

```
In [1]: # Importing Necessary Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: import re
import nltk
import string
import nlp_utils
import contractions
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize,sent_tokenize
from nltk.stem import PorterStemmer, LancasterStemmer, SnowballStemmer
```

```
In [3]: df=pd.read_csv('train.csv')
# Reading train dataset.
```

```
In [4]: df
# Loading dataset.
```

		<b>id</b>	<b>comment_text</b>	<b>toxic</b>	<b>severe_toxic</b>	<b>obscene</b>	<b>threat</b>	<b>insult</b>	<b>identity_h</b>
<b>0</b>	0000997932d777bf		Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
<b>1</b>	000103f0d9cfb60f		D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
<b>2</b>	000113f07ec002fd		Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
<b>3</b>	0001b41b1c6bb37e		"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
<b>4</b>	0001d958c54c6e35		You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
<b>159566</b>	ffe987279560d7ff		"::::And for the second time of asking, when ...	0	0	0	0	0	0
<b>159567</b>	ffa4adeee384e90		You should be ashamed of yourself \n\nThat is ...	0	0	0	0	0	0

	<b>id</b>	<b>comment_text</b>	<b>toxic</b>	<b>severe_toxic</b>	<b>obscene</b>	<b>threat</b>	<b>insult</b>	<b>identity_hate</b>
<b>159568</b>	ffee36eab5c267c9	Spitzer \n\nUmm, theres no actual article for ...	0	0	0	0	0	0
<b>159569</b>	fff125370e4aaaf3	And it looks like it was actually you who put ...	0	0	0	0	0	0
<b>159570</b>	fff46fc426af1f9a	"\nAnd ... I really don't think you understand...	0	0	0	0	0	0

159571 rows × 8 columns



In [5]:

```
df.info()
# Information about the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159571 entries, 0 to 159570
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               159571 non-null   object 
 1   comment_text     159571 non-null   object 
 2   toxic            159571 non-null   int64  
 3   severe_toxic    159571 non-null   int64  
 4   obscene          159571 non-null   int64  
 5   threat            159571 non-null   int64  
 6   insult            159571 non-null   int64  
 7   identity_hate   159571 non-null   int64  
dtypes: int64(6), object(2)
memory usage: 9.7+ MB
```

In [6]:

```
df.isnull().sum()
# There are no null values.
```

Out[6]:

```
id                0
comment_text      0
toxic             0
severe_toxic     0
obscene           0
threat            0
insult            0
identity_hate    0
dtype: int64
```

In [7]:

```
df['toxic'].value_counts()
# Counts of toxic and non toxic sentences.
```

Out[7]:

```
0    144277
1    15294
Name: toxic, dtype: int64
```

In [8]:

```
df['severe_toxic'].value_counts()
# Counts of severe_toxic and non severe_toxic sentences.
```

Out[8]:

```
0    157976
1    1595
Name: severe_toxic, dtype: int64
```

```
In [9]: df['obscene'].value_counts()
# Counts of obscene and non obscene sentences.
```

```
Out[9]: 0    151122
1     8449
Name: obscene, dtype: int64
```

```
In [10]: df['threat'].value_counts()
# Counts of threat and non threatening sentences.
```

```
Out[10]: 0    159093
1      478
Name: threat, dtype: int64
```

```
In [11]: df['insult'].value_counts()
# Counts of insult and non insulting sentences.
```

```
Out[11]: 0    151694
1      7877
Name: insult, dtype: int64
```

```
In [12]: df['identity_hate'].value_counts()
# Counts of toxic and non identity_hate sentences.
```

```
Out[12]: 0    158166
1      1405
Name: identity_hate, dtype: int64
```

