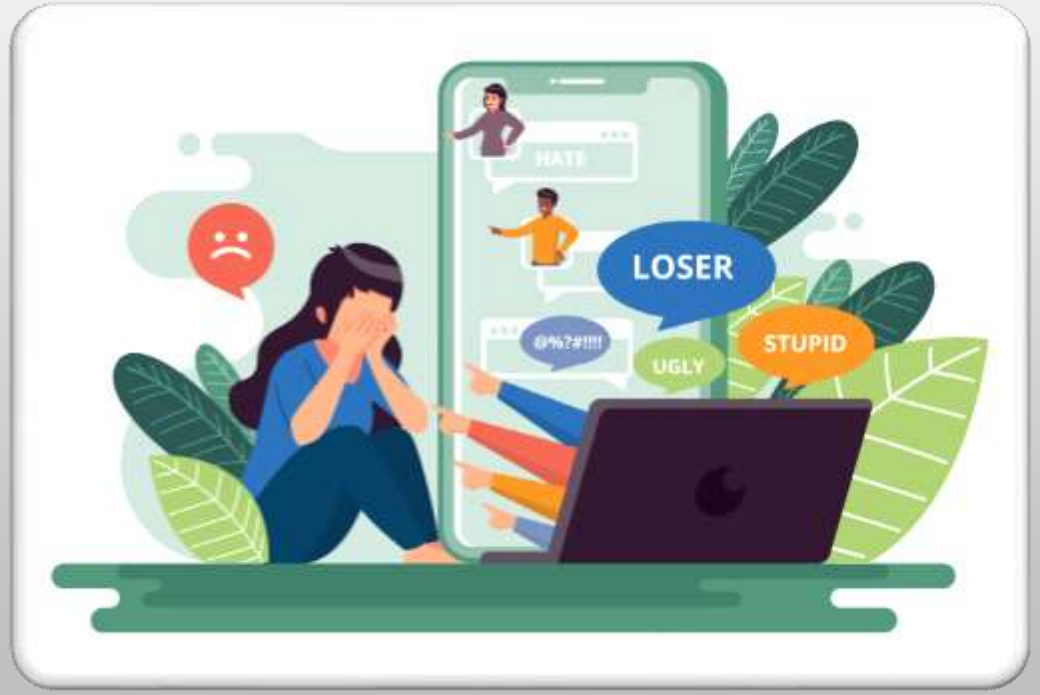
CYBERBULLYING CLASSIFICATION USING NAIVE BAYES CLASSIFIER

- PREPARED BY:-

- SHRINIVAS PATIL
- NISHANTKUAMR ASODARIYA

CONTENTS

- OUR PROJECT TALKS ABOUT ANALYZING THE SOCIAL MEDIA POSTS AND ANALYZE IT FOR CYBERBULLYING CONTEXT.
- I CYBERBULLYING IS A RELATIVELY NEW PHENOMENON. THE DIGITAL NATURE OF IT ALLOWS A PERMANENT RECORD OF NEGATIVE INFORMATION THAT HAS THE POTENTIAL TO AFFECT HUMANS CURRENT AND FUTURE PSYCHOLOGICAL AND EMOTIONAL STATES.



APPROACH

ABOUT THE DATASET

- WE CHOSE THE DATASET FROM KAGGLE DATASETS, THE LINK FOR DATASETS IS

  https://www.kaggle.com/code/vincentgupo/classifying-cyberbullying-94-accuracy/data?select=cyberbullying_tweets.csv

- THERE ARE 2 COLUMNS IN THIS DATASET.
  1. TWEET_TEXT
  2. CYBERBULLYING_TYPE

  TWEET_TEXT CONTAINS THE TWEETS POSTED BY USERS ON TWITTER. CYBERBULLYING_TYPE CONTAINS 5 CLASSES LIKE RELIGION, GENDER, AGE, NOT CYBERBULLYING, ETHNICITY.

CLEANING THE DATA

The first VITAL task is to clean the data in the dataset. The tweets contains username, links, special characters and digits.

