SOCIAL MEDIA DETOXIFIER
A STEP TOWARDS CYBER SECURITY / AGAINST CYBER CRIMES
MINOR PROJECT - GROUP 5
TEAM MEMBERS
PRABAL MANHAS (LEADER) 20BCS4513
ANURAG KUMAR 20BCS4567
GIRJANAND TIWARY 20BCS4506

SOCIAL MEDIA DETOXIFIER ~ a step towards Cybersecurity & against Cybercrimes

Brief Introduction - As we know that Cybercrimes and Cyberbullying cases are taking place at a rapid pace nowadays, which leads to several hate comments, threats to someone's personal life, fake accounts, scammers and bots, all with the help of social media. So our project aims at building a Social Media Detoxifier by which we can automatically track the hate comments, fetch the hate speech audio messages, as well as the fake accounts and bots being used for illegal activities

🔍 **Modules of our Project :-**

👉 **Audio/Video Analyzer -** To fetch toxic comments, text strings from large audio and video datasets.

👉 **IP Address Tracker -** To get the real time location coordinates of the scammers and saving the fetched details in a HTML File to save all IP Logs.

👉 **Phone Number Details Tracker -** Ever got bullied by scammers on phone calls, threats to your personal life and financial risks ?

Don't Worry, we have also designed a program to track all the data such the ISP Provider Name, Precise Location, Scam Results Reports etc. just by entering the cell no. of the Scammer.

👉 **Toxicity Analyser -** Building a toxicity analyser to perform analysis on a large dataset model trained by us, in order to fetch the toxic comments present in it, it can be the tweets posted by a twitter user, comments on Instagram/FB etc. Also classifying them into their respective categories such as ***Severe Toxic, Harassment, Bullying, Threats, Obscene, Insult or Identity Hate***

🎙️ AUDIO ANALYSER - PROGRAM (Helps to generate text strings from large audio dataset files)

In [ ]:

```
# Uploading a sample audio file containing toxic audio message to genertate text messag
es from it
```

In [ ]:

```
!pip install -q transformers
```

In [ ]:

```
import librosa
import torch
from transformers import Wav2Vec2ForCTC, Wav2Vec2Tokenizer
```

In [ ]:

```
print("TOXIC VOICE 🔊 TO TEXT 💬 || MINOR PROJECT")
print("CHECKING THE INPUT SOUND 🔊 ......")
```

```
TOXIC VOICE 🔊 TO TEXT 💬 || MINOR PROJECT
CHECKING THE INPUT SOUND 🔊 ......
```

Uploading sample toxic audio message containing threat audio messages.

In [ ]:

```
speech, rate = librosa.load("toxic_audio.wav",sr=16000)
```

In [ ]:

```
import IPython.display as display
display.Audio("toxic_audio.wav", autoplay=True)
```

Out[ ]:

0:09 / 0:09

In [ ]:

```
tokenizer = Wav2Vec2Tokenizer.from_pretrained("facebook/wav2vec2-base-960h")
model = Wav2Vec2ForCTC.from_pretrained("facebook/wav2vec2-base-960h")
```

```
The tokenizer class you load from this checkpoint is not the same type as
the class this function is called from. It may result in unexpected tokeni
zation.
The tokenizer class you load from this checkpoint is 'Wav2Vec2CTCTokenize
r'.
The class this function is called from is 'Wav2Vec2Tokenizer'.
/usr/local/lib/python3.7/dist-packages/transformers/models/wav2vec2/tokeni
zation_wav2vec2.py:752: FutureWarning: The class `Wav2Vec2Tokenizer` is de
precated and will be removed in version 5 of Transformers. Please use `Wav
2Vec2Processor` or `Wav2Vec2CTCTokenizer` instead.
  FutureWarning,
Some weights of Wav2Vec2ForCTC were not initialized from the model checkpo
int at facebook/wav2vec2-base-960h and are newly initialized: ['wav2vec2.m
asked_spec_embed']
You should probably TRAIN this model on a down-stream task to be able to u
se it for predictions and inference.
```

In [ ]:

```
input_values = tokenizer(speech, return_tensors = 'pt').input_values
```

In [ ]:

```
input_values
```

Out[ ]:

```
tensor([[-0.0024, -0.0026, -0.0024,  ..., -0.0024, -0.0026, -0.0024]])
```

