Cyber Bullying Detection Using Machine Learning Techniques

Dr. M. Sakthivel
Professor, Department of CSE
Sree Vidyanikethan Engineering College,
Tirupati, Andhra Pradesh, India

V. Nagesh
UG Scholar, Department of CSE
Sree Vidyanikethan Engineering College,
Tirupati, Andhra Pradesh, India

V.Ganga Pradeep,
UG Scholar, Department of CSE
Sree Vidyanikethan Engineering College,
Tirupati, Andhra Pradesh, India

V. Hemanth Kumar Reddy,
UG Scholar, Department of CSE
Sree Vidyanikethan Engineering College,
Tirupati, Andhra Pradesh, India

V. Sindhura
UG Scholar, Department of CSE
Sree Vidyanikethan Engineering College,
Tirupati, Andhra Pradesh, India

T. Chirudeep
UG Scholar, Department of CSE
Sree Vidyanikethan Engineering College,
Tirupati, Andhra Pradesh, India

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. Research suggests that approximately 50% of children have been affected by cyberbullying, resulting in a range of negative consequences such as mental health issues, academic struggles, and even suicidal thoughts. Unfortunately, many victims do not receive the help and support they need due to factors such as social status. In addition to the emotional toll, cyberbullying can also damage a person's reputation through the spread of false rumors and harmful content. 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.

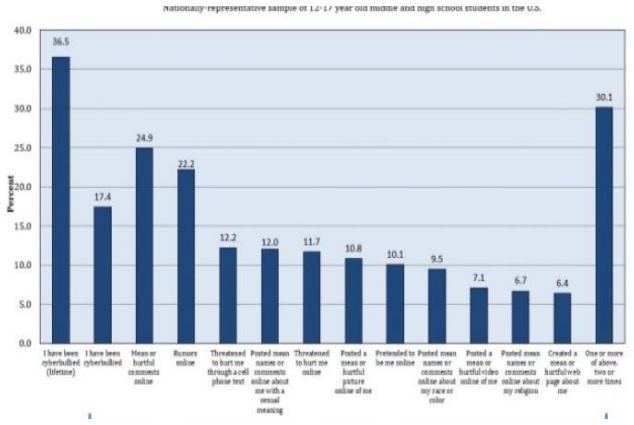


Fig.1.CyberBullying Victimization and Reasons

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 proposed work will utilize the supervised learning approach to detect and address cyberbullying. The dataset will consist of two categories of data, positive and negative, related to cyberbullying. The goal is to identify instances of cyberbullying and transform negative comments into positive ones. The work will involve implementing multiple machine learning algorithms such as Random Forest, decision tree learning, Linear Regression, and 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

The field of Machine Learning offers diverse algorithms that aid in the creation of predictive models for various applications, including the detection of cyberbullying. With the increasing use of social media platforms, cyberbullying has become prevalent, necessitating the need for models that can recognize such instances in real-time, given the constant growth of data. This can be achieved by utilizing machine learning techniques such as Naive Bayes, Decision Tree, and Random Forest, as well as ensemble algorithms that 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.

Chen et al. (2012) conducted a study that developed a method for detecting offensive language using a syntactic element which outperformed the standard learning-based method. In a similar study, Dadvar et al. (2013) utilized SVM to detect cyberbullying in YouTube data and found that integrating user-generated content improved the accuracy of SVM detection. Dadvar et al. used MySpace data sets to reach their conclusions.

The research conducted by Cynthia Van Hee and Gilles Jacobs focused on automatic cyberbullying identification in social media text. They gathered data from social networking sites and used online bullying standards to create corpora for modeling postings made by bullies and victims. Common natural language processing pre-processing steps such as tokenization, PoS-tagging, and lemmatization were utilized to convert unstructured text into a structured format that can be used for machine learning algorithms. The researchers also tested language conversion by constructing models for both English and Dutch. Although the accuracy results of 64% for English and 61% for Dutch were modest, they serve as a promising starting point for further research and improvements to the SVM ML algorithm. The complexity and diversity of language used in social media communication make automatic cyberbullying identification in social media text a challenging task.

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.

