RP-2

by SURESH K

Submission date: 31-Mar-2023 04:10PM (UTC+0530)
Submission ID: 1571244227
File name: RP-2.doc (828.5K)
Word count: 4968
Character count: 27180

Cyber Bullying Detection Using Machine Learning Techniques

I ABSTRACT

As cyberbullying has become increasingly common, there has been a lot of focus on cyberbullying detection because cyberbullying has a fatal effect on both users of social media platforms and society. Social media has become a channel for many people to communicate their opinions, knowledge, and so on. It would have been one of the best things if it had not become a tool for many people to exact revenge, manipulate others, or harass and humiliate others. As a result, we do not require a monitoring system to control the misbehavior and bullying that spreads through social networks, which has led to the development of this model to automate the identification of cyberbullying. Our main goal is to create a model that categorizes comments as positive or negative.

Keywords
Cyber Bullying, Sentiment Analysis, Personality Analysis.

II INTRODUCTION

Cyberbullying is just bullying or harassing others via various Social Media platforms. As the technological era has progressed, the use of social media platforms has skyrocketed, with more than half of the world's population now utilizing them. As its popularity has grown, a piece of news or other information can now be shared with others in seconds. This is now one of its most essential applications. However, some people took advantage of this and began to seek enjoyment from it. Cyberbullying has since begun and has grown significantly.

Cyberbullying is known as an unseen crime, and many victims have been tormented online through toxic comments. According to some data, as many as Cyberbullying has affected 50% of children. Victims of cyberbullying may endure a variety of consequences, including mental health concerns, poor academic performance, a desire to drop out, and even suicidal thinking. Despite the fact that many victims confront this, many of them go unrecognized due of social standing and other factors. Cyberbullying also taints a person's image. The false rumors disseminated about them harm their reputation. Approximately 8 out of 10 children are victims of various forms of cyberbullying.. Machine learning is used to detect cyberbullying by identifying and classifying instances of online harassment, abuse, and bullying. This technique often entails training a model on a large dataset of cyberbullying cases as well as non-cyberbullying text examples. The model can then be used to predict whether future occurrences of online text are likely to be cyberbullying. Personality Analysis, Sentiment Analysis, and User Traits are some of the factors that the algorithm takes into account while making these predictions. The purpose

of this technique is to create a rapid, scalable, and automated mechanism to detect and solve cyberbullying, so that online communities can be safer and more supportive for everyone.

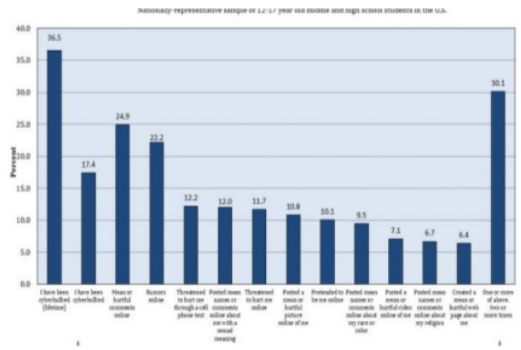


Fig.1.CyberBullying Victimization and Reasons

There have been numerous incidents of cyberbullying, ranging from a mother seeking vengeance for her child, which resulted in the suicide of a thirteen-year-old girl, to people tormenting others for the sake of pleasure.Initially, this was taken casually, and there were no rules governing it. However, it has grown swiftly, and many individuals are victims of it. Despite the fact that there are numerous laws in place to punish individuals who bully others. Despite this, the number of has never dropped. So we're opting for a preventative technique in which we'll employ a machine learning model to recognize phrases and sentences that are used to injure or harass people and then replace them with positive comments. Here, we'll apply the concept of ensemble algorithms to improve the model's accuracy.

Fig.2.Types of CyberBullying

RESEARCH OBJECTIVES

1. We would be using various machine Learning Techniques to build a model that accurately identifies instances of cyberbullying.

2. To develop a model that Classifies the type of cyberbullying being used .Various types of cyberbullying being harassment, threats, spreading rumors, Flaming etc.

3. To train and test the model using the twitter dataset that is available online.

4. The dataset consists of both positive and negative contexts of cyberbullying.

5 .The primary objective is to reduce the rate of cyberbullying.

6. Optimize the performance of model using ensemble algorithms.

III SCOPE OF THE WORK

The system will employ the supervised learning methodology. The dataset will contain two sets of data, both good and negative in the context of cyberbullying. The proposed method will detect cyberbullying and convert negative remarks into good ones. We'll use various kinds of algorithms like Random Forest, decision tree learning, Linear Regression, Naïve Bayes.