In [ ]:

```
#Store logits (non-normalized predictions)
logits = model(input_values).logits
```

In [ ]:

```
#Store predicted id's
predicted_ids = torch.argmax(logits, dim =-1)
```

In [ ]:

```
#decode the audio to generate text
transcriptions = tokenizer.decode(predicted_ids[0])
```

In [ ]:

```
print("THE TOXIC 😡 COMMENTS 💭 FOUND IN YOUR AUDIO SET ARE AS FOLLOWS:\n")
print(transcriptions)
```

```
THE TOXIC 😡 COMMENTS 💭 FOUND IN YOUR AUDIO SET ARE AS FOLLOWS:

HEY IDIOT THIS IS A WARNING FOR YOU YOU AND YOUR FAMILY CAN BE A VICTIM OF
CIBRE ATTACKS WE ARE GOING TO HACK INTO YOUR SYSTEMS THIS LAST WARNING TO
YOU LUSER
```

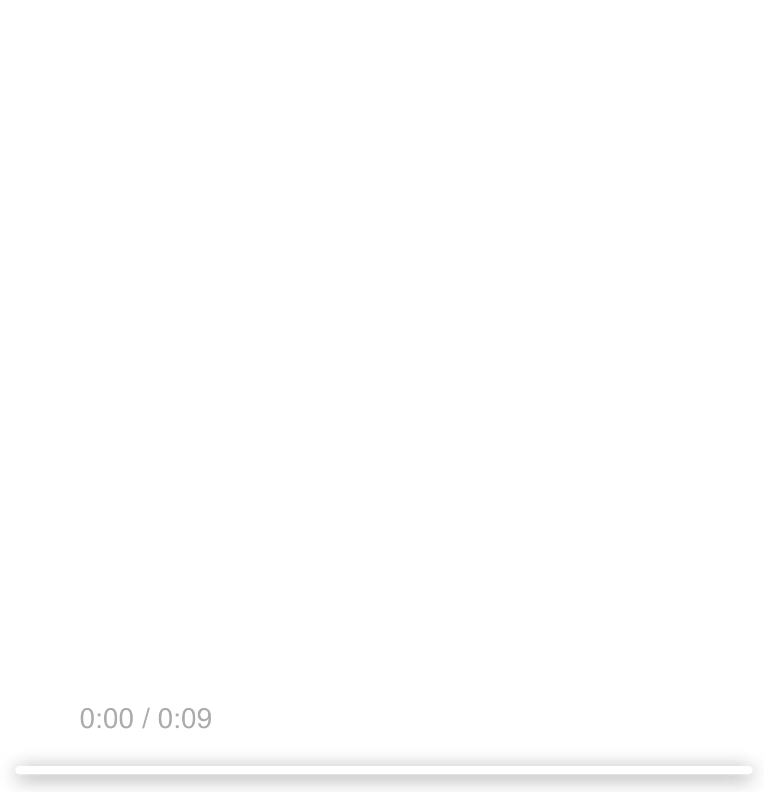
📽 **VIDEO ANALYZER - Helps to analyse large video containing toxic elements, thereby fetching the audio recording from it and automatically saving it in desired formats such as .wav, .mp3 etc. for performing further analysis**

TOXIC VIDEO ANALYSER
CONVERTING THE TOXIC VIDEO MESSAGES INTO .WAV FORMAT FOR FURTHER ANALYSIS
TOXIC

In [ ]:

```
from IPython.display import HTML
from base64 import b64encode
mp4 = open('toxic_video.mp4','rb').read()
data_url = "data:video/mp4;base64," + b64encode(mp4).decode()
HTML("""
<video width=400 controls>
      <source src="%s" type="video/mp4">
</video>
""" % data_url)
```

Out[ ]:



In [ ]:

```
!pip install ffmpeg moviepy
```

```
Requirement already satisfied: ffmpeg in /usr/local/lib/python3.7/dist-pac
kages (1.4)
Requirement already satisfied: moviepy in /usr/local/lib/python3.7/dist-pa
ckages (0.2.3.5)
Requirement already satisfied: decorator<5.0,>=4.0.2 in /usr/local/lib/pyt
hon3.7/dist-packages (from moviepy) (4.4.2)
Requirement already satisfied: tqdm<5.0,>=4.11.2 in /usr/local/lib/python
3.7/dist-packages (from moviepy) (4.64.0)
Requirement already satisfied: imageio<3.0,>=2.1.2 in /usr/local/lib/pytho
n3.7/dist-packages (from moviepy) (2.4.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-pack
ages (from moviepy) (1.21.6)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-pac
kages (from imageio<3.0,>=2.1.2->moviepy) (7.1.2)
```

```
In [ ]:
```

```python
import moviepy.editor as mp
```

```
In [ ]:
```

```python
toxic_video = mp.VideoFileClip(r"toxic_video.mp4")
```

```
In [ ]:
```

```python
toxic_video.audio.write_audiofile(r"extracted_audio_message.wav")
display.Audio("extracted_audio_message.wav", autoplay=True)
```

```
[MoviePy] Writing audio in extracted_audio_message.wav

100%|██████████| 210/210 [00:00<00:00, 2634.83it/s]

[MoviePy] Done.
```

```
Out[ ]:
```

0:09 / 0:09

In [ ]:

```
!pip install folium
!pip install geocoder
```

```
Requirement already satisfied: folium in /usr/local/lib/python3.7/dist-pac
kages (0.8.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-pack
ages (from folium) (1.21.6)
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.7/d
ist-packages (from folium) (0.5.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packag
es (from folium) (1.15.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-p
ackages (from folium) (2.23.0)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-pac
kages (from folium) (2.11.3)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.
7/dist-packages (from jinja2->folium) (2.0.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->folium) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/di
st-packages (from requests->folium) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python
3.7/dist-packages (from requests->folium) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python
3.7/dist-packages (from requests->folium) (2021.10.8)
Requirement already satisfied: geocoder in /usr/local/lib/python3.7/dist-p
ackages (1.38.1)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packag
es (from geocoder) (1.15.0)
Requirement already satisfied: ratelim in /usr/local/lib/python3.7/dist-pa
ckages (from geocoder) (0.1.6)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-p
ackages (from geocoder) (2.23.0)
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-pac
kages (from geocoder) (0.16.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-pack
ages (from geocoder) (7.1.2)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-
packages (from ratelim->geocoder) (4.4.2)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->geocoder) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python
3.7/dist-packages (from requests->geocoder) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/di
st-packages (from requests->geocoder) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python
3.7/dist-packages (from requests->geocoder) (2021.10.8)
```

🌐 IP TRACKER - Helping you to fetch the real time precise location coordinates of the Scammers. 🗺️

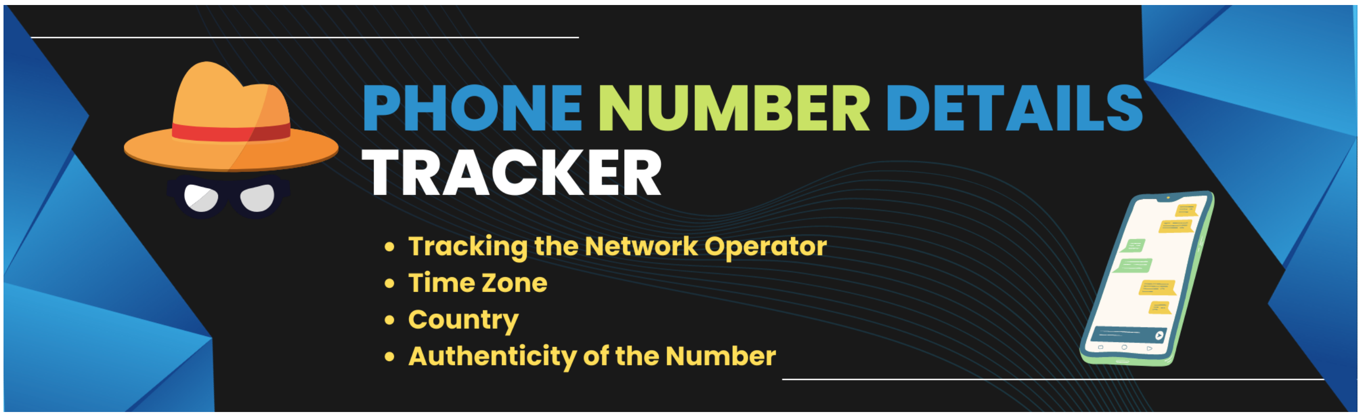
IP ADDRESS TRACKER
Fetching the real time locations of Scammers!
PRABAL MANHAS 20BCS4513 (LEADER)
GIRJANAND TIWARY 20BCS4506
ANURAG KUMAR 20BCS4567