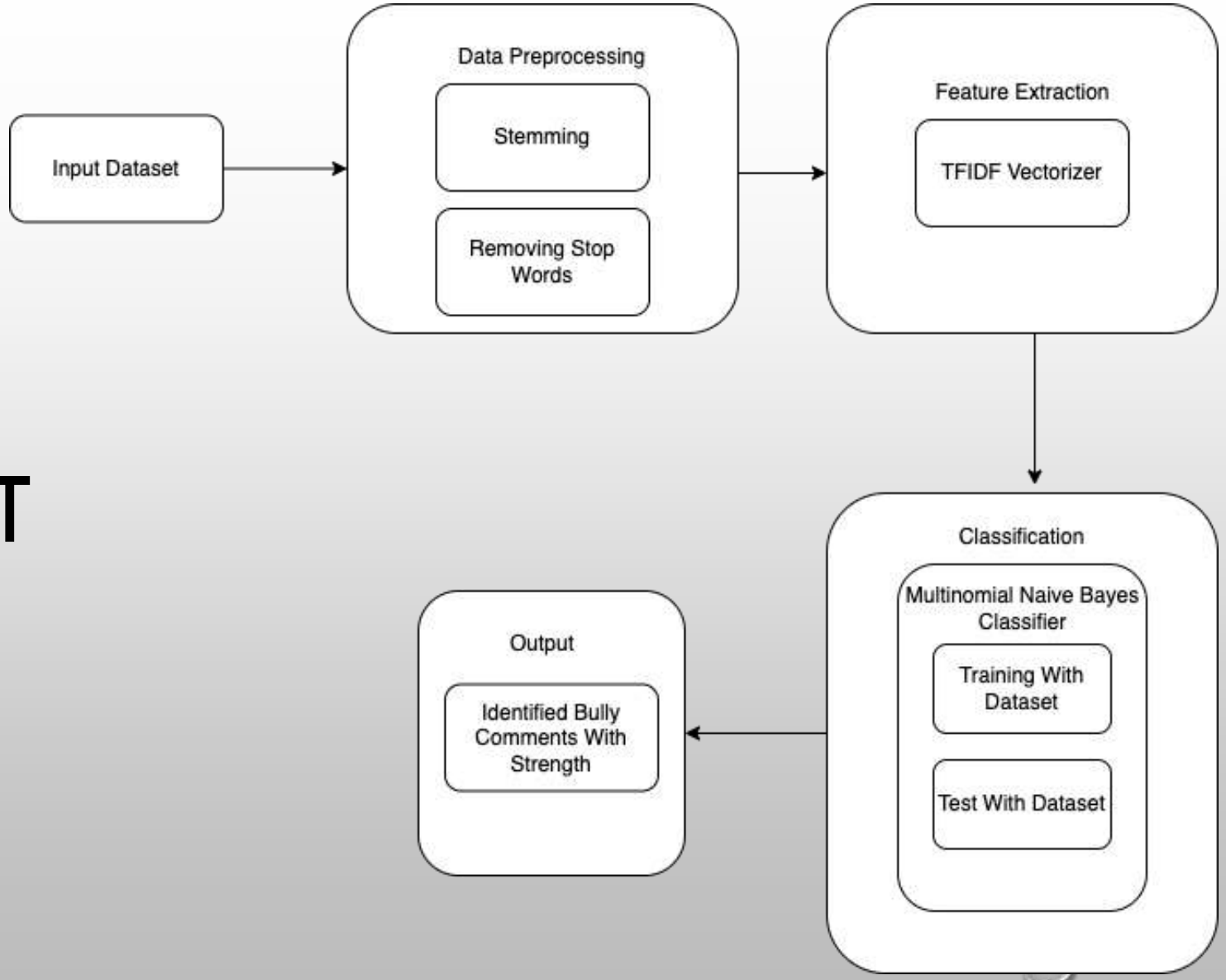
A clean dataset helps machine learning model to understand the pattern and to increase the accuracy.

The next step to find the frequency of every word compared to each sentence. This process is done by Tfldf Vectorizer.

APPLYING BAYES CLASSIFIER ..

- TO APPLY ANY MODEL, WE FIRST DIVIDE THE DATA INTO TRAINING DATA AND TESTING DATA
- USUALLY, **80%** OF THE DATA IS THE TRAINING DATA AND **20%** IS THE TESTING DATA.
- THE TRAINING DATA IS USED TO TRAIN THE MODEL AND THE MODEL IS TESTED USING THE TESTING DATA.