IV PROBLEM STATEMENT

Cyberbullying is the use of numerous social media platforms to manipulate, torment, or harass another person, and it is a developing concern in today's technology day. With the increased use of social media and other online platforms, cyberbullying has increased significantly, causing a slew of serious issues in victims such as depression and inferiority complexes. Machine Learning has various sets of algorithms that are used to develop various types of predictive models. This has simplified many challenges, one of which is cyberbullying detection.

Cyberbullying has become more common as the use of social media platforms has grown.

V LITERATURE SURVEY

Machine Learning contains several algorithms that are used to create various types of predictive models. This has simplified many challenges, one of which is cyberbullying detection. Cyberbullying has become more common as the use of social media platforms has grown. Because detecting cyberbullying involves a vast amount of data that is continually increasing, we require a model that recognizes the cyberbullying context in real time. It is possible to achieve this by utilizing machine learning techniques such as naive Bayes, Decision tree, and Random forest, as well as ensemble algorithms to optimize the model.

A RELATED WORK:

Andreas Weiler et al investigated the run-time and task-based performance of numerous Twitter event recognition algorithms. The writers took a two-pronged approach. The scientists gathered a dataset of cyberbullying-related tweets and developed a machine learning classifier to detect cyberbullying in real-time. To identify cyberbullying tweets, the classifier used a support vector machine (SVM) algorithm and a collection of variables such as sentiment analysis, frequency of abusive terms, and user mentions.

A study developed a method for detecting foul language that included a syntactic element and performed better than the standard learning-based method. (Chen, Zhou, Zhu, & Xu, 2012). In a YouTube data-based instance (Dadvar, Trieschnigg, Ordelman, & de Jong, 2013), SVM was used to detect cyberbullying, and it was discovered that integrating user-generated content boosted SVM detection accuracy. Dadvar et al. used MySpace data sets to create their findings.

Cynthia Van Hee and Gilles Jacobs' research on automatic cyberbullying identification in social media text. It's interesting to see the approach they took in creating corpora by gathering data from social networking sites and using online bullying standards to model postings authored by bullies and victims.
The pre-processing steps of tokenization, PoS-tagging, and lemmatization are commonly used in natural language processing and can help to transform unstructured text into a structured format that can be used for machine learning algorithms. It's also notable that they tested language conversion and subsequent accuracy by constructing models for both English and Dutch.The accuracy results of 64% for the English language and 61% for the Dutch language are modest, but they are a good starting point for further research and improvements to the ML algorithm SVM. Automatic cyberbullying identification in social media text is a challenging task because of the complexity and diversity of language used in social media communication.

Hannah L. Schacter et al investigated how cyberbullying victims' disclosures on Facebook influence bystanders' features of guilt, empathy, and ideas of intervening on behalf of a casualty following the completion of a cyberbullying incident. The researchers chose 118 participants at random to read the Facebook page of a cyber-victim who had posted an update with varied levels of personal disclosure and valence. (i.e., positive or negative tone).The study's findings revealed that participants who examined the high personal disclosure profile, regardless of valence, blamed the victim more and felt less empathy for the victim. As a result, the likelihood of bystander intervention in the bullying incidence was anticipated to be lower. According to the authors, this effect may be attributable to the victim's perceived responsibility and deservingness based on their level of personal exposure.

S.E.Vishwapriya, Ajay Gour, and colleagues' research on detecting hate speech and objectionable language on Twitter using machine learning. It's interesting to see that they used datasets from Crowdflower and GitHub, which provided labeled data for categories such as Hateful, Offensive, Sexism, and Racism. They also pre-processed the data by lowercasing tweets, removing Space Pattern, URLs, Twitter Mentions, Retweet Symbols, Stop words, and using stemming to minimize inflectional forms of words. Dividing the dataset into 70% training and 30% test samples is a common practice in machine learning to evaluate the performance of the model on unseen data. It's also notable that they used N-gram features weighted by their TF IDF values, which is a popular approach for

text classification tasks.Comparing the algorithms of Logistic Regression, Naive Bayes, and Support Vector Machine is also a common practice in machine learning to identify the most suitable algorithm for the task at hand. In this case, Logistic Regression with L2 Normalization and n=3 yielded an impressive accuracy of 95%. This suggests that their approach has the potential to be useful for automated detection and prevention of hate speech and objectionable language on Twitter.