In [ ]:

```
import geocoder
import folium
print("\t\t<<< 🌐 IP TRACKER 🔍 - MINOR PROJECT - PRABAL MANHAS >>>")
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n")

print("> 🔍 TRACING YOUR ENTERED IP ADDRESS ... PLEASE WAIT ⏳ ")

print("> FETCHING IP ADDRESS LOCATION COORDINATES 🗺\n")

print("YOUR LONGITUDE & LATITUDE VALUES ARE AS FOLLOWS: 📝\n")
g = geocoder.ip("117.198.224.22")
myAddress = g.latlng
print(myAddress)

my_map1 = folium.Map(location=myAddress,
                     zoom_start=12)

folium.CircleMarker(location=myAddress,
                    radius=50, popup="TRACKED LOCATION >>>").add_to(my_map1)

folium.Marker(myAddress,
              popup="TRACKED LOCATION >>>").add_to(my_map1)
my_map1.save("my_map.html ")

print("TRACED IP DETAILS SUCCESFULLY ... STORED IN THE HTML FILE 🌐")

print("OPEN HTML FILE TO TRACE ON MAP 🗺\n")
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

print('''\tPRABAL MANHAS 20BCS4513
\t\t ANURAG KUMAR 20BCS4567
\t\t\t GIRJANAND TIWARY 20BCS4506''')
```

```
                <<< 🌐 IP TRACKER 🔍 - MINOR PROJECT - PRABAL MANHAS 
>>>
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

> 🔍 TRACING YOUR ENTERED IP ADDRESS ... PLEASE WAIT ⏳
> FETCHING IP ADDRESS LOCATION COORDINATES 🗺

YOUR LONGITUDE & LATITUDE VALUES ARE AS FOLLOWS: 📝

[32.7353, 74.8617]
TRACED IP DETAILS SUCCESFULLY ... STORED IN THE HTML FILE 🌐
OPEN HTML FILE TO TRACE ON MAP 🗺

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        PRABAL MANHAS 20BCS4513
                 ANURAG KUMAR 20BCS4567
                         GIRJANAND TIWARY 20BCS4506
```

📱 PHONE NUMBER TRACKER - Helps to track all the details associated with Scammer's phone number such as Country, Location, Time Zone, and Authenticity of the Number based on the past activity logs.



In [3]:

```
!pip install phonenumbers
import phonenumbers
from phonenumbers import carrier, geocoder, timezone

mobileNo=input("\n🔢 PLEASE ENTER THE PHONE NUMBER YOU WANT TO TRACE (WITH COUNTRY COD
E) ---> ")
mobileNo=phonenumbers.parse(mobileNo)
print("\nSUCCESFULLY FETCHED THE DETAILS ... 📱🔍\n")
print("🗺 TIMEZONE --> ",timezone.time_zones_for_number(mobileNo))
print("🌐 OPERATOR NAME -->",carrier.name_for_number(mobileNo,"en"))
print("🏠 LOCATION -->",geocoder.description_for_number(mobileNo,"en"))
print("\n✅ CHECKING AUTHENTICTY .... 📞\n")
print("📊 VALIDITY REPORTS ---> ",phonenumbers.is_valid_number(mobileNo))
```

```
Requirement already satisfied: phonenumbers in /usr/local/lib/python3.7/di
st-packages (8.12.48)

🔢 PLEASE ENTER THE PHONE NUMBER YOU WANT TO TRACE (WITH COUNTRY CODE) ---
> +9118001800257

SUCCESFULLY FETCHED THE DETAILS ... 📱🔍

🗺 TIMEZONE -->  ('Asia/Calcutta',)
🌐 OPERATOR NAME -->
🏠 LOCATION --> India

✅ CHECKING AUTHENTICTY .... 📞

📊 VALIDITY REPORTS --->  True
```

📊 TOXICITY ANALYZER ☢️ - Building a toxicity analyser to perform analysis on a large dataset model trained by us, in order to fetch the toxic comments present in it, it can be the tweets posted by a twitter user, comments on Instagram/FB etc.