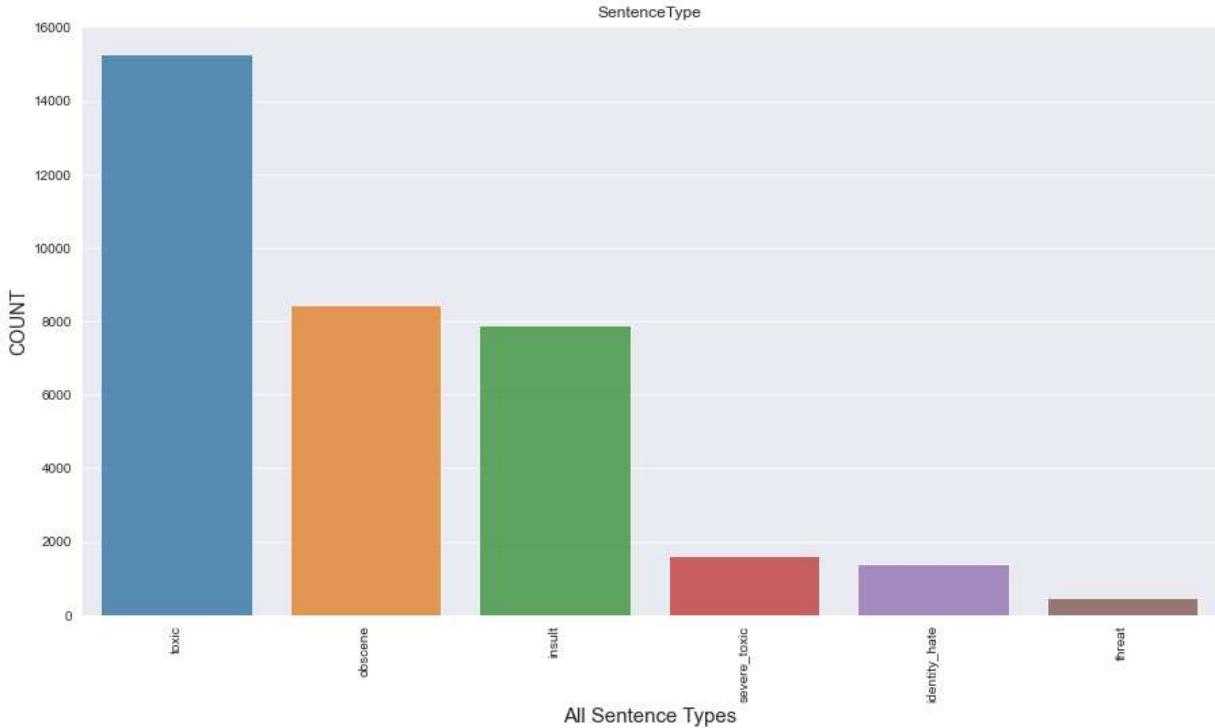
## Data Visualization

```
In [13]: sentencetype_graph=df.iloc[:,2:].sum()
# Using only numeric columns.
```

```
In [14]: sentencetype_graph
```

```
Out[14]: toxic          15294
severe_toxic      1595
obscene           8449
threat            478
insult            7877
identity_hate     1405
dtype: int64
```

```
In [15]: sns.set_style("darkgrid")
ls=sentencetype_graph.sort_values(ascending=False)
plt.figure(figsize=(15,8))
temp =sns.barplot(ls.index, ls.values, alpha=0.8)
plt.title('SentenceType')
plt.ylabel('COUNT', fontsize=14)
plt.xlabel('All Sentence Types', fontsize=15)
temp.set_xticklabels(rotation=90,labels=ls.index,fontsize=10)
plt.show()
```



In [16]: `# There are a many toxic sentences followed by obscene sentences and very few threat`