FLOWCHART


Input Dataset
Data Preprocessing
Stemming
Removing Stop Words
Feature Extraction
TFIDF Vectorizer
Classification
Multinomial Naive Bayes Classifier
Training With Dataset
Test With Dataset
Output
Identified Bully Comments With Strength

IMPLEMENTATION

- Importing the Python required libraries like pandas, matplotlib, seaborn, scikit learn

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
```

In [2]:

```
#import the dataset stored in drive
df = pd.read_csv('/kaggle/input/cyberbullying-classification/cyberbullying_tweets.csv')
df.head()
```

Out[2]:

	tweet_text	cyberbullying_type
0	In other words #katandandre, your food was cra...	not_cyberbullying
1	Why is #aussietv so white? #MKR #theblock #ImA...	not_cyberbullying
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying

- Loading the data from a csv file and checking first 5 records in dataset using head method.

DATA CLEANING & OUTPUTS

In [8]:

```
#preprocessing the input features
ps = PorterStemmer()
corpus=[]
def remove_emoji(string):
    emoji_pattern = re.compile("["
                           u"\U0001F600-\U0001F64F"  # emoticons
                           u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                           u"\U0001F680-\U0001F6FF"  # transport & map symbols
                           u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                           u"\U00002702-\U000027B0"
                           u"\U000024C2-\U0001F251"
                           "]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', string)
for i in range(len(df)):
  text = re.sub(r"(?:\@|https?\://)\S+", "", df['tweet_text'][i])
  text = re.sub(r'[^\x00-\x7f]',r'', text)
  text = re.sub("\s\s+" , " ", text)
  text = remove_emoji(text)
  text= " ".join(word.strip() for word in re.split('#(?!(?:hashtag)\b)[\w-]+(?=(?:\s+#[\w-]+)*\s*$)', text)) #remove
  text = " ".join(word.strip() for word in re.split('#|_', text)) #remove hashtag middle of sentence
  text = re.sub('[^a-zA-Z]',' ',text)
  text = text.lower()
  text = text.split()
  text = [ps.stem(words) for words in text if words not in stopwords.words('english')]
  text = " ".join(text)
  corpus.append(text)
df['tweet_clean']=corpus
```

In [9]:

```
df['tweet_clean']
```

Out[9]:

```
0                          word katandandr food crapilici
1                                           aussietv white
2                          classi whore red velvet cupcak
3        meh p thank head concern anoth angri dude twitter
4        isi account pretend kurdish account like islam...
                                 ...
47687    black ppl expect anyth depend anyth yet free p...
47688    turner withhold disappoint turner call court a...
47689    swear god dumb nigger bitch got bleach hair re...
47690    yea fuck rt your nigger fuck unfollow fuck dum...
47691    bro u gotta chill rt dog fuck kp dumb nigger b...
Name: tweet_clean, Length: 47692, dtype: object
```

Tweets contains digits, special characters, links, username, emojis and multiple spaces which can affect the accuracy of machine learning algorithms. This code removes the all this irregularities and clean the data. This clean data will be stored in new column of dataset. Each tweet has most common words those will be removed from the text using porter stemmer. This process is called stemming.

DATA CLEANING & OUTPUTS

```
In [17]:
df.sort_values(by="tweet_len",ascending=False)
```

Out[17]:

	tweet_text	cyberbullying_type	tweet_clean	tweet_len
44035	You so black and white trying to live like a n...	ethnicity	black white tri live like nigger pahahahaha co...	188
45165	@hermdiggz: *@tayyoung_: FUCK OBAMA, dumb ass ...	ethnicity	fuck obama dumb ass nigger bitch lt whore smh ...	165
33724	... I don't feel guilty for killing him, I jus...	age	feel guilti kill feel guilti enjoy torment sin...	139
1317	@EurekAlertAAAS: Researchers push to import to...	not_cyberbullying	research push import top anti bulli program us...	139
47037	@Purely_Ambition: Sooo mad. RT @TracePeterson ...	ethnicity	sooo mad rt fuck obama dumb nigger go switzerl...	128
...	...	...	...	...
5069	@Louie__88 weres it at	not_cyberbullying	were	1
1672	@halalflaws @AMohedin @islamdefense @haroonsty...	not_cyberbullying	say	1
8979	@BlackOpal80 I'm blocked.	gender	block	1
227	@EvvyKube not sure.	not_cyberbullying	sure	1
10	@Jord_Is_Dead http://t.co/UsQlnYW5Gn	not_cyberbullying		0

38715 rows x 4 columns

```
In [18]:
df = df[df['tweet_len']>3]
df = df[df['tweet_len'] < 100]
```

```
In [19]:
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Label'] = le.fit_transform(df['cyberbullying_type'])
```