The study by S.E. Vishwapriya, Ajay Gour, and their team on detecting hate speech and objectionable language on Twitter using machine learning is noteworthy. The researchers utilized labeled datasets from Crowdflower and GitHub, which had categories such as Hateful, Offensive, Sexism, and Racism. They pre-processed the data by performing various text cleaning tasks, including lowercasing tweets, removing URLs, stop words, and Twitter mentions, and using stemming. They also divided the dataset into 70% training and 30% test samples, which is a standard practice in machine learning. The team employed N-gram features with TF-IDF weighting, which is a popular approach for text classification. To identify the most suitable algorithm for the task, they compared the performances of Logistic Regression, Naive Bayes, and Support Vector Machine. The results showed that Logistic Regression with L2 Normalization and n=3 had an impressive accuracy of

95%. These findings suggest that their approach has great potential 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.

In their research, Lida Ketsbaia, Biju Issac, and colleagues focused on identifying hateful and harmful tweets using machine learning and deep learning. One notable aspect of their approach was the use of datasets from two different universities, with one labeled as "Hate" and the other labeled as "Non-Hate," to train their model. This allowed them to create a more diverse and comprehensive dataset for their research.

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:

To build a machine learning model for identifying cyberbullying, the first step is to gather a comprehensive and varied dataset of text that includes examples of both cyberbullying and non-cyberbullying. This dataset can be obtained from social media platforms, online forums, and comments sections. The aim is to ensure that the dataset is large and diverse, with a wide range of texts representing different contexts and scenarios. This will allow the model to learn and generalize from the data effectively, resulting in better performance.

Data Preprocessing:

In the data preprocessing step, the collected data is cleaned and standardized to prepare it for machine learning algorithms. This involves removing irrelevant information like URLs and special characters that may interfere with the analysis process. Additionally, text normalization techniques like converting all text to lowercase may be applied to ensure consistency in the data. The goal of data preprocessing is to obtain a clean dataset that can be used for feature engineering and model training.

Feature Engineering:

Feature engineering is a crucial step in building a machine learning model for text classification. It involves selecting the features that the model will use to make its predictions. In the case of text, this typically involves creating numerical representations of the text that capture the most important information. This can be achieved through techniques such as term frequency-inverse document frequency (TF-IDF) or word embeddings, which convert the text into numerical vectors. These features are then used as input to the machine learning algorithm for training and prediction.

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:

After building and fine-tuning the model, it's essential to evaluate its performance. The evaluation step involves testing the model on a separate test set and computing metrics such as accuracy, precision, recall, and F1 score. These metrics help to assess the model's effectiveness in detecting cyberbullying. By evaluating the model's performance, we can identify potential areas for improvement and fine-tune the model further to enhance its accuracy and effectiveness in detecting cyberbullying.

Model Fine-tuning:

Fine-tuning the model involves adjusting its parameters or testing different algorithms to improve its performance, and repeating the evaluation step if necessary. This step is crucial as it helps to optimize the model and improve its accuracy. Depending on the results of the evaluation, the model may need to be fine-tuned by adjusting its parameters or trying a different algorithm to achieve better results. By repeating the evaluation step, we can assess whether the changes made to the model have improved its performance. Overall, fine-tuning is a necessary step in the model development process to ensure that the model is as accurate and effective as possible.

B CYBER BULLYING DETECTION MODEL :

The cyberbullying detection framework consists of two key components: Natural Language Processing (NLP) and Artificial Intelligence. To train our model, we collected real-time tweets from various social media platforms such as Twitter, WhatsApp, and YouTube comments in both English and Hinglish. Pre-processing text data is an essential step in NLP to standardize the data and remove unwanted characters such as emojis, special characters, and punctuation that may interfere with the analysis and modeling process. Therefore, we cleaned and prepared the data for detection by removing unwanted patterns such as

numerical characters, hashtags, and special symbols using numpy and vectorize routines. We also manually created a list of stopwords in those languages and used it to delete them from the clean data since their existence can reduce the model's accuracy and predictions.

We then utilized NLP techniques such as tokenization, lemmatization, and vectorization to transform raw text into numerical vectors. After completing pre-processing, we divided the dataset into two parts: testing data and training data. We used two significant text selection functions, Inverse Frequency and Count Vectorizer, to train our model. We also employed a variety of machine learning algorithms such as SVC, Decision Tree, Bagging Classifier, Naive Bayes, Logistic Regression, Random Forest, and K Neighbours Classifier to evaluate the accuracy for each model in the second phase. We used the F1 score for evaluation and improved the accuracy of the model by iterating through the various stages.

