TOXIC COMMENT CLASSIFICATION IN SOCIAL NETWORKING

Using CNN Algorithm

Developed by Sonika Prakash, Poornima G B, Anu Bai, Ramyashree G T, Spoorthi R S
Smart Bridge - Remote Summer Internship Program

1.INTRODUCTION

There are various areas in which Deep Learning can be used in Toxic comment classification sectors. One of the main benifits of introducing Deep Learning to toxic comment classification is more accurate predictions.

Over the years, social media and social networking use have been increasing exponentially due to an upsurge in the use of the internet. Plenty of information arises from online conservation in a daily basis as people are able to discuss, express themselves and shoot their opinion via these platforms. Social networks sometimes become a place for threats and other components of Cyberbullying. This could be dangerous and insults the personality, it is quite obvious that debates may arise due to differences in the openion during which offensive language termed as toxic comments may be used then these type of conservation may take into destructive level and ends in fight over the social media. These toxic comments may be insulting, obsence sometimes threatening so, these clearly pose the threat of abuse, consequently some people stop giving up their opinions which results in unhealthy and unfair discussion.

As a result, different platforms and communities find it very difficult to facilitate fair conversation and are often forced to either limit the user comments on the social media or get dissolved by shutting down user comments completly. so that to prevent these type of identity hate through comments in social media we come up with a solution to detect different types of toxicity in the comment using Deep Learning.

There are huge number of phases in the prediction based on Deep Learning.
Data Collections the first phase, where data should be collected. It should be huge data set according to the requirements. One should collect or create the data for the prediction. Second phase is Text Pre-processing and contains lot of sub-phases for text pre-processing, it include Import the libraries, one hot encoding, tokenization, spliting.
In splitting data is to be split into two parts as train_data and test_data for training of

the model. Last phase is model building that includes importing the model building libraries, configure the learning process, and save the model. Fourth phase is creation of HTML file and building python code.

1.1 OVERVIEW

Nowadays the flow of data on the internet has grown dramatically, especially with the appearance of social sites. Social networks sometimes become a place for threats, insults and other components of cyberbullying. A huge number of people are involved in online social networks. Hence, the protection of network users from anti-social behaviour is an important activity.

One of the major tasks of such activity is automated detecting the toxic comments. Toxic comments are texual comments with threats, obscene, racism etc.

To prevent this we come up with a solution, in that various techniques are used for human-free detecting the toxic comments. Bag of words statics and bag of symbols statics are the typical source of information for the toxic comments detection. Usually, the following statistics-based features are used: length of the comment, number of tokens with non-alphabet symbols, number of abusive, arguments. aggressive, and threatening words in the comment, etc. A neural network model is used to classify the comments.

1.2 PURPOSE

Our aim from the project is to make use of pandas, matplotlib and seaborn libraries from python to extract the libraries for deep learning for the Toxic comment classification. Here we used CNN algorithm to identify the toxic comments.

2. LITERATURE SURVEY

Data mining is the process of analysing data from different perspectives and extracting useful knowledge from it. It is the core of knowledge discovery process. Different data mining techniques include classification, clustering, association rule mining ,prediction and sequential patterns, neural network etc. Classification is the most commonly applied data mining technique, which employs a set of preclassified examples to develop a model that can classify population of records at large. Toxic comment detection is well suited to classification technique. This approach frequently employs CNN algorithm. In classification a training set is used to build the model as the classifier which can classify the data items into its appropriate classes. A test set is used to validate the model.

2.1. EXISTING PROBLEM

The previous models have high time complexity and space complexity whereas this model is constrained with the lot of advantages and with a higher accuracy than any other model already proposed. In this model we used CNN algorithm which give an accuracy of 98% and there is an user friendly user interface tocheck the comment is how much toxic.

2.2 PROPOSED SOLUTION

Convolution Neural Network(CNN) is a Deep Learning algorithm which can take in an input . A large proportion of Online comments.