Also classifying them into their respective categories such as Severe Toxic, Harassment, Bullying, Threats, Obscene, Insult or Identity Hate

PRABAL MANHAS
TOXICITY ANALYZER
TO ANALYZE THE TOXIC ELEMENTS FETCHED FROM AUDIO/VIDEO/TEXT FILES AND FURTHER SORTING THEM INTO CATEGORIES SUCH AS HATE, THREATS, HARASSMENT ETC.
#!?@%!

In [ ]:

```
#IMPORTING THE REQUIRED LIBRARIES AND UPLOADING THE DATASET
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [ ]:

```
# READING THE UPLOADED DATASET IN ONE DATAFRAME
df = pd.read_csv("train.csv")
print(df.shape)
```

```
(159570, 8)
```

In [ ]:

```
# LISTING ALL THE FRAMES PRESENT IN OUR DATASET
print(df.dtypes)
```

```
id                object
comment_text      object
toxic              int64
severe_toxic       int64
obscene            int64
threat             int64
insult             int64
identity_hate      int64
dtype: object
```

In [ ]:

```
# below line causes shuffling of indices, to avoid using train_test_split later
df = df.reindex(np.random.permutation(df.index))
```

Separating the data fields of Comments Label and Outcome Labels

In [ ]:

```
comment = df['comment_text']
print(comment.head())
comment = comment.to_numpy()
```

```
92619                          Wow, why don't you get a life?
98185     Neofuel\nThe Neofuel references ( 4 and 5 curr...
153730                          attack \n\nim not attacking!
100004    querie \nOur town near Bronkhorstspruit recent...
153610    That wasn't me. That IP as well as .58 are Pro...
Name: comment_text, dtype: object
```

In [ ]:

```
label = df[['toxic', 'severe_toxic' , 'obscene' , 'threat' , 'insult' , 'identity_hate'
]]
print(label.head())
label = label.to_numpy()
```

```
        toxic  severe_toxic  obscene  threat  insult  identity_hate
92619       0             0        0       0       0              0
98185       0             0        0       0       0              0
153730      0             0        0       0       0              0
100004      0             0        0       0       0              0
153610      0             0        0       0       0              0
```

Let us find out the frequency of occurence of multilabelled data ct1 counts samples having atleast one label ct2 counts samples having 2 or more than 2 labels

In [ ]:

```
ct1,ct2 = 0,0
for i in range(label.shape[0]):
    ct = np.count_nonzero(label[i])
    if ct :
        ct1 = ct1+1
    if ct>1 :
        ct2 = ct2+1
print(ct1)
print(ct2)
```

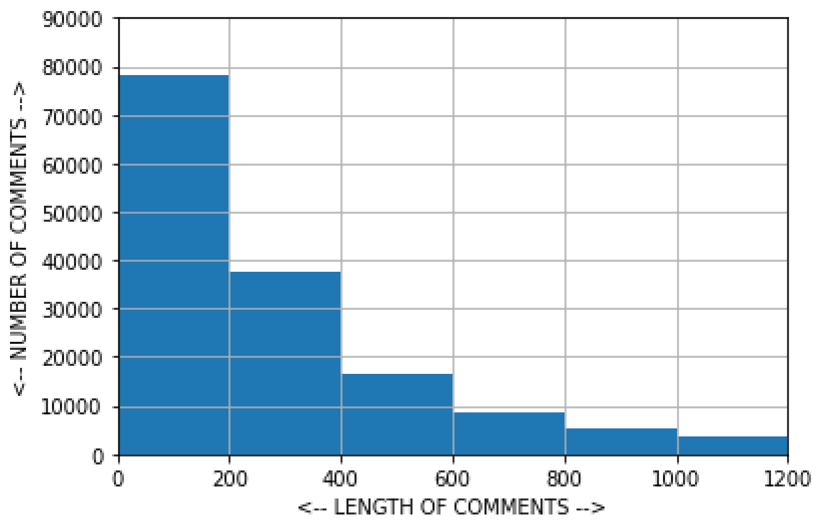
```
16224
9864
```

DATA VISUALISATION - To get insights about comments, length, number of comments etc.