To find the best suitable combination of dimension selection methods such as TF-IDF and count vectorizers and machine learning models, we compared both TF-IDF and count vectorizer. Based on the comparison, we selected 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 to compare the accuracy of various algorithms. Using these techniques, our model can predict whether the text belongs to the context of bullying or not in English.

i. Data collection

We collected a dataset consisting of both English and Hinglish text, which contained tweets from various social media platforms and networking websites. The dataset consisted of approximately 15,307 entries in English, and for the Hinglish dataset, we extracted tweets and WhatsApp messages in real-time, resulting in approximately 9,482 entries in total. Human annotators manually labeled all entries as toxic or non-toxic. In addition, we extracted tweets and comments from YouTube and Twitter chats, which were combined to create a larger dataset of approximately 3000 entries. The dataset contains two columns: Tweets and Labels, with labels representing bullying, non-bullying, and offensive messages with numbers 0, 1, and 2, respectively. This diverse collection of messages helps in detecting offensive contexts in tweets effectively. However, since real-time data often contains unwanted characters and symbols, preprocessing the data is a crucial yet time-consuming step to prepare it for the detection process.

ii. Data Preparation

In order to prepare the data for machine learning models, it is crucial to clean the data by removing unnecessary characters, symbols, and stopwords. This step can significantly enhance the accuracy of the model and prevent overfitting. It is commendable that you have already taken the necessary steps to clean the data and remove unwanted words and symbols. Additionally, it is worth noting that while a manual list of stop words can be helpful, there are also pre-built libraries and packages that can automatically remove stop words based on the language. These packages can be time-saving and further improve the accuracy of 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 refers to the process of breaking down a text sequence into smaller units or tokens such as words, phrases, concepts, and symbols. It is a crucial step in natural language processing and is typically performed as a pre-processing step before feeding text data into machine learning models. Tokenization enables the efficient analysis and processing of text by dividing it into smaller units that can be processed more easily, leading to more accurate natural languages processing tasks like sentiment analysis, part-of-speech tagging, and text classification. Tokenization can be performed at various levels, including word level, sentence-level, and document level.

• Lemmatization - Lemmatization is a crucial process in natural language processing that involves transforming a word into its base or dictionary form, which is known as its lemma. This process is important because it helps to decrease the number of distinct words and variations present in a text document, thus improving the accuracy of text analysis. An example of lemmatization is converting

the word "running" to its lemma "run," and the word "better" to its lemma "good."

• Vectorization - Vectorization is a technique used in natural language processing to convert text data into numerical vectors that can be processed by machine learning algorithms. This process involves representing words or documents as numerical vectors, where each dimension of the vector represents a particular feature. The values in the vector correspond to the importance or weight of each feature, enabling the algorithm to understand the text data and make predictions based on it.

iv. Data segmentation:

In the process of building a machine learning model for text analysis, it is necessary to divide the available data into two categories, namely training data and testing data. The testing data is obtained by extracting text data from various sources using techniques such as text mining. Both the training and testing data go through preprocessing steps and are then used for training and evaluating multiple machine learning models. This segmentation of data helps in ensuring that the developed model performs well on unseen data and can be used effectively in real-world applications.

v. Feature choice:

Feature selection is an essential step in natural language processing (NLP) that involves extracting relevant text features after segregating the data. This method helps to assess the accuracy of the resulting vector representations. It operates by identifying related words that often appear near terms with different degrees of similarity. Vectorization is then applied to convert the text features into numerical vectors before feeding the data into the machine learning models for training and testing.

1) Count Vectorization: Count Vectorization is a popular technique in Natural Language Processing (NLP) that converts a collection of text documents into a matrix of token counts. This technique involves breaking down the text into individual words or tokens, counting the frequency of each token in the document, and then converting the counts into numerical values. The result is a sparse matrix of integers, where each row represents a document and each column represents a unique token in the corpus. Count Vectorization is a fundamental step in many NLP applications, such as document classification, sentiment analysis, and topic modeling.