Matthew Pittman and Brandon Reich performed a mixed-design survey to investigate whether text- and image-based social media sites like Twitter and Yik Yak, as well as platforms like Instagram and Snapchat, can help reduce loneliness by promoting more intimacy. Since young adults are the group most likely to use social media and experience loneliness, they were the focus of their study. Both quantitative and qualitative information from the survey was analyzed by the researchers. The quantitative findings revealed that as a result of using image-based social media, loneliness may diminish while happiness and life satisfaction may increase. Use of text-based media, in contrast, seemed ineffectual in this regard. The qualitative findings revealed that the increased intimacy provided by image-based social media use was what caused the observed impacts. Using image-based social media platforms reportedly increased participants' sense of community and sense of shared experience.

College students' experiences with various types of social media cyberbullying on social networking sites were the subject of a study by Kassandra Gahagan. (SNS). 196 undergraduate students from a Northwestern university participated in the study. The study's findings revealed that 46% of college students had seen cyberbullying on SNS and that 19% of them had experienced bullying on the platform. In addition, 61% of college students who saw cyberbullying on SNS took no action. The college students were asked about their experiences with cyberbullying as well as their perceptions of their responsibilities when they observed cyberbullying on SNS. Two distinct themes that arose from the responses were found by the study. Some college students thought their duty to step in depended on the circumstances, while others said there was a consistent, obvious level of obligation for college students who witnessed cyberbullying on SNS. According to the study's findings, college students frequently engage in cyberbullying on social networking sites, and bystander assistance is frequently missing. The report also emphasizes the necessity of bystander education and awareness efforts to encourage appropriate and proactive behavior in cases of cyberbullying.

H. Watanabe, M. Bouazizi, and T. Ohtsuki's research on detecting hate speech on Twitter using unigrams and automatically gathered patterns.It's interesting to see that they employed three different datasets to train and evaluate their model, which included tweets categorized as clean, offensive, hateful, sexist, racist, or neither. Pooling these datasets together is a common approach to create a larger and more diverse dataset for machine learning.Pre-processing steps such as removing URLs and tags from tweets, as well as lemmatization, part of speech tagging, and tokenization are commonly used in natural language processing to transform unstructured text into structured format that can be used for machine learning algorithms.Using unigrams and automatically gathered patterns from the dataset is a popular approach for text classification tasks, especially for detecting hate speech and offensive language. However, it's important to note that unigrams may not capture the full context and meaning of the text, so it may be necessary to consider other features such as bigrams, trigrams, or even semantic features.

Lida Ketsbaia, Biju Issac, et al.'s research on identifying hateful and harmful tweets using machine learning and deep learning.It's interesting to see that they used datasets from two different universities, with labels of "Hate" and "Non-Hate" to train their model. Pre-processing steps such as changing tweets to lowercase, eliminating numerals, URLs, and user mentions, as well as removing stop words, punctuation, special characters, and contradictions are common steps in natural language processing to clean up and standardize text data.Using unigrams, bigrams, and trigrams along with classifiers such as Bernoulli, Multinomial, and Logistic Regression is a popular approach for text classification tasks, including identifying hateful and harmful tweets. It's notable that they were able to achieve a high level of accuracy, up to 96%, by using the Word2Vec method, which is a popular technique for embedding words in a high-dimensional vector space.

VI METHODOLOGY

We use various machine learning algorithms together to detect the context of cyberbullying .Here we will be using a supervised machine learning model as the dataset is of labeled data .The dataset has both the positive comments and negative comments .we will be using various algorithms like naïve bayes ,Random Forest and decision tree .Next we are going to combine all these using ensemble algorithms which are going to give us the optimized output .The performance also increases .Along the bunch of words here we would be concentrating on personality analysis .

The dataset is first collected, then divided into two halves. Training accounts for 75% of the time while testing accounts for 25%. Then, various machine learning techniques are used to build the model .Then this model is used to predict the context of cyberbullying.

A MODEL DESCRIPTION

Building a machine learning model to identify cyber bullying requires several steps:

Data Collection:

Collect a large and diverse dataset of text (such as social media posts or online comments) that contain instances of cyber bullying and noncyber bullying examples.

Data Preprocessing:

Clean the data by removing any irrelevant information, such as URLs, and perform any necessary text normalization, such as converting all text to lowercase.

Feature Engineering:

Decide on the features that the model will use to make its predictions. In the case of text classification, this usually involves creating numerical representations of the text, such as using term frequency-inverse document frequency (TF-IDF) or word embedding's.

Model Selection:

Choose an appropriate machine learning algorithm for the task of text classification, such as a support vector machine (SVM), a neural network, or a random forest.