3.THEORETICAL ANALYSIS

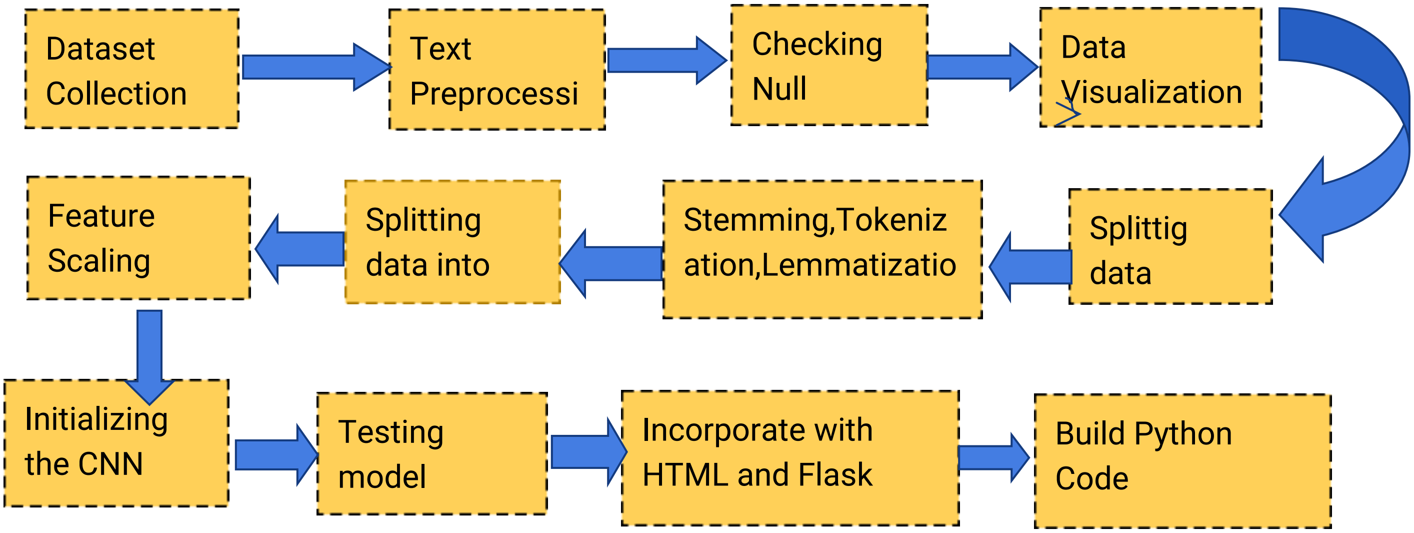
While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results abruptly accurate and from them we selected only one algorithm for the prediction problem i.e., CNN, it assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. That's how the prediction work great with the CNN algorithm.

The pecularity of this problem is collecting the comments and working with prediction of toxic comment at the same time, so we develop an user interface for the identification of the toxic comment. Accuracy is defined as the ratio of number of samples correctly classified by the classifier to the total number of sales for a given test dataset. The formula is as follows

$$\text{Accuracy}=TP+TN/TP+TN+FT+FN$$

Here we used CNN algorithm because it gives 98% of accuracy.

3.1.BLOCK DIAGRAM



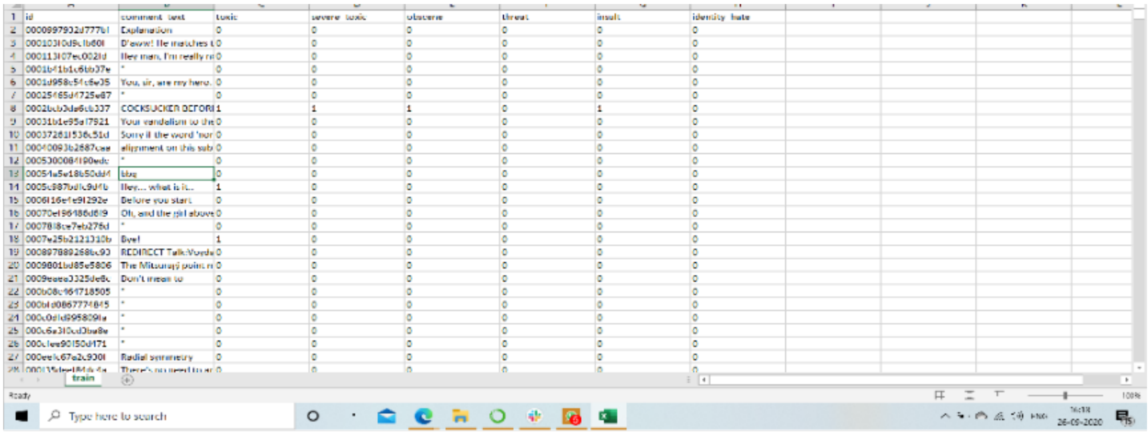
3.2. Software Design

Jupyter Notebook Environment
Spyder Ide
Deep Learning algorithms
Python(Pandas , Numpy, Matplotlib, Seaborn, Sklearn)
HTML

