DETECTION OF CYBERBULLYING

Mini Project Report

Submitted by

SREETHU T NAIR

Reg NO: FIT20MCA2107

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications

Of

A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY-683577, ERNAKULAM(DIST)

MARCH 2022

DECLARATION

I, **SREETHU T NAIR**, hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date : 04-03-2022

Place: Angamaly

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

<u>CERTIFICATE</u>

This is to certify that the project report titled **”DETECTION OF CYBERBULLYING”** submitted by **SREETHU T NAIR** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by them during the year 2022.

Project Guide **Head of the Department**

Submitted for the viva-voice held on . . . . at . . . .

Examiner1 :

Examiner2 :

ACKNOWLEDGEMENT

I am extremely glad to present my mini project which I did as a part of our curriculum. I take this opportunity to express my sincere thanks to those who helped me in bringing out the report of my project.

I am deeply grateful to Dr. George Manoj George, Principal, FISAT, Angamaly and Dr. C. Sheela, Vice Principal, FISAT, Angamaly.

My sincere thanks to Dr.Deepa Mary Mathews, Head of the department of MCA, FISAT, who had been a source of inspiration. During the period of my project work, I have received generous help from Ms. Rose Mary Mathew, my project guide, which I like to put on record here with deep gratitude and great pleasure.

I express my heartfelt thanks to all the faculty members in my department for their constant encouragement and never ending support throughout the project.

Finally I am grateful to all my friends who gave me a lot of suggestions for the successful completion of this project.

ABSTRACT

Cyberbullying is totally different from conventional harassing, however it is as yet tormenting. The results and risks continue as before, if not extended in their seriousness and span. Despite the fact that it happens through online sites rather than face to face, cyber bullying should be viewed as appropriately. At the appropriate time, cyberbullying comes in different various structures. It doesn't really mean hacking somebody's profile or presenting to be another person. It likewise incorporates posting negative remarks about someone or spreading bits of hearsay to criticize somebody. Cyberbullying or Social Media Bullying incorporates activities and measures to control, annoy or stigmatize any individual. These horrible activities are solemnly harming and can influence anybody effectively and seriously. They basically happen via web-based media, public gatherings, and other online sites.

The main objective of the proposed system is to detect cyber bullying that occurs on various tweets from twitter. The data on these sites is present in the form of tweets and comments from different online users. Next Step is Data preprocessing which means we will process our data before feeding it into our machine. In Data preprocessing, we first remove any irrelevant data from our dataset, then we treat outliers and lastly we handle any missing data which may be present in our dataset. After splitting our dataset into training and testing dataset, we will train our machine using this training set which will help our classifier algorithm in learning to classify the data into positive and negative tweets/comments.

Contents

Chapter 1

INTRODUCTION

Cyberbullying is totally different from conventional harassing, however it is as yet tormenting. The results and risks continue as before, if not extended in their seriousness and span. Despite the fact that it happens through online sites rather than face to face, cyber bullying should be viewed as appropriately. At the appropriate time, cyberbullying comes in different various structures. It doesn't really mean hacking somebody's profile or presenting to be another person. It likewise incorporates posting negative remarks about someone or spreading bits of hearsay to criticize somebody. Cyberbullying or Social Media Bullying incorporates activities and measures to control, annoy or stigmatize any individual. These horrible activities are solemnly harming and can influence anybody effectively and seriously. They basically happen via web-based media, public gatherings, and other online sites.

Cyberbullying is a planned and repetitive act to harm or humiliate a person using information and communication technologies, including e-mails and social media. It is categorized into various forms, like cyber harassment (repetitively harassing and threatening someone), denigration/slandering (sharing false information about someone), flaming (brief insulting online interactions), etc. Since the physical appearance of the bully is not required, it can go on nonstop. With the increasing adverse impact of cyberbullying on society, it's necessary to seek out ways to detect this phenomenon. Automatically recognizing emojis, bully words, and audio features from online social platforms, especially micro-blogging sites like Twitter, facebook and video-sharing platforms like YouTube is vital to research. The process of detecting cyberbullying activities begins with input datasets from social networking sites.

The input dataset consists of text comments and messages published on twitter. Data pre-processing is performed on input data to enhance the quality of the research data. Subsequent analytical steps include removing extra characters, stop words, and hyperlinks. Feature extraction is done after completing pre-processing on the given input data. Features like Pronoun, Noun, and Adjective from the text are obtained by feature extraction and the frequency of words in the text is determined. The extracted features are then given as input to the Classification Algorithm. Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem The output of this classification algorithm shows if the given message contains bullying words or not. This is a supervised approach because it uses a tagged or labeled training data set.

Chapter 2

PROOF OF CONCEPT

Objectives

The main objective is to detect the Cyberbullying words, which is a classic text classification problem with a straight forward proposition. We need to identify whether the text is bullying or not. Our study explores different textual properties that could be used to distinguish bullying words. By using those properties, we train a combination of different machine learning algorithms using various machine learning methods that are not thoroughly explored in the current literature.