Model Training:

Train the model on the preprocessed and feature-engineered data. This involves providing the model with the features and corresponding labels (e.g., cyber bullying or non-cyber bullying) for each instance in the training data.

Model Evaluation:

Evaluate the performance of the model by testing it on a held out test set and computing metrics such as accuracy, precision, recall, and F1 score.

Model Fine-tuning:

If necessary, fine-tune the model by adjusting its parameters or trying a different algorithm, and then repeat the evaluation step.

B CYBER BULLYING DETECTION MODEL :

The cyberbullying detection framework consists of two key components, which are as follows:

1. Natural Language Processing (NLP)
2. Artificial Intelligence.

We have pulled real-time tweets from Twitter, Whatsapp chats, and YouTube comments in both English and Hinglish.Pre-processing text data is an important step in natural language processing to clean up and standardize the data. It's common to remove superfluous characters, such as emojis, special characters, and punctuation, as they can interfere with the analysis and modeling process., we must clean and prepare the data for detection before applying machine learning algorithms to it.During the preprocessing stage we have removed various unwanted patterns like hexadecimal patterns, stopwords, numerical Characters, Hashtags and other special symbols. It is accomplished by utilizing numpy and vectorize routines. We have manually come up with a list of stopwords in those languages that are taken and used it to delete them.

These terms should be removed from the clean data because their existence reduces the model's accuracy and predictions. We then used NLP techniques such as tokenization to break raw text into words known as tokens, lemmatization to reduce a given word to its core word, and vectorization to convert raw text into vectors or numbers.

After completion of pre-processing, we have divided the dataset into two parts such as testing data and training data. After that, we have used the two most significant text selection functions, which are:

1. Inverse frequency
2. Vectorizer of Counts

Based on the literature survey, we have used variety of machine learning algorithms like SVC, Decision Tree, , Bagging classifier, Naive Bayes, Logistic Regression, Random Forest and K Neighbours

Classifier, to train our model and to evaluate the accuracy for every model in the second phase. We used F1 score for evaluation and improved the accuracy of the model by recurring the various stages. We thought of finding the best suitable combination of dimension selection methods like and count vectorizers , TF-IDF and machine learning models. For completing this, we have compared both TF-IDF and count vectorizer; based on the comparison we got, we finalized the best pair with high accuracy and low prediction time and created its pickle file. After that, we fed the test data to the models inorder to compare the accuracy of various algorithms. using these our model will predict whether the text read belongs to the context of bullying or no in English.

i. Data collection

Our data set consists of text in both English and Hinglish. tweets in total, which were manually labeled as toxic or non-toxic by human annotators. For the Hinglish dataset, we collected real-time tweets and Whatsapp messages. The Hinglish dataset has approximately 9,482 entries in total, which were also manually labeled as toxic or non-toxic. networking platforms. It has approximately 15,307

We extracted tweets from various social media platforms like Twitter chats, and YouTube comments for Hinglish dataset. The dataset has approximately 3000 entries. We combined them to create a large dataset. Tweets and Label are the two columns of the dataset. The label contains the numbers 0, 1 , 2, which represent bulllying, non bullying and offensive respectively. The collection contains real samples of messages extracted from various social media platforms and networking websites. It has a large variety of negative words that are generally used by people in their messages. This would help us in detecting every offensive context of tweet. The next step is to preprocess the data after it has been extracted. It is performed because real-time data contains a large number of unwanted characters, necessitating cleaning of data in order to prepare the data for the detection . This is a time-consuming yet critical task.

ii. Data Preparation

Before running several ML models, the data must be cleaned, Cleaning the data and removing unnecessary characters, symbols, and stopwords is an essential step in preparing the data for ML models. It helps to improve the accuracy of the model and prevent overfitting. It's great to see that you have taken the necessary steps to clean the data and remove unwanted words and symbols. It's also important to note that using a manual list of stopwords can be useful, but there are also pre-built libraries and packages available that can automatically remove stopwords based on the language. These packages can save time and improve accuracy in the cleaning process.

iii. Preprocessing Methodologies

We used natural language processing techniques after cleaning the data because the algorithm cannot perform directly with unprepared text, that is, it cannot understand the sentences given to it, so we transform these sentences into understandable format using some preprocessing techniques. We use the following

• Tokenization - Tokenization is the process of dividing a text sequence into smaller parts, which include phrases, words, concepts, and symbols known as tokens. Tokenization is a fundamental step in natural language processing and is often performed as a pre-processing step before text data is fed into machine learning models. The goal of tokenization is to break down text into smaller units that can be analyzed and processed more easily, allowing for more accurate natural language processing tasks such as sentiment analysis, part-of-speech tagging, and text classification. Tokenization can be performed at different levels, including word-level, sentence-level, and document-level.