We developed this toxic comment prediction by using the python language. Which is a interpreted and high level programming language and using the deep learning algorithm. For coding we used the jupyter notebook environment of the anaconda distributions and the spyder, it is an integrated specific programming in the python language.

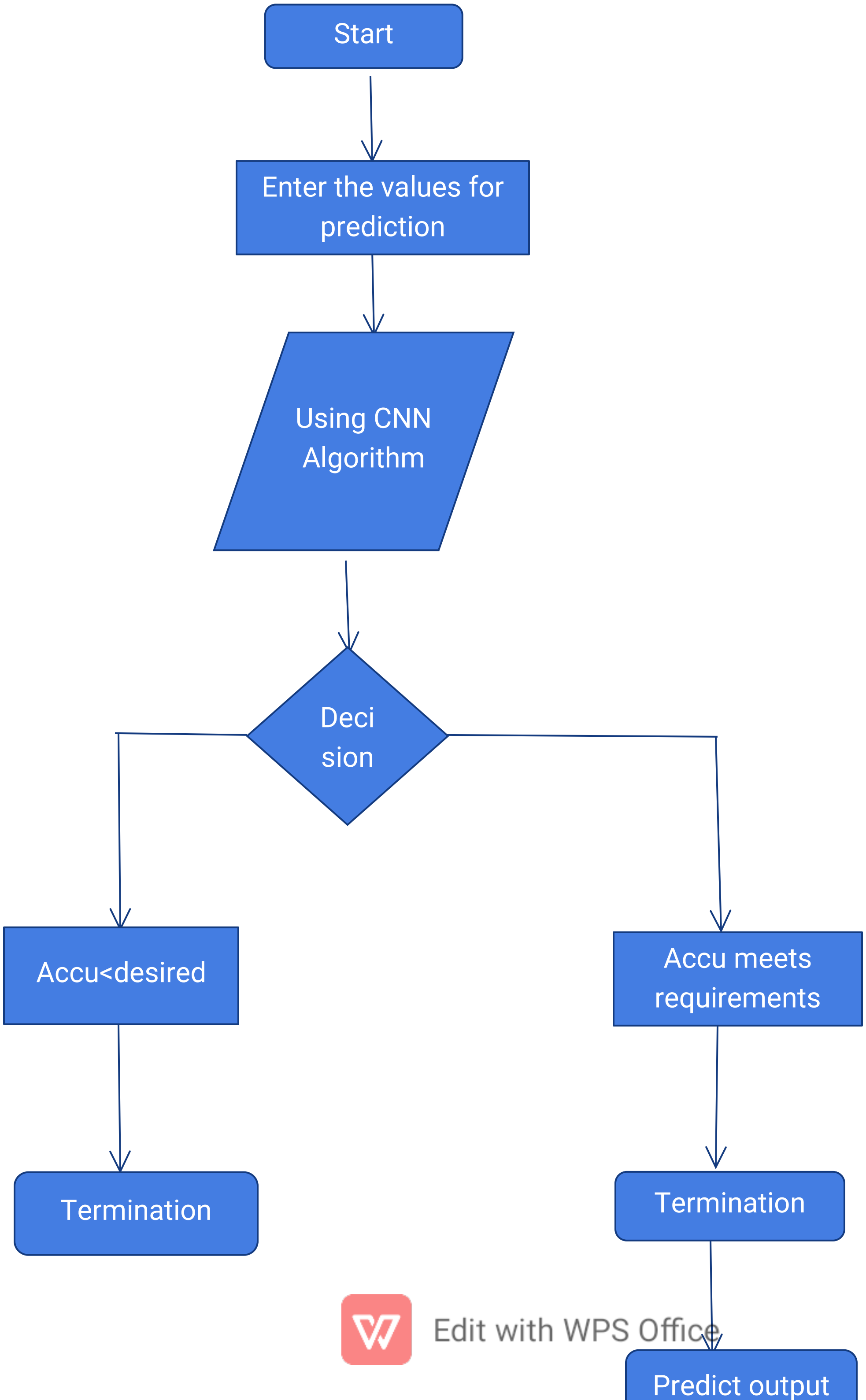
4.EXPERIMENTAL INVESTIGATION

The dataset we used containing 159571 rows and 8 columns. It had no missing values. There were no outliers found. The below figure shows the dataset of our project.



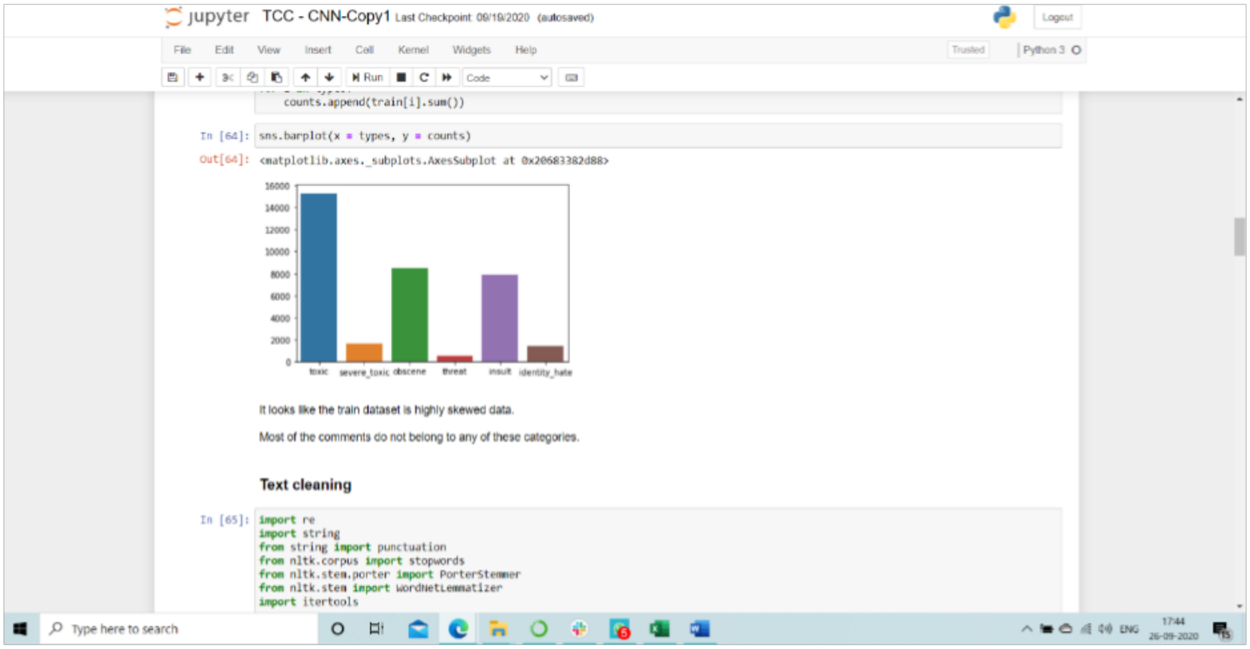
id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0000997932d777bf	Explanation	0	0	0	0	0	0
000103f0d9cfb60f	D'aww! He matches t	0	0	0	0	0	0
000113f07ec002fd	Hey man, I'm really n	0	0	0	0	0	0
0001b41b1c6bb37e	"	0	0	0	0	0	0
0001d958c54c6e35	You, sir, are my hero.	0	0	0	0	0	0
