in the tweets.



Fig 4.2 Real Time Prediction for the text in images in the tweet.

Cyberbullying Detection Text Profile Image Logout

Choose your Cybercrime Branch:

Cybercrime Branch email:

- Mumbai
- Banglore
- Delhi
- Kolkata
- Chennai

Message:

Tweet link :
<https://twitter.com/laury7n/status/1507239821772795904>

Tweet category : sexism

Fig 4.3 Reporting the tweet to the nearest Cyber Crime Branch

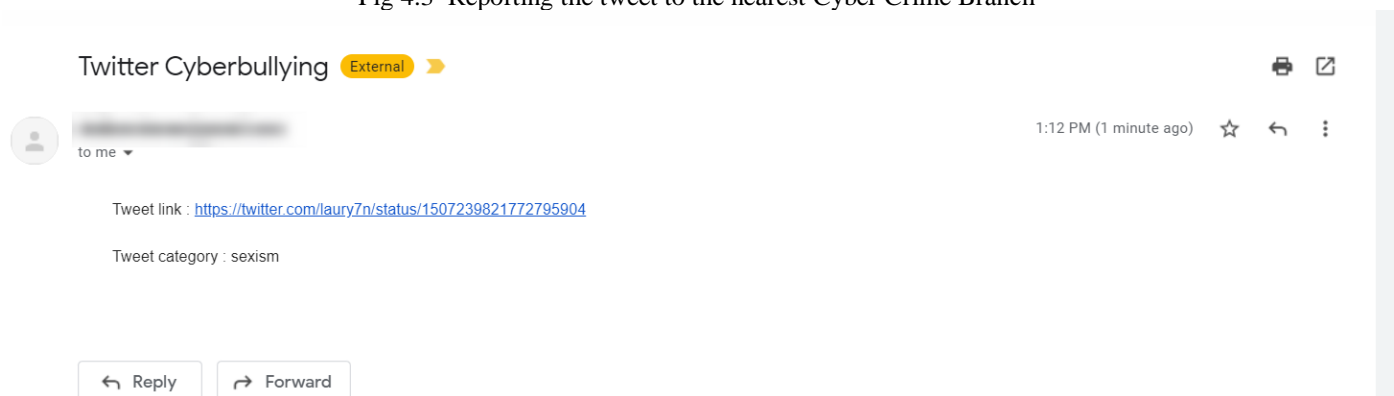


Fig 4.4 Report received at Cyber Crime Branch.

Welcome back Steve Jobs

Your report has been sent to Cybercrime Branch. Thank you for reporting!!

Enter the keyword :

nigga

Predict

Fig. 4.5 Acknowledgement of Report to the user.

V. CONCLUSION

As the dataset contains only 10 thousands tweets we have achieved only 80% accuracy. Also the algorithms are not tuned. If we increase the size of our dataset and tune the algorithm we can achieve more accuracy. The future scope of the research is to to classify and predict the videos in the tweet.

VI. ACKNOWLEDGEMENT

We would like to thank our Guide and Head of the Department, Dr Satish Kumar Verma and project coordinator Dr. Sushopti Gawade who always encourages us to do innovative things that will help to increase our knowledge.

REFERENCES

- [1] Jyothi Shetty, K.N. Chaithali, Aditi M. Shetty, B. Varsha, and V.Puthran (2020). Cyber-Bullying Detection: A Comparative Analysis of Twitter Data
- [2] Anvitha Keni ,Deepa, Prof Mangala Kini, Deepika K V.,,Divya C H (2020) Cyber-Bullying Detection using Machine Learning Algorithms
- [3] Rohini K R, Sreehari T Anil, Sreejith P M, Yedu Mohan P M (2020). Comparative Study of Cyberbullying Detection using different Machine Learning Algorithms
- [4] M. Dadvar and F.de Jong. 2012."Cyberbullying detection:a step toward a safer internet yard". In Proceedings of the 21st International Conference on World Wide Web(WWW '12 Companion). ACM, New York, NY, USA, 121-126