2) **TF-IDF**: TF-IDF, an acronym for Term Frequency-Inverse Document Frequency, is a numerical metric that measures the relevance of a word in a document within a corpus. This metric is frequently employed in natural language processing and information retrieval to conduct text analysis and ranking in search engines. The importance of a word is determined by both its frequency in the document and the frequency of the word in the corpus. The term frequency (TF) of a word in a document is determined by the number of times the word appears in the 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 importance of a word is determined by multiplying the TF and IDF scores, and words with higher TF-IDF scores are deemed more significant in the document and given greater weight in analysis and ranking.

1) SVM (Support Vector Machine): Support Vector Machines (SVMs) are a popular supervised learning algorithm that is widely used for classification and regression analysis. SVMs are particularly useful when the data points are not easily separable by a linear boundary. Instead, SVMs use a hyperplane to separate the data points of different classes.

The primary objective of SVM is to find the hyperplane that maximizes the margin between the closest data points of different classes. These closest data points are referred to 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.

One of the major strengths of SVMs is their ability to handle high-dimensional data and perform well on small to medium-sized datasets. They have been successfully employed in numerous applications, including image classification, text classification, and bioinformatics. SVMs are a powerful algorithm that has gained popularity due to its flexibility and strong performance in a wide range of applications.

2) KNN (K Neighbors): KNN, short for K Nearest Neighbors, is a non-parametric classification algorithm that operates by identifying the k-nearest data points to a given data point in the feature space. Based on the majority class of its k-nearest neighbors, the algorithm then classifies the data point. The value of k is a

critical parameter in the algorithm, as a low k value may cause overfitting, while a high k value may lead to underfitting. KNN is a straightforward yet effective algorithm suitable for binary and multi-class classification problems..

3) Logistic Regression: Logistic regression is a well-known machine learning technique employed for binary classification problems where the outcome variable has only two potential results. The primary objective of logistic regression is to determine the most appropriate S-shaped curve, or sigmoid function, that maps input features to the target variable. The sigmoid function's output represents the likelihood of the input belonging to the positive class.

The logistic regression model uses the maximum likelihood estimation approach to compute the sigmoid function's parameters. The model aims to identify the coefficient values that maximize the probability of the input features given the observed data. This optimization problem is resolved utilizing various optimization algorithms such as gradient descent.

Logistic regression is an effective algorithm that is simple to interpret and has low computational requirements. The algorithm can handle non-linear relationships between the input features and the target variable by including polynomial or interaction terms into the model. However, logistic regression presupposes that the input features are independent and linearly associated with the target variable, which may not always be valid in real-world applications.

4) Decision Tree Learning: In the field of research, decision trees have proven to be a useful method for detecting instances of cyberbullying. This supervised learning technique involves categorizing data into multiple groups based on certain criteria. Decision trees can be applied in cyberbullying detection to analyze various forms of online content, including text, photos, and videos, and determine whether they constitute cyberbullying or not. In case the model's performance is not satisfactory, one can adjust its settings or redefine its characteristics to enhance its accuracy. By repeating this process, a more effective decision tree model for detecting cyberbullying can be developed. 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.