• Lemmatization - Lemmatization is the process of converting a word to its base or dictionary form, known as the lemma. It is an essential step in natural language processing that helps reduce the number of distinct words and variations in a text document. For example, the lemma of the word "running" is "run," and the lemma of the word "better" is "good."

• Vectorization - Vectorization is the process of converting text into numerical vectors that can be processed by machine learning algorithms. In NLP, vectorization is used to represent words or documents as numerical vectors, where each dimension of the vector corresponds to a specific feature. The values in the vector indicate the importance or weight of each feature.

.

iv. Data segmentation:

The dataset is divided into two types of data that is training data and testing data. For the model to be used in real time, the testing data must be retrieved from the various platforms using text mining. Both datasets are preprocessed and subjected to multiple ML models.

v. Feature choice:

We prepare the relevant text features after separating the data. This technique aids in determining the quality of the produced vector representations. This works with related words that tend to close with terms with varying degrees of resemblance. Vectorization is done before running the training and testing data sets through the ML models.

1) **Count Vectorization**: Count Vectorization is a technique in Natural Language Processing (NLP) that converts a collection of text documents into a matrix of token counts. In this technique, a count of each word occurrence in the text is calculated, and then the word occurrences are transformed into numerical values.

The process of Count Vectorization involves the following steps:

1. Tokenization: The text is first split into individual words or tokens.
2. Counting: The number of times each token appears in the document is counted.
3. Vectorizing: The counts are transformed into a sparse matrix of integers, where each row represents a document and each column represents a unique token in the corpus.

2) **TF-IDF**: TF-IDF stands for Term Frequency-Inverse Document Frequency. It is a numerical statistic that reflects how important a word is to a document in a collection or corpus. It is commonly used in natural language processing and information retrieval for text analysis and search engine ranking.

TF-IDF takes into account both the frequency of a word in a document and the frequency of the word in the entire corpus. The term frequency (TF) of a word is the number of times the word appears in a document, divided by the total number of words in the document. The inverse document frequency (IDF) of a word is calculated as the logarithm of the total number of documents in the corpus, divided by the number of documents that contain the word.

The TF-IDF score for a word in a document is calculated by multiplying the term frequency of the word in the document by the inverse document frequency of the word. Words with higher TF-IDF scores are considered to be more important to the document, and are therefore given higher weight in analysis and ranking.

1) SVM (Support Vector Machine): upport Vector Machines (SVMs) are a type of supervised learning algorithm used for classification and regression analysis. SVMs are particularly useful for classification tasks where the data points are not easily separable by a linear boundary.

In SVM, a hyperplane is used to separate the data points of different classes. The goal is to find the hyperplane that maximizes the margin between the closest data points of different classes. These closest points are known as support vectors. SVMs use a kernel function to transform the data into a higher-dimensional space, which can help to find a better separating hyperplane.

SVMs are known for their ability to handle high-dimensional data and perform well on small to medium-sized datasets. They have been successfully applied to a wide range of applications, including image classification, text classification, and bioinformatics.

2) KNN (K Neighbors): KNN (K Nearest Neighbors) is a non-parametric classification algorithm that works by finding the k-nearest data points to a given data point in the feature space. It then classifies the data point based on the majority class of its k-nearest neighbors. The choice of k is an important parameter, as a lower value of k may lead to overfitting while a higher value may result in underfitting. KNN is a simple yet effective algorithm for classification and can be used for both binary and multi-class classification problems.

.

3) Logistic Regression: Logistic regression is a popular machine learning algorithm used for binary classification problems, where the target variable has only two possible outcomes. In logistic regression, the goal is to find the best fitting S-shaped curve, also known as the sigmoid function, that maps the input

features to the target variable. The output of the sigmoid function is interpreted as the probability of the input belonging to the positive class.

The logistic regression model uses the maximum likelihood estimation technique to estimate the parameters of the sigmoid function. The model tries to find the values of the coefficients that maximize the likelihood of the observed data given the input features. The optimization problem is solved using gradient descent or other optimization algorithms.

Logistic regression is a simple and efficient algorithm that is easy to interpret and has low computational overhead. It can handle non-linear relationships between the input features and the target variable by adding polynomial or interaction terms to the model. However, logistic regression assumes that the input features are independent and linearly related to the target variable, which may not always be the case in practice.