Here we focused on a particular area for detecting cyberbullying. Includes the comparison of various previous methodologies proposed using different datasets and with different characteristics and accomplishments. Time saving Accuarcy

Chapter 3

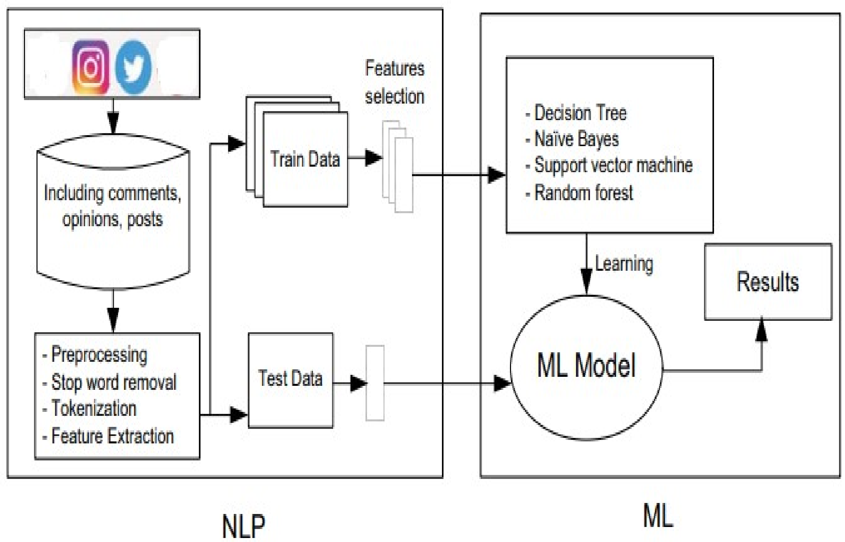
IMPLEMENTATION

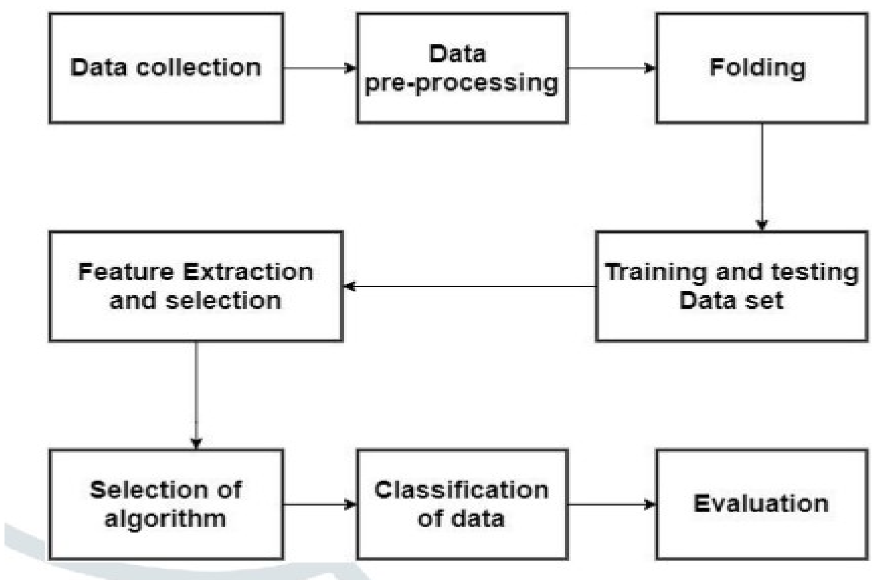
The dataset we took is a cyberbullying from twitter dataset from Kaggle.com. Shape of dataset (20,000). We have isolated the label column from the data frame. Preprocessing steps are applied and then we split the dataset into train and test data set. Feature are extracted using tfidf vectorizer. Feature extraction is done after completing pre-processing on the given input data. Features like Pronoun, Noun, and Adjective from the text are obtained by feature extraction and the frequency of words in the text is determined. The extracted features are then given as input to the Classification Algorithm. Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem The output of this classification algorithm shows if the given message contains bullying words or not. This is a supervised approach because it uses a tagged or labeled training data set.

3.1 TOOLS OR PROGRAMMING LANGUAGE

- FRONT END :
  - Html, CSS
- BACK END :
  - Python

3.2 ARCHITECTURE





3.3 MODULES

3.3.1 Data Preprocessing :

The data collected had to be preprocessed since it had traces of unstructured contents. It basically meant we needed to clean or trim the data in order to obtain a higher accuracy. There were various steps that were needed to be followed for preprocessing the data such as data cleaning, stop word removal, tokenization. With the help of a stop word filter we deleted any needless words on all the text conversation in line with the English vocabulary. The term stop words mean those words that don't give any helpful data to decide in which category a text should be classified. For facilitating the further processes with the motive of not distinguishing among capital letters and lowercase letters, we transformed the whole data into lower case. Furthermore tokenization had to be practiced on these text contents to facilitate the feature extraction step. Tokenization can be defined as a way of separating or isolating every word that compiles in a document or even a conversation.

3.3.2 Feature Extraction :

A primary concern of ours was to examine the effect of feature set size on text categorization effectiveness. All potential features were ranked for each category by expected mutual information between assignment opposite effects on the properties of a text representation, which led us to investigate combining the two techniques. However, the small size of standard text retrieval test collections, and the variety of approaches available for query interpretation, made it difficult to study purely representational issues in text retrieval experiments. In this work we examine indexing language properties using two text categorization data sets. We obtain much clearer results, as well as producing a new text categorization method capable of handing multiple, overlapping categories.