```
In [20]:
df
```

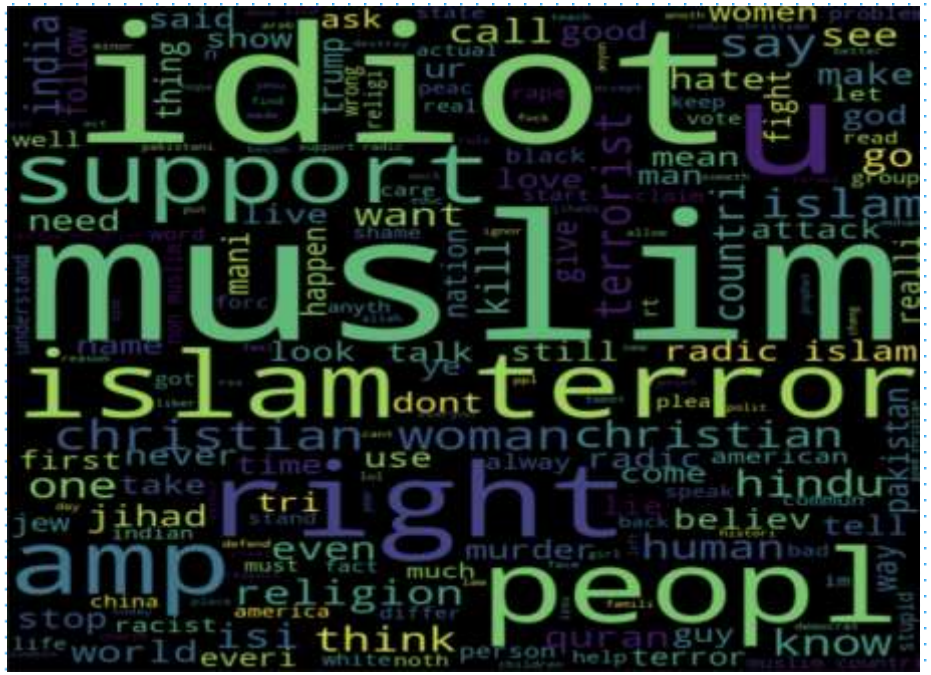
Out[20]:

	tweet_text	cyberbullying_type	tweet_clean	tweet_len	Label
0	In other words #katandandre, your food was cra...	not_cyberbullying	word katandandr food crapilici	4	3
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying	classi whore red velvet cupcak	5	3
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying	meh p thank head concern anoth angri dude twitter	9	3
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying	isi account pretend kurdish account like islam...	8	3
5	@Raja5aab @Quickieleaks Yes, the test of god i...	not_cyberbullying	ye test god good bad indiffer weird whatev pro...	11	3

Firstly, We found out length of each tweet to analyze what can be the maximum and minimum length of each tweet after cleaning. We chose tweets having length between 3 to 100. After that We encoded the types to machine readable format that is integer.

WORD CLOUDS

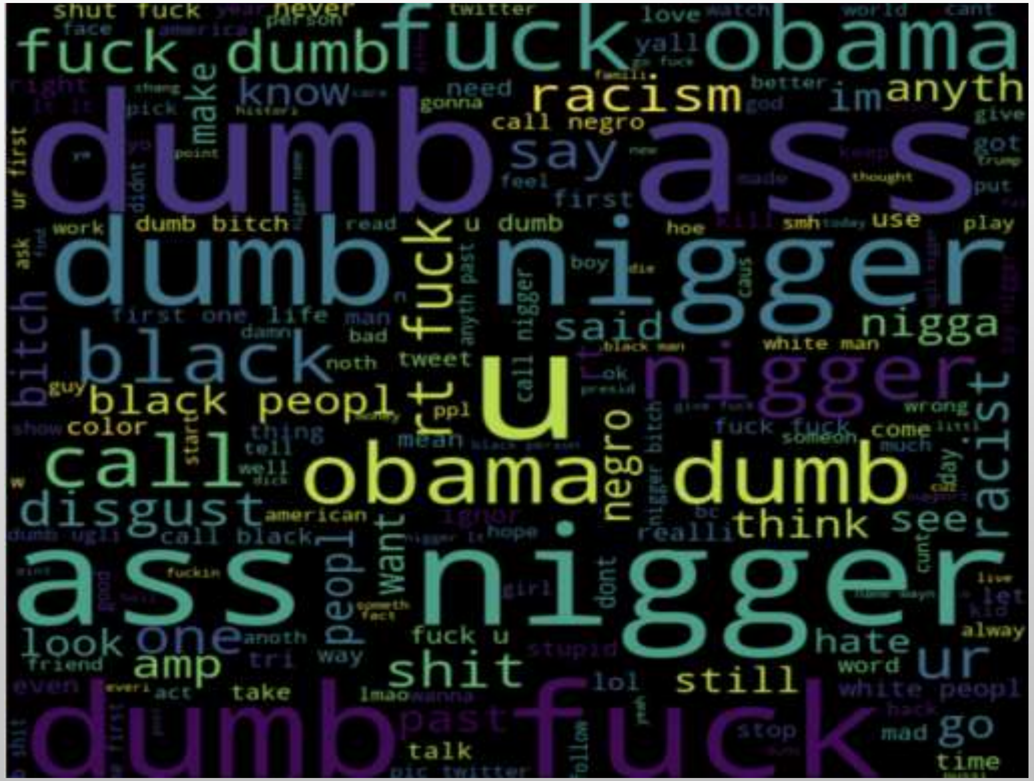
```
In [23]:

for c in range(len(le.classes_)):
    string = ""
    for i in df[df.Label == c].tweet_clean.values:
        string = string + " " + i.strip()
    wordcloud = WordCloud(width = 800, height = 800,
                background_color ='black',
                min_font_size = 10).generate(string)
    plt.figure(figsize = (8, 8), facecolor = None)
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.tight_layout(pad = 0)
```



Word clouds are the visual representation of words used in tweets. The bigger the size of each word more the frequency or importance of that word.

WORD CLOUDS

high school
school bulli
girl bulli
bulli high
girl
bulli
peopl
friend
got bulli
middl school
school
look
one
girl high
see
start
lot
let
even
rememb
got
tell
said
post
talk
think
stop
alway
know
say
class
love
end
much
life
never
good
right
call
person
caus
guy
amp
feel
want
lol
everyon
im
realli
need
teacher
time
meanmade
take
use
back
day
kid
tri
go
didnt
fuck
come
someon
year
thing
work
make
yall

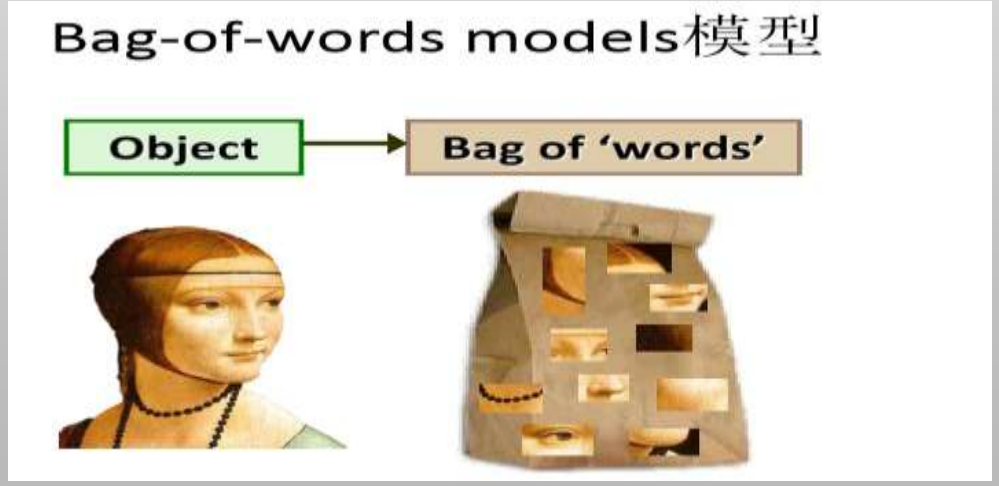


dumb
ass
fuck
obama
dumb nigger
ass nigger
dumb fuck
fuck dumb
racism
know
say
anyth
im
black
nigger
nigga
u
said
fuck
call
obama
dumb
negro
racist
black peopl
disgust
think
see
one
look
amp
shit
still
hate
ur
peopl
want
bitch
rt
go
time

Bag of Words and TFIDF Vectorizer

- Both Bag of Words and TFIDF are preprocessing techniques that generate numeric from from text data.
- Bag of words converts the text into fixed length vectors by converting how many times each word appears in sentence.
- consider an example,
  - Text processing is necessary. [1,1,1,1,0,0]
  - Text processing is necessary and important.[1,1,1,1,1,1]

- TFIDF vectorizer processes text by calculating how many times that word appears in sentence and counterbalanced by total number of sentences in which it is present.
- Consider an example
  - Text processing is necessary. [0,0,0.4678932,0.567434,0.578903,0]
  - Text processing is necessary and important.[0.456734,0.564457,0,0,0.34245,0]



UP SAMPLING

- ***UPSAMPLING*** IS A PROCEDURE WHERE SYNTHETICALLY GENERATED DATA POINTS (CORRESPONDING TO MINORITY CLASS) ARE INJECTED INTO THE DATASET. AFTER THIS PROCESS, THE COUNTS OF BOTH LABELS ARE ALMOST THE SAME.
- THIS EQUALIZATION PROCEDURE PREVENTS THE MODEL FROM INCLINING TOWARDS THE MAJORITY CLASS. FURTHERMORE, THE INTERACTION(BOUNDARY LINE)BETWEEN THE TARGET CLASSES REMAINS UNALTERED.