00025465d4725e87	"	0	0	0	0	0	0
0002bcb3da6cb337	COCKSUCKER BEFORE	1	1	1	0	1	0
00031b1e95af7921	Your vandalism to the	0	0	0	0	0	0
00037261f536c51d	Sorry if the word 'nor	0	0	0	0	0	0
00040093b2687caa	alignment on this sub	0	0	0	0	0	0
0005300084f90edc	"	0	0	0	0	0	0
00054a5e18b50dd4	bbq	0	0	0	0	0	0
0005c987bdfc9d4b	Hey... what is it..	1	0	0	0	0	0
0006f16e4e9f292e	Before you start	0	0	0	0	0	0
00070ef96486d6f9	Oh, and the girl above	0	0	0	0	0	0
00078f8ce7eb276d	"	0	0	0	0	0	0
0007e25b2121310b	Bye!	1	0	0	0	0	0
000897889268bc93	REDIRECT Talk:Voyde	0	0	0	0	0	0
0009801bd85e5806	The Mitsurugi point n	0	0	0	0	0	0
0009eaea3325de8c	Don't mean to	0	0	0	0	0	0
000b08c464718505	"	0	0	0	0	0	0
000bfd0867774845	"	0	0	0	0	0	0
000c0dfd995809fa	"	0	0	0	0	0	0
000c6a3f0cd3ba8e	"	0	0	0	0	0	0
000cfee90f50d471	"	0	0	0	0	0	0
000eefc67a2c930f	Radial symmetry	0	0	0	0	0	0
000f35deef84dc4a	There's no need to ar	0	0	0	0	0	0