## Text Pre-processing

In [17]: `df['comment_text'][10]`

Out[17]:  

```
'"\nFair use rationale for Image:Wonju.jpg\n\nThanks for uploading Image:Wonju.jpg. I notice the image page specifies that the image is being used under fair use but there is no explanation or rationale as to why its use in Wikipedia articles constitutes fair use. In addition to the boilerplate fair use template, you must also write out on the image description page a specific explanation or rationale for why using this image in each article is consistent with fair use.\n\nPlease go to the image description page and edit it to include a fair use rationale.\n\nIf you have uploaded other fair use media, consider checking that you have specified the fair use rationale on those pages too. You can find a list of '\image\' pages you have edited by clicking on the ""my contributions"" link (it is located at the very top of any Wikipedia page when you are logged in), and then selecting ""Image"" from the dropdown box. Note that any fair use images uploaded after 4 May, 2006, and lacking such an explanation will be deleted one week after they have been uploaded, as described on criteria for speedy deletion. If you have any questions please ask them at the Media copyright questions page. Thank you. (talk • contribs • ) \nUnspecified source for Image: Wonju.jpg\n\nThanks for uploading Image:Wonju.jpg. I noticed that the file\'s description page currently doesn\'t specify who created the content, so the copyright status is unclear. If you did not create this file yourself, then you will need to specify the owner of the copyright. If you obtained it from a website, then a link to the website from which it was taken, together with a restatement of that website\'s terms of use of its content, is usually sufficient information. However, if the copyright holder is different from the website\'s publisher, then their copyright should also be acknowledged.\n\nAs well as adding the source, please add a proper copyright licensing tag if the file doesn\'t have one already. If you created/took the picture, audio, or video then the tag can be used to release it under the GFDL. If you believe the media meets the criteria at Wikipedia:Fair use, use a tag such as or one of the other tags listed at Wikipedia:Image copyright tags#Fair use. See Wikipedia:Image copyright tags for the full list of copyright tags that you can use.\n\nIf you have uploaded other files, consider checking that you have specified their source and tagged them, too. You can find a list of files you have uploaded by following [ this link]. Unsourced and untagged images may be deleted one week after they have been tagged, as described on criteria for speedy deletion. If the image is copyrighted under a non-free license (per Wikipedia:Fair use) then the image will be deleted 48 hours later.'
```

s after . If you have any questions please ask them at the Media copyright questions page. Thank you. (talk • contribs • ) ''

In [18]:

```
import re
import string
```

In [19]:

```
alphanumeric = lambda x: re.sub('\w*\d\w*', ' ', x)
punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ', x.lower())
remove_n = lambda x: re.sub("\n", " ", x)
remove_non_ascii = lambda x: re.sub(r'[^\x00-\x7f]',r' ', x)
df['comment_text'] = df['comment_text'].map(alphanumeric).map(punc_lower).map(remove_n)
# Removing special characters
```

In [20]:

```
Insulting_comment_df=df.loc[:,['id','comment_text','insult']]
# Creating insult dataframe
```

In [21]:

```
IdentityHate_comment_df=df.loc[:,['id','comment_text','identity_hate']]
# Creating identityhate dataframe
```

In [22]:

```
Obscene_comment_df=df.loc[:,['id','comment_text','obscene']]
# Creating obscene comment dataframe
```

In [23]:

```
Threatening_comment_df=df.loc[:,['id','comment_text','threat']]
# Creating threatening dataframe
```

In [24]:

```
Severetoxic_comment_df=df.loc[:,['id','comment_text','severe_toxic']]
# Creating severotoxic dataframe
```

In [25]:

```
Toxic_comment_df=df.loc[:,['id','comment_text','toxic']]
# Creating toxic dataframe
```

In [26]:

```
# Subset datasets.
```

In [27]:

```
Severetoxic_comment_df
```

Out[27]:

	<b>id</b>	<b>comment_text</b>	<b>severe_toxic</b>
<b>0</b>	0000997932d777bf	explanation why the edits made under my userna...	0
<b>1</b>	000103f0d9cfb60f	d aww he matches this background colour i m s...	0
<b>2</b>	000113f07ec002fd	hey man i m really not trying to edit war it...	0
<b>3</b>	0001b41b1c6bb37e	more i can t make any real suggestions on im...	0
<b>4</b>	0001d958c54c6e35	you sir are my hero any chance you remember...	0
...	...	...	...
<b>159566</b>	ffe987279560d7ff	and for the second time of asking when ...	0
<b>159567</b>	ffea4adeee384e90	you should be ashamed of yourself that is a ...	0

	<b>id</b>	<b>comment_text</b>	<b>severe_toxic</b>
<b>159568</b>	ffee36eab5c267c9	spitzer umm theres no actual article for pr...	0
<b>159569</b>	fff125370e4aaaf3	and it looks like it was actually you who put ...	0
<b>159570</b>	fff46fc426af1f9a	and i really don t think you understand ...	0