```
In [27]: y_train.value_counts()

Out[27]: 4    6308
         0    6260
         1    6170
         2    5790
         3    4932
         Name: Label, dtype: int64
```

Before

```
In [28]: from imblearn.over_sampling import SMOTE
         vc = y_train.value_counts()
         while (vc[0] != vc[4]) or (vc[0] != vc[2]) or (vc[0] != vc[3]) or (vc[0] != vc[1]):
             smote = SMOTE(sampling_strategy='minority')
             X_train, y_train = smote.fit_resample(X_train, y_train)
             vc = y_train.value_counts()
         y_train.value_counts()

Out[28]: 1    6308
         3    6308
         4    6308
         2    6308
         0    6308
         Name: Label, dtype: int64
```

After

NAÏVE BAYES VS MULTINOMIAL NAÏVE BAYES

- NAIVE BAYES IS USED WHEN VARIABLES ARE CONTINUOUS IN NATURE. IT ASSUMES THAT ALL THE VARIABLES HAVE A NORMAL DISTRIBUTION. SO, IF YOU HAVE SOME VARIABLES WHICH DO NOT HAVE THIS PROPERTY, YOU MIGHT WANT TO TRANSFORM THEM TO THE FEATURES HAVING DISTRIBUTION NORMAL.
- MULTINOMIAL NB IS USED WHEN THE FEATURES REPRESENT THE FREQUENCY. SUPPOSE YOU HAVE A TEXT DOCUMENT, AND YOU EXTRACT ALL THE UNIQUE WORDS AND CREATE MULTIPLE FEATURES WHERE EACH FEATURE REPRESENTS THE COUNT OF THE WORD IN THE DOCUMENT. IN SUCH A CASE, WE HAVE A FREQUENCY AS A FEATURE. IN SUCH A SCENARIO, WE USE MULTINOMIAL NAIVE BAYES.

Accuracy Score
0.506856754921928
classification Report

	precision	recall	f1-score	support
0	0.51	0.32	0.40	1565
1	0.72	0.65	0.68	1543
2	0.36	0.76	0.48	1447
3	0.45	0.34	0.39	1233
4	0.73	0.45	0.56	1577
accuracy			0.51	7365
macro avg	0.55	0.50	0.50	7365
weighted avg	0.56	0.51	0.51	7365

In [34]:
```
nb_pred = nb_clf.predict(X_test)
print(accuracy_score(y_test,nb_pred))
```

0.8562118126272913

In [35]:
```
print(classification_report(y_test,nb_pred))
```

	precision	recall	f1-score	support
0	0.80	0.98	0.88	1565
1	0.91	0.91	0.91	1543
2	0.88	0.86	0.87	1447
3	0.84	0.51	0.63	1233
4	0.86	0.96	0.90	1577
accuracy			0.86	7365
macro avg	0.86	0.84	0.84	7365
weighted avg	0.86	0.86	0.85	7365

FUTURE SCOPE

- We can save this model in .pkl file and deploy it in web flask application for use.
- We can improve the accuracy up to 95% to 96% by using Word2Vec or LSTM neural networks.
- We can use on social media platforms like twitter, Instagram, Facebook and Snapchat to find out type of cyberbullying. It will help to implement respective laws and actions.

print(”Thank You”)