5. FLOW CHART


Start
Enter the values for prediction
Using CNN Algorithm
Deci sion
Accu<desired
Accu meets requirements
Termination
Termination
Predict output

6.RESULT

In this paper, the CNN algorithm is used to predict its performance.
The obtained results are displayed below, CNN algorithm performs the best with an accuracy of 98%



```
    counts.append(train[i].sum())
```

In [64]:

```
sns.barplot(x = types, y = counts)
```

Out[64]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20683382d88>
```

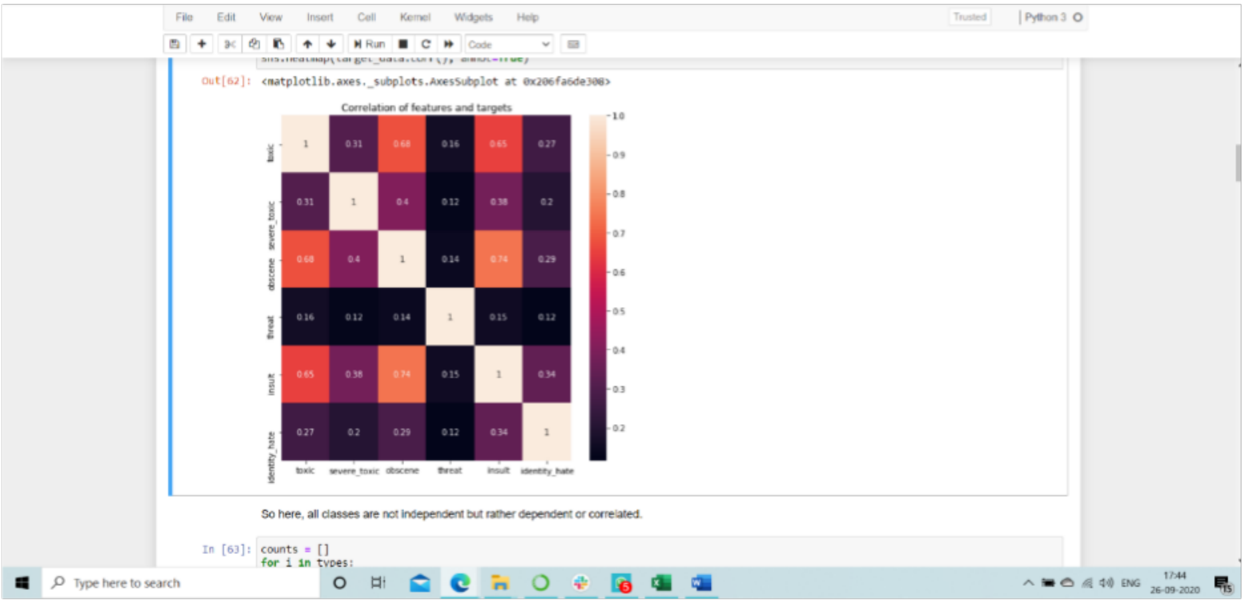
It looks like the train dataset is highly skewed data.

Most of the comments do not belong to any of these categories.

Text cleaning

In [65]:

```
import re
import string
from string import punctuation
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer
import itertools
```



```
sns.heatmap(target_data.corr(), annot=True)
```

```
Out[62]: <matplotlib.axes._subplots.AxesSubplot at 0x206fa6de308>
```

Correlation of features and targets

	toxic	severe_toxic	obscene	threat	insult	identity_hate
toxic	1	0.31	0.68	0.16	0.65	0.27
severe_toxic	0.31	1	0.4	0.12	0.38	0.2
obscene	0.68	0.4	1	0.14	0.74	0.29
threat	0.16	0.12	0.14	1	0.15	0.12
insult	0.65	0.38	0.74	0.15	1	0.34
identity_hate	0.27	0.2	0.29	0.12	0.34	1

So here, all classes are not independent but rather dependent or correlated.

```
In [63]: counts = []
         for i in types:
```

7.ADVANTAGES AND DISADVANTAGES

Advantages:

- Easy and simple User Interface to predict toxic comment status.
- CNN gives the accurate result of the prediction up to 98% which is the algorithm we used for prediction.
- It is composed using HTML and python for the web usage in real time.
- It can work in real time and predict as soon as the necessary details for the prediction are given to the model.

Disadvantages

- It could not work anywhere like an web-application, if one is using other should be quite.
- Needs more than a single value for the prediction.

9.CONCLUSION

In this paper, the CNN algorithm is adopted to build a UI model for predicting toxic comment. The experiment shows that the CNN algorithm perform well with an accuracy 98%. There is no definitive guide of which algorithms to use given any situation. What may work on some data sets may not necessarily work on others. Therefore, always evaluate methods using cross validation to get a reliable estimates.

10.FUTURE SCOPE

In future, more datasets about the toxic prediction can be collected and merged into one single dataset which can improve the accuracy. In further study, we will try to conduct experiments on larger datasets or try to tune the model so as to achieve the state-of-art performance of the model and a great UI support system making it complete web application model.

11.BIBLIOGRAPHY

- https://www.kaggle.com/sathishkumarsg10/beginner-bidirectional-lstm-using-glove-vectors
- https://www.kaggle.com/rhodiumbeng/classifying-multi-label-comments-0-9741-lb
- https://www.kaggle.com/yekenot/textcnn-2d-convolution

APPENDIX

HTML:

about.html

```
<!DOCTYPE html>
<html>
<head>
<title>About</title>
<!-- Compiled and minified CSS -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css"
>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
  background-color: #26a69a;
  background-image:url('https://media-assets-03.thedrum.com/cache/images/thedrum-
prod/s3-news-tmp-90538-social-media-mobile-icons-snapchat-facebook-instagram-ss-
1920--2x1--940.jpg');
  background-repeat:repeat;
  background-attachment:fixed;
  background-size:cover;
  min-height: 100%;
  min-width: 100%;
}
* {
  box-sizing: border-box;
}


/* Add padding to containers */
.container {
  padding: 2px;
  background-color: white;
  width: 80%;
}

/* Full-width input fields */
input[type=text]{
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}

input[type=text]:focus{
  background-color: #ddd;
  outline: none;
}

/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

.brand{
                            background: #703fba !important;
}
.brand-text{
                         color: #703fba !important;
}

</style>
</head>
<body class="grey lighten-4">

<nav class="white z-depth-0">
<div class="brand-logo brand-text" style="font-family: Candara; margin-left: 15px;"
 ><b>TOXIC COMMENT CLASSIFICATION</b></div>
<ul id="nav-mobile" class="right hide-on-small-and-down">
 <li><a href="/" class="btn brand z-depth-0">Home</a></li>
 <li><a href="/about.html" class="btn brand z-depth-0">About the project</a></li>
 <li><a href="/index1.html" class="btn brand z-depth-0">Make predictions</a></li>
</ul>
</nav>
<div class="container"  align="justified" style="margin-top: 50px; background: rgba(255, 255, 255, 1);">
<pre style="color:black; font-family: Century Gothic; font-size: 16px; opacity: 1; margin-left: 10px; margin-right: 10px;">
<h5><b>PROJECT DESCRIPTION:</b></h5>
Nowadays, social networks sometimes have become a place for threats, insults, and other components of cyberbullying. A huge number of people are
involved in online social networks. Hence, the protection of network users from anti-social behavior is an important activity. One of the major tasks of
such activity is automated detection of the toxic comments. Toxic comments are textual comments with threats, insults, obscene, racism, etc. In recent
years there have been many cases in which authorities arrested some users of social sites because of the negative (abusive) content of their personal
pages.
</pre>
```