159571 rows × 3 columns

In [28]:

Threatening\_comment\_df

Out[28]:

	<b>id</b>	<b>comment_text</b>	<b>threat</b>
<b>0</b>	0000997932d777bf	explanation why the edits made under my userna...	0
<b>1</b>	000103f0d9cfb60f	d aww he matches this background colour i m s...	0
<b>2</b>	000113f07ec002fd	hey man i m really not trying to edit war it...	0
<b>3</b>	0001b41b1c6bb37e	more i can t make any real suggestions on im...	0
<b>4</b>	0001d958c54c6e35	you sir are my hero any chance you remember...	0
...	...	...	...
<b>159566</b>	ffe987279560d7ff	and for the second time of asking when ...	0
<b>159567</b>	ffea4adeee384e90	you should be ashamed of yourself that is a ...	0
<b>159568</b>	ffee36eab5c267c9	spitzer umm theres no actual article for pr...	0
<b>159569</b>	fff125370e4aaaf3	and it looks like it was actually you who put ...	0
<b>159570</b>	fff46fc426af1f9a	and i really don t think you understand ...	0

159571 rows × 3 columns

In [29]:

Obscene\_comment\_df

Out[29]:

	<b>id</b>	<b>comment_text</b>	<b>obscene</b>
<b>0</b>	0000997932d777bf	explanation why the edits made under my userna...	0
<b>1</b>	000103f0d9cfb60f	d aww he matches this background colour i m s...	0
<b>2</b>	000113f07ec002fd	hey man i m really not trying to edit war it...	0
<b>3</b>	0001b41b1c6bb37e	more i can t make any real suggestions on im...	0
<b>4</b>	0001d958c54c6e35	you sir are my hero any chance you remember...	0
...	...	...	...
<b>159566</b>	ffe987279560d7ff	and for the second time of asking when ...	0
<b>159567</b>	ffea4adeee384e90	you should be ashamed of yourself that is a ...	0
<b>159568</b>	ffee36eab5c267c9	spitzer umm theres no actual article for pr...	0
<b>159569</b>	fff125370e4aaaf3	and it looks like it was actually you who put ...	0
<b>159570</b>	fff46fc426af1f9a	and i really don t think you understand ...	0

159571 rows × 3 columns

In [30]: `Toxic_comment_df`

Out[30]:

	<b>id</b>	<b>comment_text</b>	<b>toxic</b>
<b>0</b>	0000997932d777bf	explanation why the edits made under my userna...	0
<b>1</b>	000103f0d9cfb60f	d aww he matches this background colour i m s...	0
<b>2</b>	000113f07ec002fd	hey man i m really not trying to edit war it...	0
<b>3</b>	0001b41b1c6bb37e	more i can t make any real suggestions on im...	0
<b>4</b>	0001d958c54c6e35	you sir are my hero any chance you remember...	0
...	...	...	...
<b>159566</b>	ffe987279560d7ff	and for the second time of asking when ...	0
<b>159567</b>	ffea4adeee384e90	you should be ashamed of yourself that is a ...	0
<b>159568</b>	ffee36eab5c267c9	spitzer umm theres no actual article for pr...	0
<b>159569</b>	fff125370e4aaaf3	and it looks like it was actually you who put ...	0
<b>159570</b>	fff46fc426af1f9a	and i really don t think you understand ...	0

159571 rows × 3 columns

In [31]: `IdentityHate_comment_df`

Out[31]:

	<b>id</b>	<b>comment_text</b>	<b>identity_hate</b>
<b>0</b>	0000997932d777bf	explanation why the edits made under my userna...	0
<b>1</b>	000103f0d9cfb60f	d aww he matches this background colour i m s...	0
<b>2</b>	000113f07ec002fd	hey man i m really not trying to edit war it...	0
<b>3</b>	0001b41b1c6bb37e	more i can t make any real suggestions on im...	0
<b>4</b>	0001d958c54c6e35	you sir are my hero any chance you remember...	0
...	...	...	...
<b>159566</b>	ffe987279560d7ff	and for the second time of asking when ...	0
<b>159567</b>	ffea4adeee384e90	you should be ashamed of yourself that is a ...	0
<b>159568</b>	ffee36eab5c267c9	spitzer umm theres no actual article for pr...	0
<b>159569</b>	fff125370e4aaaf3	and it looks like it was actually you who put ...	0
<b>159570</b>	fff46fc426af1f9a	and i really don t think you understand ...	0