Clustering features requires defining a set of meta features on which the similarity of the features will be judged. We experimented with forming clusters from words under three meta feature definitions, and from phrases under eight meta feature definitions. Meta features were based on presence or absence of features in documents, or on the strength of association of features with categories of documents. In all cases, similarity between meta feature vectors was measured using the cosine correlation. The sets of clusters formed were examined by the author, and categorization experiments were run with the three sets of word clusters and with the two sets of phrase clusters that appeared best. The proposed method is an unsupervised feature selection method and evaluates the discriminative power of terms in group form. The motivation behind this approach is that it is probable that some terms have low discriminative power for clustering but when they form a group, they may have good discriminative power. The reason is that when some correlated terms form a group together they represent a concept and this is what we are looking for in a clustering task. The clustering algorithm best known unsupervised learning algorithms that solve the well-known clustering problem. The clustering algorithm aims to partition a set of objects, based on their attributes/features, into k clusters, where k is a predefined or user-defined constant. The main idea is to define k centroids, one for each cluster. The centroid of a cluster is formed in such a way that it is closely related (in terms of similarity function; similarity can be measured by using different methods such as cosine similarity, Euclidean distance, Extended Jaccard) to all objects in that cluster. The first step of clustering is to select as initial cluster centers K randomly selected documents, the seeds. The algorithm then moves the cluster centers seed around in space to minimize tweets. This is done iteratively by repeating two steps until a stopping criterion is met: Reassigning documents to the cluster with the closest centroid and recomputing each centroid based on the current members of its cluster. Assignment of documents to clusters (the partitioning function ;) does not change between iterations. Except for cases with a bad local minimum, this produces a good clustering, but run-time may be unacceptably long. Terminate when the decrease in tweets falls below a threshold O. For small O, this indicates that we are close to convergence. Again, we need to combine it with a bound on the number of iterations to prevent very long run-times.

3.3.3 Training the Model :

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.

3.3.4 Evaluation :

Model evaluation techniques in machine learning are helping us to find a better model among all other models in machine learning. It is simply the selection of machine learning models or measuring the performance of machine learning models.

3.4 DATASET

The dataset used to test the efficiency of the model is produced by kaggle, containing 20,000 data.

- Attributes of dataset :
  - Id
  - Content

{"content": " This is a test message":{"notes":"","label":["1"]},"extras":null}
{"content": " You are looking ugly":{"notes":"","label":["1"]},"extras":null}
{"content": " Get fucking real
dude.","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": " She is as dirty as they come and that crook Rengel the Dems are
so fucking corrupt it's a joke. Make Republicans look like
...","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": " why did you fuck it up. I could do it all day too. Let's do it
when you have an hour. Ping me later to sched writing a book
here.","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": " Dude they dont finish enclosing the fucking showers. I hate half
assed jobs. Whats the reasononing behind it? Makes no
sense.","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": " WTF are you talking about Men? No men thats not a menage that's
just gay.","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "Ill save you the trouble sister. Here comes a big ol fuck France
block coming your way here on the
twitter.","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": " Im dead serious.Real athletes never cheat don't even have the
appearance of at his level. Fuck him dude seriously I think he
did","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "...go absolutely insane.hate to be the bearer of bad
news..LoL..dont shoot the messenger (cause we all know you bought that
pistol","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "Lmao im watching the same thing ahaha. The gay guy is hilarious!
\"Dede having a good day and I dont want anyone to mess it
up.\"","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "LOL no he said What do you call a jail cell to a gay guy?
Paradise! ahaha.","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "truth on both counts that guy is an ass and their product is sub
par. I tell people try Dalesandros
orJim's","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "Shakespeare
nerd!","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "you are SUCH a fucking
dork","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "Heh. Fuck 'em
WHERE?!?","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "damn it i totally forgot that
one!","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "wow damn I would have been pissed @
that...","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "nigga u geigh lmao! fuck yo finals
beeeeeitch","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "that sucks
:(","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "read that this morning. my fav is how they just straight up say
\"cum shots\"","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "Unibroue 17 !!!! Another damn good
Unibroue","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "damn your evil 60 minute IPA beckoning me from the fridge right
now...","annotation":{"notes":"","label":["1"]},"extras":null}
{"content": "It did then my fucking dad turned it off. I just don't think it
likes your movies. I was tryin to watch Nanny

3.5 ALGORITHMS TO BE USED

3.5.1 Random Forest Classifier:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Random Forest classifier is consists of multiple decision tree classifiers [25]. Each tree gives a class prediction individually. The maximum number of the predicted class is our final result. This classifier is a supervised learning model which provides accurate result because several decision trees are merged to make the outcome. Instead of relying on one decision tree, the random forest takes the prediction from each generated tree and based on the majority votes of predictions, and it decides the final output. For example, if we have two classes namely A and B and the most of the decision tree predict the class label B of any instance, then RF will decides the class label B as follows: f(x) = majority vote of all tree as B