4) Decision Tree Learning: In research, decision trees can be a useful approach for detecting cyberbullying. A decision tree is a form of supervised learning technique that may be used to categorize data into several groups. Decision trees can be used in cyber bullying detection to examine aspects of online content (such as text, photos, or videos) and classify them as instances of cyber bullying or non-cyber bullying. If the model's performance is inadequate, you may need to refine it by adjusting its settings or redefining its characteristics. This iterative procedure will assist you in developing a decision tree model that is successful at detecting cyberbullying.

Overall, decision trees can be a valuable technique for detecting cyberbullying in studies. Decision trees can help identify instances of cyberbullying and aid in the creation of effective intervention techniques by examining aspects of online content.

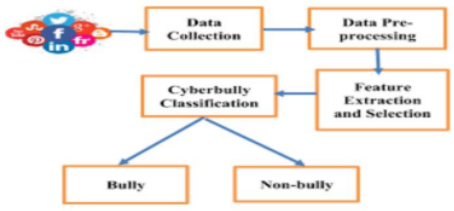


Fig.3.Model

3.4 DATASET DESCRIPTION:

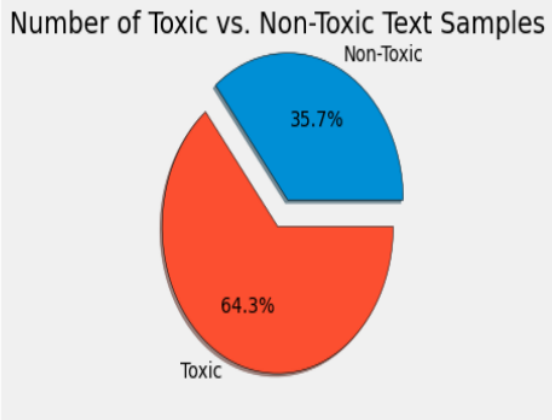
The dataset is the snapshot of the twitter information d uring a particular period of time .

It consists of tweets from all kinds of users .It has seven columns na--mely 'Unnamed,'count', 'hate_spe ech' , 'offensive_language ', 'neither', 'class', 'tweet '.

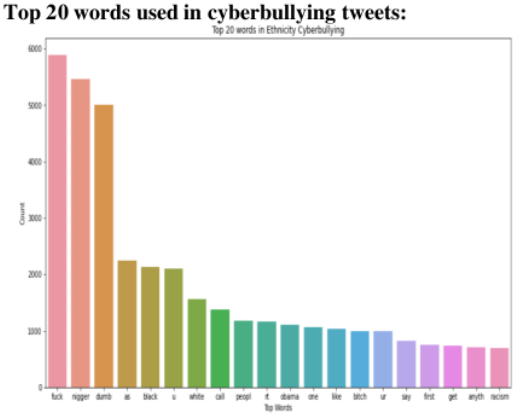
The class is our dependent variable and remaining are the independent varibles .The class consists of nu meric values so we would be converting them into dis crete values by labelling them.The dataset have nearly 24000 rows .we would be partitioning it for testing a nd training .

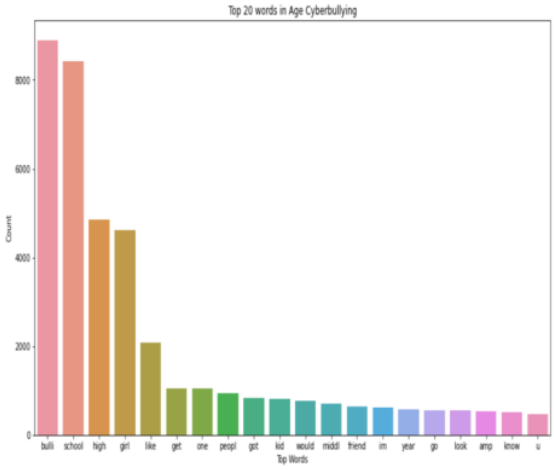
It does not have any missing values or inconsistent values .

The tweets is the most important variable as we are g oing to use it for classifying the comments .The clas s is our target variable one which will be predicted.



Top 20 words used in cyberbullying tweets:





VII EXPERIMENTS AND RESULTS

IMPORTING LIBRARIES:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.svm import SVC,LinearSVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.pipeline import Pipeline
import pickle
```

IMPORTING AND DOWNLOADING NLP TOOLS :

```
import re
import nltk
nltk.download('stopwords')
from nltk.util import pr
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words("english"))
```

LOADING DATASET :

```
ds=pd.read_csv("twitter_data.csv")
print(ds.head())
ds.describe()
ds.columns
len(ds.index)
```

```
ds['labels']=ds['class'].map({0:"Hate Speech Detected", 1:"cyber Bullying detected", 2:"no hate and offensive speech"})
print(ds.head())
```

Here, we would be converting the target variable into a discrete variable as we would be working with decision tree classifier.