```
</div>
<div class="container"  align="justified" style="margin-top: 50px; background: rgba(255, 255, 255, 1);">
<pre style="color:black; font-family: Century Gothic; font-size: 16px; opacity: 1; margin-left: 10px; margin-right: 10px;">
<h5><b>SOLUTION:</b></h5>
We are proposing a solution in which various techniques are used for human-free
detection of the toxic comments. Bag of words statistics and bag of
symbols statistics are the typical source of information for the toxic comments
detection. Usually, the following statistics-based features are used:
length of the comment, number of capital letters, number of exclamation marks,
number of question marks, number of spelling errors, number of tokens
with non-alphabetic symbols, number of abusive, aggressive, and threatening words in
the comment, etc. A neural network model is used to classify the
comments.
We have developed a Convolutional Neural Network(CNN) model for this. The accuracy
we have achieved is 98%. To check the working of our model
click on the "Prediction" button in the navigation bar.
</pre>
</div>
<div class="container"  align="justified" style="margin-top: 50px; background: rgba(255, 255, 255, 1);">
<pre style="color:black; font-family: Century Gothic; font-size: 16px; opacity: 1; margin-left: 10px; margin-right: 10px;">
<b>Project developed by:</b>
Sonika Prakash, Poornima GB, Spoorthy RS, Ramyashree GT, Anu Bai
```

Home.html

```
<!DOCTYPE html>
<html>
<head>
<title>Home</title>
<!-- Compiled and minified CSS -->
<link rel="stylesheet"
```

```
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css"
>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
  background-color: #26a69a;
  background-image:url('https://media-assets-03.thedrum.com/cache/images/thedrum-
prod/s3-news-tmp-90538-social-media-mobile-icons-snapchat-facebook-instagram-ss-
1920--2x1--940.jpg');
  background-repeat:repeat;
  background-attachment:fixed;
  background-size:cover;
  min-height: 100%;
  min-width: 100%;
}
* {
  box-sizing: border-box;
}


/* Add padding to containers */
.container {
  padding: 2px;
  background-color: white;
  width: 80%;
}

/* Full-width input fields */
input[type=text]{
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}
```

```
input[type=text]:focus{
  background-color: #ddd;
  outline: none;
}

/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

.brand{
                            background: #703fba !important;
}
.brand-text{
                         color: #703fba !important;
}

</style>
</head>
<body class="grey lighten-4">

<nav class="white z-depth-0">
<div class="brand-logo brand-text" style="font-family: Candara; margin-left: 15px;"
 ><b>TOXIC COMMENT CLASSIFICATION</b></div>
<ul id="nav-mobile" class="right hide-on-small-and-down">
 <li><a href="/" class="btn brand z-depth-0">Home</a></li>
 <li><a href="/about.html" class="btn brand z-depth-0">About the project</a></li>
 <li><a href="/index1.html" class="btn brand z-depth-0">Make predictions</a></li>
</ul>
</nav>
<div class="container"  align="justified" style="margin-top: 50px; background: rgba(255,
255, 255, 0.7);">
<center>
<div style="margin-left:10px; margin-right:10px;">
<b>
<pre style="color:black; font-family:  Century Gothic; font-size: 45px; opacity: 1;">
Hello! Welcome to our website.
```

```
</pre>
<pre style="color:black; font-family: Century Gothic; font-size: 25px; opacity: 1;">
Everyone loves social media these days.
The flow of data over the internet has grown dramatically, especially with the appearance of social
networking sites. We have developed a project which focuses on the comments given on these
social networking platforms.
</pre>
</b>
</center>
</div>
</div>
</body>
```

Index.html