3.5.2 Decision Tree Classifier :

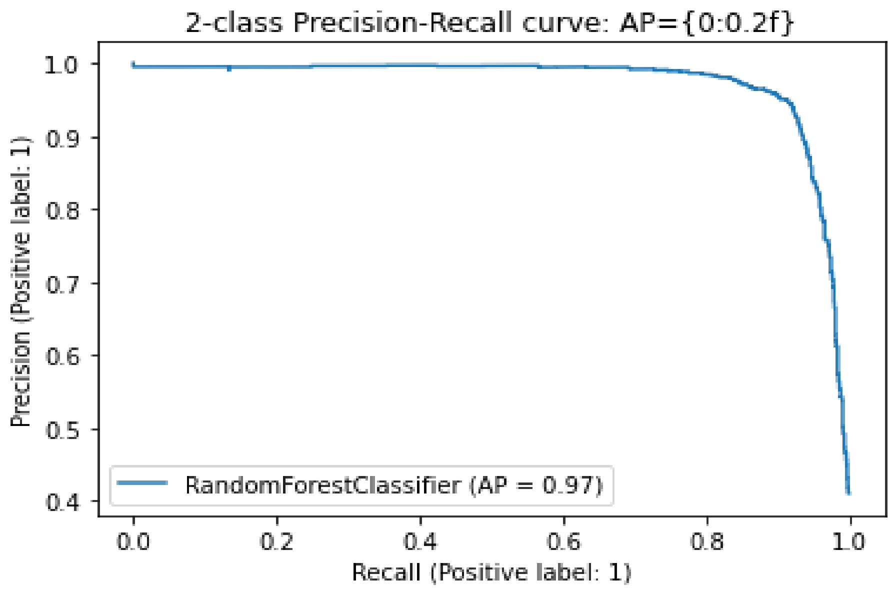
Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset.

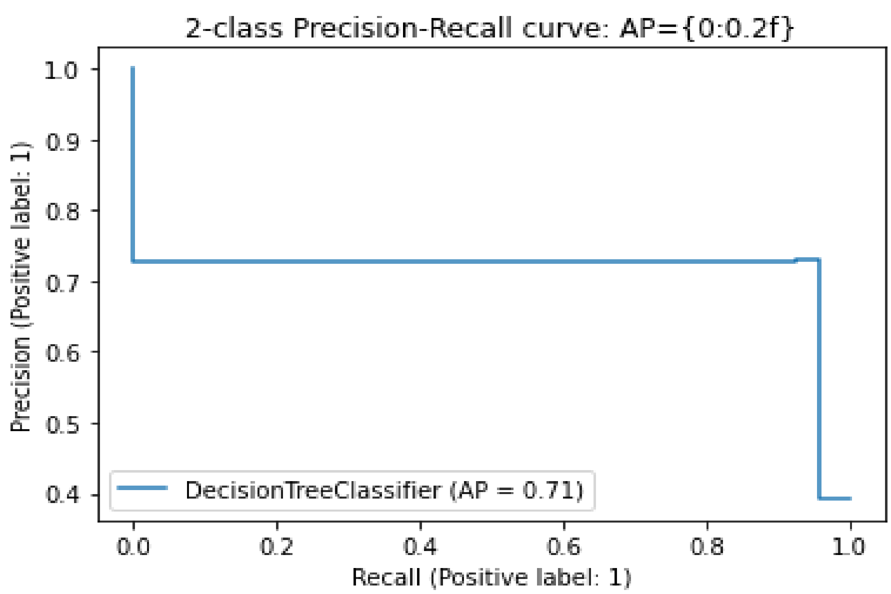
The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).The decision tree classifier can be used in both classification and regression. It can help represent the decision as well as make a decision. The decision tree is a treelike structure where each internal node represents a condition, and each leaf node represents a decision. A classification tree returns the class where the target falls. A regression tree yields the predicted value for an addressed input.

3.5.3 Gaussian :

The Gaussian Processes Classifier is a classification machine learning algorithm. Gaussian Processes are a generalization of the Gaussian probability distribution and can be used as the basis for sophisticated non-parametric machine learning algorithms for classification and regression. They are a type of kernel model, like SVMs, and unlike SVMs, they are capable of predicting highly calibrated class membership probabilities, although the choice and configuration of the kernel used at the heart of the method can be challenging.

Gaussian probability distribution functions summarize the distribution of random variables, whereas Gaussian processes summarize the properties of the functions, e.g. the parameters of the functions. As such, you can think of Gaussian processes as one level of abstraction or indirection above Gaussian functions. A Gaussian process is a generalization of the Gaussian probability distribution. Whereas a probability distribution describes random variables which are scalars or vectors (for multivariate distributions), a stochastic process governs the properties of functions.