```
ds=ds[['tweet','labels']]
ds.head()
```

Here, we would be visualizing the changes we have made.

USER DEFINED FUNCTIONS:

```
def clean(text):
    text=str(text).lower()
    text=re.sub('\[.*?\]','',text)
    text=re.sub('https?://\S+|www\.\S+','',text)
    text=re.sub('<.*?>+','',text)
    text=re.sub('[%s]'% re.escape(string.punctuation),'',text)
    text=re.sub('\n','',text)
    text=re.sub('\w*\d\w*','',text)
    text=[word for word in text.split(' ')if word not in stopword]
    text=" ".join(text)
    text=[stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
ds["tweet"]=ds["tweet"].apply(clean)
print(ds.head())
```

The inbuilt function clean is used to remove the unwanted gibberish or symbols from the tweets that are in the dataset.

Function to remove emoji's in text:

```
In [ ]: def strip_emoji(text):
            return emoji.replace_emoji(text,replace="")
```

SPLITTING THE DATASET:

```
x=np.array(ds["tweet"])
y=np.array(ds["labels"])
cv=CountVectorizer()
x=cv.fit_transform(x)
x_train, x_test, y_train, y_test=train_test_split(x,y, train_size=0.73, random_state=0)
```

CREATION AND TRAINING OF DIFFERENT MODELS:

i) Decision Tree Classifier

```
In [37]: dtc = DecisionTreeClassifier()
         dtc.fit(X_over, y_over)
         y_pred = dtc.predict(X_test)
         print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
         print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
         getStatsFromModel(dtc)
```

```
Accuracy:  0.8455386153461635
Confusion Matrix:
 [[1883  546]
 [  72 1500]]
              precision    recall  f1-score   support

           0       0.96      0.78      0.86      2429
           1       0.73      0.95      0.83      1572

    accuracy                           0.85      4001
   macro avg       0.85      0.86      0.84      4001
weighted avg       0.87      0.85      0.85      4001
```

ii) Support Vector Machine

```
In [69]: lin_svc = LinearSVC()
```

```
In [70]: lin_svc_cv_score = cross_val_score(lin_svc,X_train_tf,y_train,cv=5,scoring='f1_macr
         mean_lin_svc_cv = np.mean(lin_svc_cv_score)
         mean_lin_svc_cv
```

```
Out[70]: 0.8220066371295554
```

```
In [32]: gnb = GaussianNB()
         gnbmodel = gnb.fit(X_over, y_over)
         y_pred = gnbmodel.predict(X_test)
         print ("Score:", gnbmodel.score(X_test, y_test))
         print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
         getStatsFromModel(gnb)
```

```
Score: 0.6160959760059985
Confusion Matrix:
 [[ 924 1505]
 [  31 1541]]
              precision    recall  f1-score   support

           0       0.97      0.38      0.55      2429
           1       0.51      0.98      0.67      1572

    accuracy                           0.62      4001
   macro avg       0.74      0.68      0.61      4001
weighted avg       0.79      0.62      0.59      4001
```

iii) Naïve Bayes

iv) Logistic Regression:

```
In [34]: lgr = LogisticRegression()
         lgr.fit(X_over, y_over)
         y_pred = lgr.predict(X_test)
         print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
         print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
         getStatsFromModel(lgr)
```

Logistic Regression Confusion matrix:

```
Accuracy:  0.8007998000499875
Confusion Matrix:
 [[1907  522]
 [ 275 1297]]
              precision    recall  f1-score   support

           0       0.87      0.79      0.83      2429
           1       0.71      0.83      0.76      1572

    accuracy                           0.80      4001
   macro avg       0.79      0.81      0.80      4001
weighted avg       0.81      0.80      0.80      4001
```

TESTING THE MODEL :

```
test_data="your work sucks"
ds=cv.transform([test_data]).toarray()
print(clf.predict(ds))
```

RESULT

```
from sklearn import metrics
y_pred = clf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8734309623430963
```

```
clf=DecisionTreeClassifier()
clf.fit(x_train,y_train)
```

```
DecisionTreeClassifier()
```

```
test_data="your work sucks"
ds=cv.transform([test_data]).toarray()
print(clf.predict(ds))
```