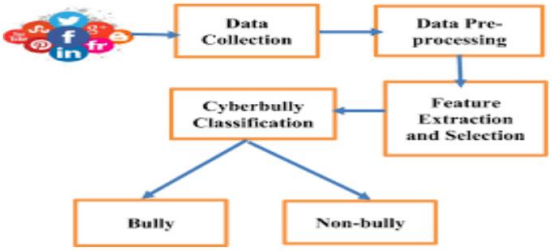


Fig.3.Model

3.4 DATASET DESCRIPTION:

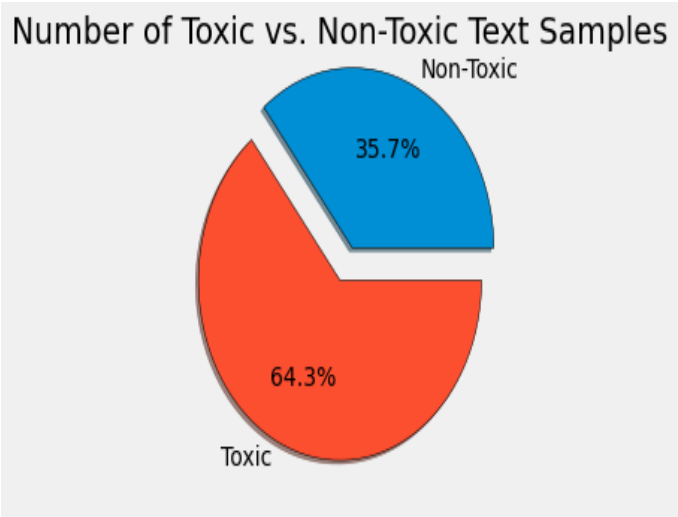
The dataset is the snapshot of the twitter information d uring a particular period of time .

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 '.

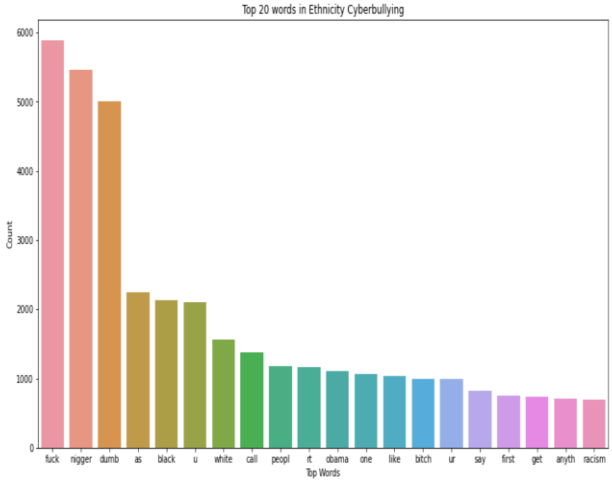
The class is our dependent variable and remaining are the independent varibles .The class consists of nu meric values so we would be converting them into dis crete values by labelling them.The dataset have nearly 24000 rows .we would be partitioning it for testing a nd training .

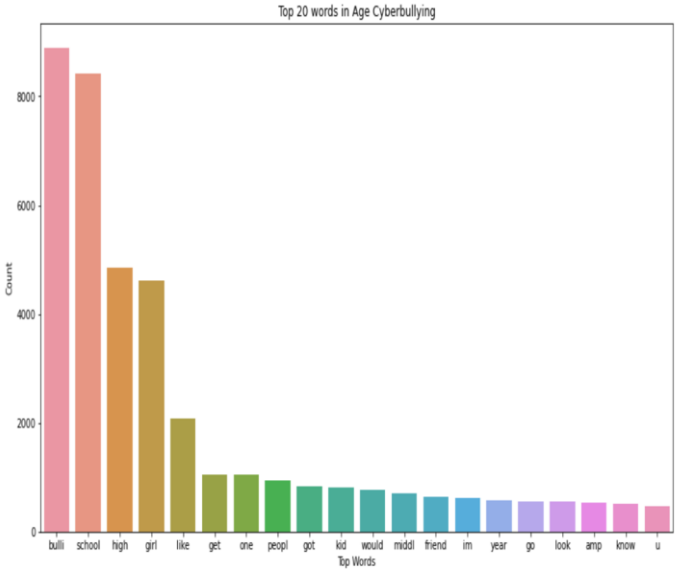
It does not have any missing values or inconsistent values .

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

[1] Vimala Balakrishnan Ph.D , Shahzaib Khan , Hamid R. Arabnia Ph.D , Improving Cyberbullying Detection using Twitter Users' Psychological Features and Machine Learning, Computers & Security (2019), doi: https://doi.org/10.1016/j.cose.2019.101710

[2] Desai, Aditya, et al. "Cyber Bullying Detection on Social Media using Machine Learning." *ITM Web of Conferences*. Vol. 40. EDP Sciences, 2021.

[3] H. Rosa, N. Pereira, R. Ribeiro, P.C. Ferreira, J.P. Carvalho, S. Oliveira, L. Coheur, P. Paulino, A.M. Veiga Simão, I. Trancoso, Automatic cyberbullying detection: A systematic review., Computers in Human Behavior (2018), doi: 10.1016/j.chb.2018.12.021.

[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.