2-class Precision-Recall curve: AP={0:0.2f}
Precision (Positive label: 1)
Recall (Positive label: 1)
RandomForestClassifier (AP = 0.97)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.0
0.2
0.4
0.6
0.8
1.0


2-class Precision-Recall curve: AP={0:0.2f}
Precision (Positive label: 1)
Recall (Positive label: 1)
DecisionTreeClassifier (AP = 0.71)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.0
0.2
0.4
0.6
0.8
1.0

Chapter 4

RESULT ANALYSIS

Accuracy is often the most used metric representing the percentage of correctly predicted observations, either true or false. To calculate the accuracy of a model performance, the following equation can be used: In most cases, high accuracy value represents a good model, but considering the fact that we are training a classification model in our case, an article that was predicted as true while it was actually false (false positive) can have negative consequences; similarly, if an article was predicted as false while it contained factual data, this can create trust issues.

Accuarcy = TP+TN/TP+TN+FP+FN

Confusion Matrix

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:

1. TP = True Positives

2. FP = False Positives

3. TN = True Negatives

4. FN = False Negatives

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Online harassment or cyberbullying behaviors have become a severe issue that damages the life of people on a large scale. Successful cyberbullying detection would enable early identification of damaging and threatening scenarios and control such incidents from happening.The use of internet and social media has clear advantages for societies, but their frequent use may also have significant adverse consequences. This involves unwanted sexual exposure, cybercrime and cyberbullying. We developed a model for detecting cyberbullying behavior and its severity in Twitter. Feature generation with PMI at pre-processing stage has proven to be the efficient technique to handle class imbalance in binary and multi-class classification where misclassification for minority class (es) has higher cost in terms of its impact on reliability of detection model. The developed model is a feature-based model that uses features from tweets contents to develop a machine learning classifier for classifying the tweets as cyberbullying or non-cyberbullying and its severity as low, medium, high or none

5.2 Future Scope

The anti-harassment policy and standards supplied by social platforms and power to flag and block or report the bully are useful steps towards safer online community, but they are not enough. Popular social media platforms such as Twitter, Facebook, and Instagram or others receive an enormous number of such flagged content every day; hence, scrutinizing immense reported content and users is very time-consuming and not practical and effective. In such cases, it will be significantly helpful to design automated, datadriven methods for evaluating and detecting such harmful behaviors in social media. Successful cyberbullying detection would enable early identification of damaging and threatening scenarios and control such incidents from happening. Future study could enhance automated cyberbullying detection by combining textual data with video and images to build a machine learning model to detect cyberbullying behavior and its severity, which could be step towards automated systems for analyzing contemporary social online behaviors from written text and visual content that can negatively affect mental health. The detection algorithm could analyse the bully's posts and then align it to preselected level of severity thus gives early awareness about extent of cyberbullying detection

Chapter 6

APPENDIX

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import TfidfTransformer, CountVectorizer, TfidfVector
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split

from nltk.stem.porter import PorterStemmer
import nltk
import re, string
from nltk.corpus import stopwords

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import cross_val_score

from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import plot_precision_recall_curve
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.metrics import classification_report
from sklearn import metrics


url = 'https://drive.google.com/uc?export=download&id=12fBlhsa5GIdtme1jT3KlPPIgIdjzqhv1'
df = pd.read_json(url, lines= True,orient='columns')
df.head


for i in range(0,len(df)):
    if df.annotation[i]['label'][0] == '1':
        df.annotation[i] = 1
    else:
        df.annotation[i] = 0


df.drop(['extras'],axis = 1,inplace = True)
df
```

```
df.shape

df['annotation'].value_counts().sort_index().plot.bar()

print("PosiNon cyber trollingtive: ", df.annotation.value_counts()[0]/len(df.annotation)*1
print("Cybertrolling: ", df.annotation.value_counts()[1]/len(df.annotation)*100,"%")

nltk.download('stopwords')
stop = stopwords.words('english')

regex = re.compile('[%s]' % re.escape(string.punctuation))

def test_re(s):
    return regex.sub('', s)

df ['content_without_stopwords'] = df['content'].apply(lambda x: ' '.join([word for word i
df ['content_without_puncs'] = df['content_without_stopwords'].apply(lambda x: regex.sub('
del df['content_without_stopwords']
del df['content']
df

#Stemming
porter_stemmer = PorterStemmer()
#punctuations
nltk.download('punkt')
tok_list = []
size = df.shape[0]

for i in range(size):
  word_data = df['content_without_puncs'][i]
  nltk_tokens = nltk.word_tokenize(word_data)
  final = ''
  for w in nltk_tokens:
    final = final + ' ' + porter_stemmer.stem(w)
  tok_list.append(final)

df['content_tokenize'] = tok_list
del df['content_without_puncs']
df

noNums = []
for i in range(len(df)):
  noNums.append(''.join([i for i in df['content_tokenize'][i] if not i.isdigit()]))

df['content'] = noNums
df

tfIdfVectorizer=TfidfVectorizer(use_idf=True, sublinear_tf=True)
tfIdf = tfIdfVectorizer.fit_transform(df.content.tolist())
```

```
print(tfIdf)

print(tfIdf.shape)


df2 = pd.DataFrame(tfIdf[2].T.todense(), index=tfIdfVectorizer.get_feature_names(), column
df2 = df2.sort_values('TF-IDF', ascending=False)
print (df2.head(10))


dfx = pd.DataFrame(tfIdf.toarray(), columns = tfIdfVectorizer.get_feature_names())
print(dfx)


def display_scores(vectorizer, tfidf_result):
    scores = zip(vectorizer.get_feature_names(),
                 np.asarray(tfidf_result.sum(axis=0)).ravel())
    sorted_scores = sorted(scores, key=lambda x: x[1], reverse=True)
    i=0
    for item in sorted_scores:
        print ("{0:50} Score: {1}".format(item[0], item[1]))
        i = i+1
        if (i > 25):
          break


display_scores(tfIdfVectorizer, tfIdf)