```
['cyber Bullying detected']
```

```
from sklearn import metrics
y_pred = clf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8734309623430963
```

VIII CONCLUSION AND DISCUSSION

A. CHALLENGES FACED:

1) Long Execution time: Running all of the 450 experiments in the first set was challenging since it required a huge amount of time to complete. To overcome the slow execution time of the models, the experiments were conducted in parallel on high-performance compute nodes in Compute Canada clusters.

2) Random under sampling*:* The use of random under sampling had a negative effect on model's performance. This can be due to the huge loss of information that happens when using this under sampling method. It is possible to experiment with over sampling techniques in the future to overcome this issue.

FUTURE WORK:

Along with the laws that are used to punish those people who cause cyber violence having an system that automatically detects the context of cyberbullying and changes it into a positive comment will be of great help as the saying goes prevention is the best .Using this will prevent lot of people from depression ,low self-esteem and also suicides.it also never let the users to use the social media as a tool to humiliate or bully others .The cyberbullying detecting system will lead to healthy environment on social media .It can be embedded into all social media and messaging apps.

IX REFERENCES

RP-2

ORIGINALITY REPORT

22%	16%	6%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijert.org Internet Source	6%
2	Submitted to Harrisburg University of Science and Technology Student Paper	1%
3	Submitted to Santa Monica College Student Paper	1%
4	Al-garadi, Mohammed Ali, Kasturi Dewi Varathan, and Sri Devi Ravana. "Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network", Computers in Human Behavior, 2016. Publication	1%
5	Submitted to London School of Commerce Student Paper	1%
6	Submitted to University of Bradford Student Paper	1%
7	medium.com Internet Source	1%

8	Submitted to National Institute of Technology Karnataka Surathkal Student Paper	1 %
9	"Data Management, Analytics and Innovation", Springer Science and Business Media LLC, 2023 Publication	<1 %
10	Submitted to Middlesex University Student Paper	<1 %
11	sparkbyexamples.com Internet Source	<1 %
12	Submitted to RMIT University Student Paper	<1 %
13	Submitted to University of Reading Student Paper	<1 %
14	www.ncbi.nlm.nih.gov Internet Source	<1 %
15	Submitted to CSU, San Jose State University Student Paper	<1 %
16	Submitted to Intercollege Student Paper	<1 %
17	towardsdatascience.com Internet Source	<1 %
18	www.bloggersideas.com Internet Source	<1 %

19	Submitted to Nottingham Trent University Student Paper	<1%
20	Submitted to University of Warwick Student Paper	<1%
21	Submitted to Infile Student Paper	<1%
22	Submitted to Liverpool John Moores University Student Paper	<1%
23	Wai Ming Wang, Eric Wing Kuen See-To, Hong Tao Lin, Zhi Li. "Comparison of Automatic Extraction of Research Highlights and Abstracts of Journal Articles", Proceedings of the 2nd International Conference on Computer Science and Application Engineering - CSAE '18, 2018 Publication	<1%
24	Submitted to Nanyang Technological University Student Paper	<1%
25	Submitted to University of Essex Student Paper	<1%
26	Submitted to Coventry University Student Paper	<1%
27	Submitted to Staffordshire University Student Paper	<1%

28	Submitted to UC, Boulder Student Paper	<1 %
29	Submitted to University of Sunderland Student Paper	<1 %
30	www.researchgate.net Internet Source	<1 %
31	Submitted to National University of Ireland, Galway Student Paper	<1 %
32	Submitted to University of Ulster Student Paper	<1 %
33	ebin.pub Internet Source	<1 %
34	www.freepatentsonline.com Internet Source	<1 %
35	Submitted to University of London External System Student Paper	<1 %
36	www.fireblazeaischool.in Internet Source	<1 %
37	www.semanticscholar.org Internet Source	<1 %
38	Submitted to University of Bedfordshire Student Paper	<1 %

39	www.mdpi.com Internet Source	<1 %
40	KAYE. "Ebook: Issues and Debates in Cyberpsychology", Ebook: Issues and Debates in Cyberpsychology, 2021 Publication	<1 %
41	Submitted to Leiden University Student Paper	<1 %
42	www.icsd.aegean.gr Internet Source	<1 %
43	cpentalk.com Internet Source	<1 %
44	link.springer.com Internet Source	<1 %
45	mafiadoc.com Internet Source	<1 %
46	www.marinha.mil.br Internet Source	<1 %
47	Han Liu, Pete Burnap, Wafa Alorainy, Matthew L. Williams. "A Fuzzy Approach to Text Classification With Two-Stage Training for Ambiguous Instances", IEEE Transactions on Computational Social Systems, 2019 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		