```
In [ ]:
```

```
# Let us analyse the no. of comments having lengths varying from 0 to 1200
x = [len(comment[i]) for i in range(comment.shape[0])]

print('AVERAGE COMMENT LENGTH {:.3f}'.format(sum(x)/len(x)) )
bins = [1,200,400,600,800,1000,1200]
plt.hist(x, bins=bins)
plt.xlabel('<-- LENGTH OF COMMENTS -->')
plt.ylabel('<-- NUMBER OF COMMENTS -->')
plt.axis([0, 1200, 0, 90000])
plt.grid(True)
plt.show()
```

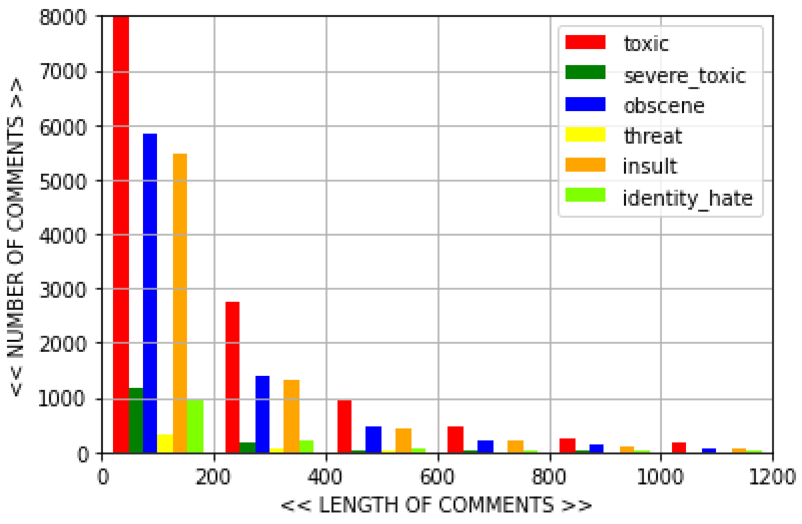
```
AVERAGE COMMENT LENGTH 393.542
```



Number of comments classified as toxic,severe_toxic,....etc depending on their lengths

In [ ]:

```
y = np.zeros(label.shape)
for ix in range(comment.shape[0]):
    l = len(comment[ix])
    if label[ix][0] :
        y[ix][0] = l
    if label[ix][1] :
        y[ix][1] = l
    if label[ix][2] :
        y[ix][2] = l
    if label[ix][3] :
        y[ix][3] = l
    if label[ix][4] :
        y[ix][4] = l
    if label[ix][5] :
        y[ix][5] = l

label
labelsplt = ['toxic','severe_toxic','obscene','threat','insult','identity_hate']
color = ['red','green','blue','yellow','orange','chartreuse']
plt.hist(y,bins = bins,label = labelsplt,color = color)
plt.axis([0, 1200, 0, 8000])
plt.xlabel('<< LENGTH OF COMMENTS >>')
plt.ylabel('<< NUMBER OF COMMENTS >>')
plt.legend()
plt.grid(True)
plt.show()
```



In [1]:

```
!pip install detoxify
```

```
Collecting detoxify
  Downloading detoxify-0.5.0-py3-none-any.whl (12 kB)
Collecting sentencepiece>=0.1.94
  Downloading sentencepiece-0.1.96-cp37-cp37m-manylinux_2_17_x86_64.manyli
nux2014_x86_64.whl (1.2 MB)
     |████████████████████████████████| 1.2 MB 5.2 MB/s
Requirement already satisfied: torch>=1.7.0 in /usr/local/lib/python3.7/di
st-packages (from detoxify) (1.11.0+cu113)
Collecting transformers!=4.18.0
  Downloading transformers-4.19.2-py3-none-any.whl (4.2 MB)
     |████████████████████████████████| 4.2 MB 43.5 MB/s
Requirement already satisfied: typing-extensions in /usr/local/lib/python
3.7/dist-packages (from torch>=1.7.0->detoxify) (4.2.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-p
ackages (from transformers!=4.18.0->detoxify) (2.23.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-p
ackages (from transformers!=4.18.0->detoxify) (3.7.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python
3.7/dist-packages (from transformers!=4.18.0->detoxify) (2019.12.20)
Collecting pyyaml>=5.1
  Downloading PyYAML-6.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_6
4.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (596 kB)
     |████████████████████████████████| 596 kB 48.2 MB/s
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist
-packages (from transformers!=4.18.0->detoxify) (4.64.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.
7/dist-packages (from transformers!=4.18.0->detoxify) (21.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dis
t-packages (from transformers!=4.18.0->detoxify) (1.21.6)
Collecting tokenizers!=0.11.3,<0.13,>=0.11.1
  Downloading tokenizers-0.12.1-cp37-cp37m-manylinux_2_12_x86_64.manylinux
2010_x86_64.whl (6.6 MB)
     |████████████████████████████████| 6.6 MB 29.9 MB/s
Requirement already satisfied: importlib-metadata in /usr/local/lib/python
3.7/dist-packages (from transformers!=4.18.0->detoxify) (4.11.3)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.6.0-py3-none-any.whl (84 kB)
     |████████████████████████████████| 84 kB 3.0 MB/s
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/
python3.7/dist-packages (from packaging>=20.0->transformers!=4.18.0->detox
ify) (3.0.9)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->transformers!=4.18.0->detoxify) (3.8.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers!=4.18.
0->detoxify) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python
3.7/dist-packages (from requests->transformers!=4.18.0->detoxify) (2021.1
0.8)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/di
st-packages (from requests->transformers!=4.18.0->detoxify) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python
3.7/dist-packages (from requests->transformers!=4.18.0->detoxify) (3.0.4)
Installing collected packages: pyyaml, tokenizers, huggingface-hub, transf
ormers, sentencepiece, detoxify
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed detoxify-0.5.0 huggingface-hub-0.6.0 pyyaml-6.0 sen
tencepiece-0.1.96 tokenizers-0.12.1 transformers-4.19.2
```

In [4]:

```
from detoxify import Detoxify
```

In [5]:

```
predictor = Detoxify('multilingual')
```

```
Downloading: "https://github.com/unitaryai/detoxify/releases/download/v0.4
-alpha/multilingual_debiased-0b549669.ckpt" to /root/.cache/torch/hub/chec
kpoints/multilingual_debiased-0b549669.ckpt
```

In [6]:

```
predictor.predict('you are such an idiot shut up!')
```

Out[6]:

```
{'identity_attack': 0.0021715963,
 'insult': 0.98686695,
 'obscene': 0.24050981,
 'severe_toxicity': 0.004830556,
 'sexual_explicit': 0.0057667117,
 'threat': 0.0017694455,
 'toxicity': 0.9974275}
```

In [7]:

```
predictor.predict('Eres una idiota callate')
```

Out[7]:

```
{'identity_attack': 0.0070628854,
 'insult': 0.4857168,
 'obscene': 0.13782535,
 'severe_toxicity': 0.0044060494,
 'sexual_explicit': 0.0066691204,
 'threat': 0.0021975825,
 'toxicity': 0.9916694}
```

In [8]:

```
demo_comments= [
        'Eres una idiota callate',
        'How much is this bag?',
        'I am going to hack you fool',
        'Thanks mate see you soon',
        'I will hurt you'
]
```

In [9]:

```
for comments in demo_comments:
    results = predictor.predict(comments)
    print (results)
```

```
{'toxicity': 0.9916694, 'severe_toxicity': 0.0044060494, 'obscene': 0.1378
2535, 'identity_attack': 0.0070628854, 'insult': 0.4857168, 'threat': 0.00
21975825, 'sexual_explicit': 0.0066691204}
{'toxicity': 0.0016920738, 'severe_toxicity': 1.4579835e-05, 'obscene': 0.
00018599258, 'identity_attack': 7.167022e-05, 'insult': 0.0005639618, 'thr
eat': 3.5181794e-05, 'sexual_explicit': 2.974496e-05}
{'toxicity': 0.99755245, 'severe_toxicity': 0.052452806, 'obscene': 0.4155
392, 'identity_attack': 0.010864722, 'insult': 0.9536885, 'threat': 0.8267
7287, 'sexual_explicit': 0.05452031}
{'toxicity': 0.0006031596, 'severe_toxicity': 3.784469e-05, 'obscene': 0.0
0024074431, 'identity_attack': 7.6673714e-05, 'insult': 0.0003838271, 'thr
eat': 5.2895502e-05, 'sexual_explicit': 3.2755106e-05}
{'toxicity': 0.9314441, 'severe_toxicity': 0.00785032, 'obscene': 0.042038
243, 'identity_attack': 0.00375442, 'insult': 0.036926, 'threat': 0.767486
7, 'sexual_explicit': 0.017453872}
```