X=tfIdf.toarray()
y = np.array(df.annotation.tolist())
#Spltting
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)


#Training data biasness
unique_elements, counts_elements = np.unique(y_train, return_counts=True)
print(np.asarray((unique_elements, counts_elements)))


unique_elements, counts_elements = np.unique(y_test, return_counts=True)
print(np.asarray((unique_elements, counts_elements)))


#Random oversampling on training data
from imblearn.over_sampling import RandomOverSampler

oversample = RandomOverSampler(sampling_strategy='not majority')
X_over, y_over = oversample.fit_resample(X_train, y_train)


print(X_over.shape)
print(y_over.shape)
```

```
unique_elements, counts_elements = np.unique(y_over, return_counts=True)
print(np.asarray((unique_elements, counts_elements)))


def getStatsFromModel(model):
  print(classification_report(y_test, y_pred))
  disp = plot_precision_recall_curve(model, X_test, y_test)
  disp.ax_.set_title('2-class Precision-Recall curve: '
                   'AP={0:0.2f}')

  logit_roc_auc = roc_auc_score(y_test, model.predict(X_test))
  fpr, tpr, thresholds = roc_curve(y_test, model.predict_proba(X_test)[:,1])
  plt.figure()
  plt.plot(fpr, tpr, label='(area = %0.2f)' % logit_roc_auc)
  plt.plot([0, 1], [0, 1],'r--')
  plt.xlim([0.0, 1.0])
  plt.ylim([0.0, 1.05])
  plt.xlabel('False Positive Rate')
  plt.ylabel('True Positive Rate')
  plt.title('Receiver operating characteristic')
  plt.legend(loc="lower right")
  plt.savefig('Log_ROC')
  plt.show()


#Supervised Methods
# 3 normal methods
# 2 ensemble methods
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))
score_gau=gnbmodel.score(X_test, y_test)
getStatsFromModel(gnb)


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)
score_dec=metrics.accuracy_score(y_test, y_pred)


#Ensemble methods from here
#abc = AdaBoostClassifier()
#abc.fit(X_over, y_over)
#y_pred = abc.predict(X_test)
#print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
#print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
#score_ada=metrics.accuracy_score(y_test, y_pred);
#getStatsFromModel(abc)
```

```
rfc = RandomForestClassifier(verbose=True) #uses randomized decision trees
rfcmodel = rfc.fit(X_over, y_over)
y_pred = rfc.predict(X_test)
print ("Score:", rfcmodel.score(X_test, y_test))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(rfc)
score_rfc=rfcmodel.score(X_test, y_test)
output = pd.DataFrame({'Predicted':y_pred}) # Heart-Disease yes or no? 1/0
print(output.head())


plt.rcdefaults()
fig, ax = plt.subplots()
algorithms = ('Gaussian', 'Decision Tree Classifier','Random Forest Classifier')
y_pos = np.arange(len(algorithms))
x = (score_gau, score_dec, score_rfc) # scores
ax.barh(y_pos, x, align='center')
ax.set_yticks(y_pos)
ax.set_yticklabels(algorithms)
ax.invert_yaxis() # labels read top-to-bottom
ax.set_xlabel('Performance')
ax.set_title('Which one is the best algorithm?')
for i, v in enumerate(x):
    ax.text(v + 1, i, str(v), color='black', va='center', fontweight='normal')
plt.show()


results=pd.DataFrame(columns=['score'])
results.loc['Gaussian']=[score_gau]
results.loc['Random Forest Classifier']=[score_rfc]
results.loc['Decision Tree Classifier']=[score_dec]


results.sort_values('score',ascending=False).style.background_gradient(cmap='Greens',subse


output.to_csv('output.csv', index=False)
print("Success!")