```
<!DOCTYPE html>
<html>
<head>
<!-- Compiled and minified CSS -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
  background-color: #26a69a;
  background-image:url('https://media-assets-03.thedrum.com/cache/images/thedrum-prod/s3-news-tmp-90538-social-media-mobile-icons-snapchat-facebook-instagram-ss-1920--2x1--940.jpg');
  background-repeat:repeat;
  background-attachment:fixed;
  background-size:cover;
  min-height: 500px;
  min-width: 100px;
```

```
}



* {
  box-sizing: border-box;
}


/* Add padding to containers */
.container {
  padding: 16px;
  background-color: white;
  width: 400px;
}

/* Full-width input fields */
input[type=text]{
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}

input[type=text]:focus{
  background-color: #ddd;
  outline: none;
}

/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

.brand{
```

```
                  background: #703fba !important;
}
.brand-text{
                color: #703fba !important;
}

</style>
</head>
<body class="grey lighten-4">

<nav class="white z-depth-0">
<div class="brand-logo brand-text" style="font-family: Candara; margin-left: 15px;"
 ><b>TOXIC COMMENT CLASSIFICATION</b></div>
<ul id="nav-mobile" class="right hide-on-small-and-down">
 <li><a href="/" class="btn brand z-depth-0">Home</a></li>
 <li><a href="/about.html" class="btn brand z-depth-0">About the project</a></li>
 <li><a href="/index1.html" class="btn brand z-depth-0">Make predictions</a></li>
</ul>
</nav>
<br>
<br>
<br>
<form action="{{ url_for('y_predict')}}"method="post">
  <div class="container" align="center">

   Type a comment here to classify it.
   <input type="text" name="comment" placeholder="Type your comment here.."
required="required" />

   <button type="submit" class="btn brand z-depth-0 btn-large">Predict</button>
  <br>
  <br>
  {{ pred_text1 }}
  <br/>
  {{ pred_text2 }}
  <br/>
  {{ pred_text3 }}
  <br/>
```

```
   {{ pred_text4 }}
   <br/>
   {{ pred_text5 }}
   <br/>
   {{ pred_text6 }}
   <br/>
  </div>


</form>
</center>

<br>

</body>
</html>
```

APP.PY

```
import numpy as np
from flask import Flask, request, jsonify, render_template
from joblib import load
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow import keras
app = Flask(__name__)
model = keras.models.load_model('newmodel.h5')
tks = load('newtokens.save')

@app.route('/')
def home():
    return render_template('home.html')

@app.route('/about.html')
def about():
    return render_template('about.html')
```

```
@app.route('/index1.html')
def index():
    return render_template('index1.html')

@app.route('/index1.html/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    comment = [[x for x in request.form.values()]]
    print(comment)
    comment_seq = tks.texts_to_sequences(comment[0])
    req = pad_sequences(comment_seq, maxlen=200)
    p = model.predict(req)
    new_p = [i*100 for i in p]
    new_p = list(new_p[0])
    print(new_p)
    output=new_p

    #return render_template('index.html', prediction_text='Toxic : {}\nSevere toxic :
{}\nObscene : {}\nThreat : {}\nInsult : {}\nIndentity hate :
{}'.format("{:.2f}%".format(output[0]), "{:.2f}%".format(output[1]),
"{:.2f}%".format(output[2]), "{:.2f}%".format(output[3]), "{:.2f}%".format(output[4]),
"{:.2f}%".format(output[5])))
    return render_template('index1.html', pred_text1 = 'Toxic :
{}'.format("{:.2f}%".format(output[0])),
                           pred_text2 = 'Severe toxic : {}'.format("{:.2f}%".format(output[1])),
                           pred_text3 = 'Obscene : {}'.format("{:.2f}%".format(output[2])),
                           pred_text4 = 'Threat : {}'.format("{:.2f}%".format(output[3])),
                           pred_text5 = 'Insult: {}'.format("{:.2f}%".format(output[4])),
                           pred_text6 = 'Identity hate : {}'.format("{:.2f}%".format(output[5])))

'''@app.route('/predict_api',methods=['POST'])
def predict_api():

    #For direct API calls trought request
    data = request.get_json(force=True)
    prediction = model.y_predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)'''

if __name__ == "__main__":
    app.run(debug=True)
```