output.head(10)
```

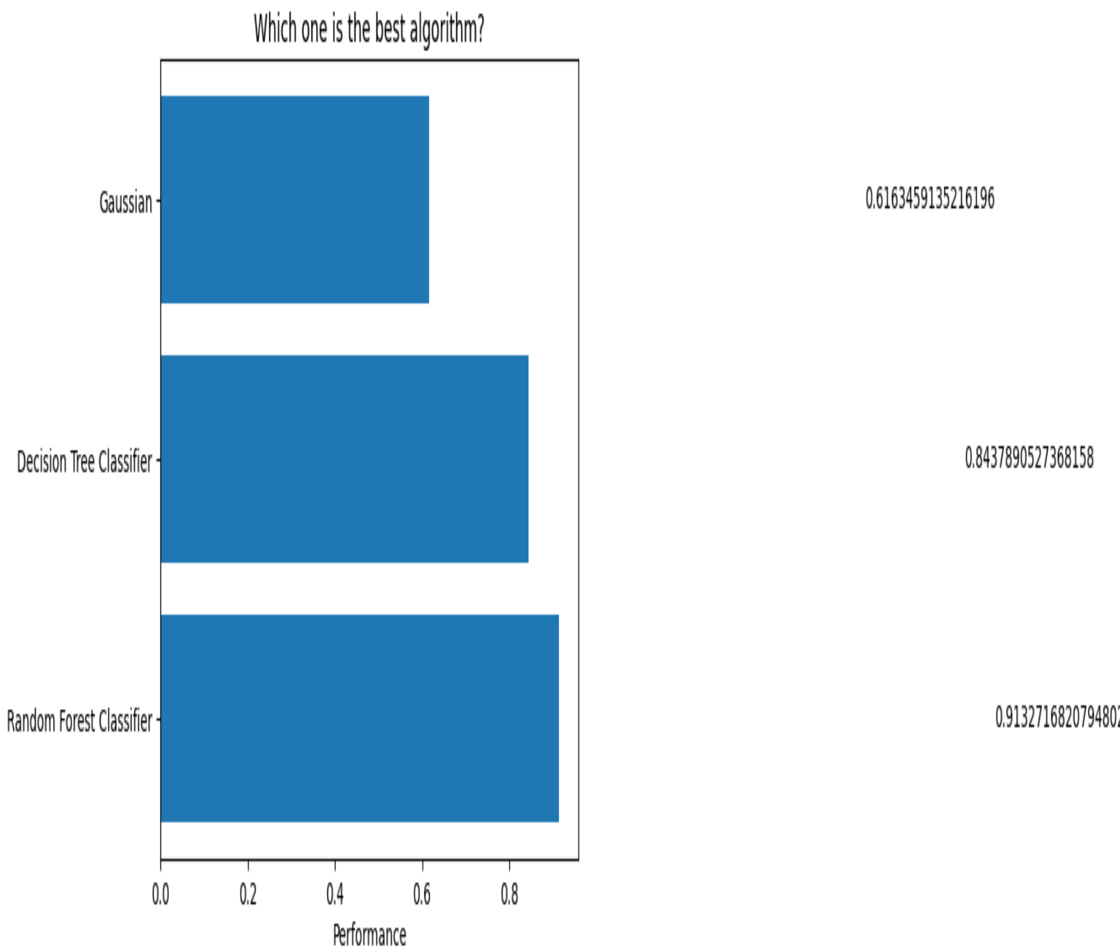
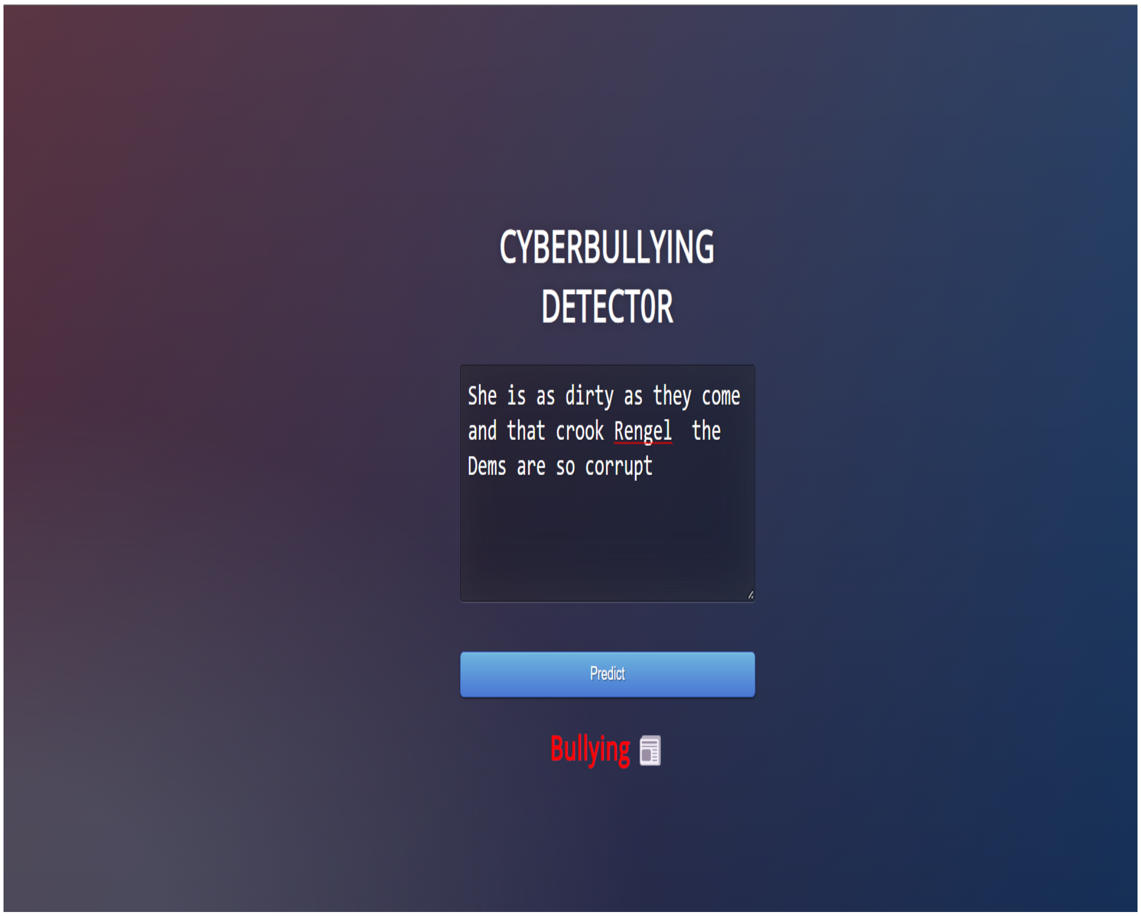
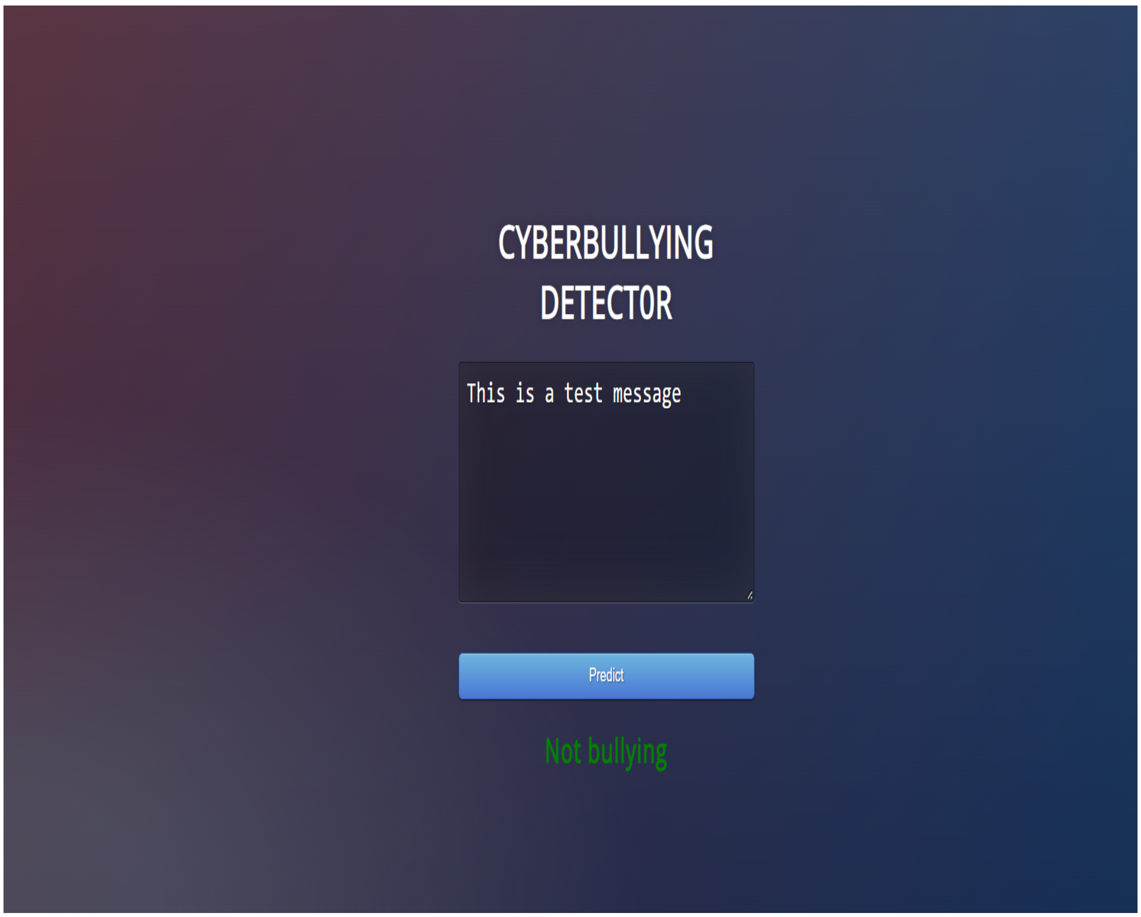


Figure 6.1: Sample Dataset.

	score
Random Forest Classifier	0.913272
Decision Tree Classifier	0.843789
Gaussian	0.616346


CYBERBULLYING
DETECTOR
She is as dirty as they come and that crook Rengel the Dems are so corrupt
Predict
Bullying


CYBERBULLYING
DETECTOR
This is a test message
Predict
Not bullying

Chapter 7

REFERENCES

[1] C. Fuchs, Social media: A critical introduction. Sage, 2017

[2] N. Selwyn, “Social media in higher education,” The Europa world of learning, vol. 1, no. 3, pp. 1–10, 2012.

[3] H. Karjaluoto, P. Ulkuniemi, H. Keinanen, and O. Kuivalainen, “An- ¨ tecedents of social media b2b use in industrial marketing context: customers’ view,” Journal of Business Industrial Marketing, 2015.

[4] W. Akram and R. Kumar, “A study on positive and negative effects of social media on society,” International Journal of Computer Sciences and Engineering, vol. 5, no. 10, pp. 351–354, 2017.

[5] D. Tapscott et al., The digital economy. McGraw-Hill Education,, 2015.