159571 rows × 3 columns

In [32]: `Insulting_comment_df`

Out[32]:

	<b>id</b>	<b>comment_text</b>	<b>insult</b>
<b>0</b>	0000997932d777bf	explanation why the edits made under my userna...	0
<b>1</b>	000103f0d9cfb60f	d aww he matches this background colour i m s...	0

	<b>id</b>	<b>comment_text</b>	<b>insult</b>
<b>2</b>	000113f07ec002fd	hey man i m really not trying to edit war it...	0
<b>3</b>	0001b41b1c6bb37e	more i can t make any real suggestions on im...	0
<b>4</b>	0001d958c54c6e35	you sir are my hero any chance you remember...	0
...	...	...	...
<b>159566</b>	ffe987279560d7ff	and for the second time of asking when ...	0
<b>159567</b>	ffa4adeee384e90	you should be ashamed of yourself that is a ...	0
<b>159568</b>	ffee36eab5c267c9	spitzer umm theres no actual article for pr...	0
<b>159569</b>	fff125370e4aaaf3	and it looks like it was actually you who put ...	0
<b>159570</b>	fff46fc426af1f9a	and i really don t think you understand ...	0

159571 rows × 3 columns

In [33]:

```
import wordcloud
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from nltk.corpus import stopwords
```

In [34]:

```
def wordcloud(df, label):

    subset=df[df[label]==1]
    text=subset.comment_text.values
    wc= WordCloud(background_color="black",max_words=2000)

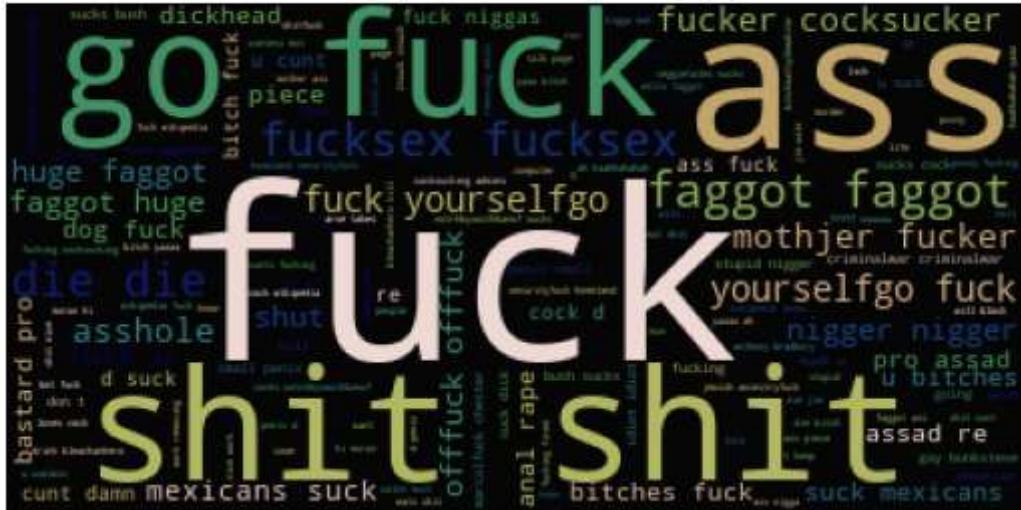
    wc.generate(" ".join(text))

    plt.figure(figsize=(20,20))
    plt.subplot(221)
    plt.axis("off")
    plt.title("Words frequented in {}".format(label), fontsize=20)
    plt.imshow(wc.recolor(colormap= 'gist_earth' , random_state=244), alpha=0.98)
    # Visualising the subset datasets using wordcloud
```

In [35]:

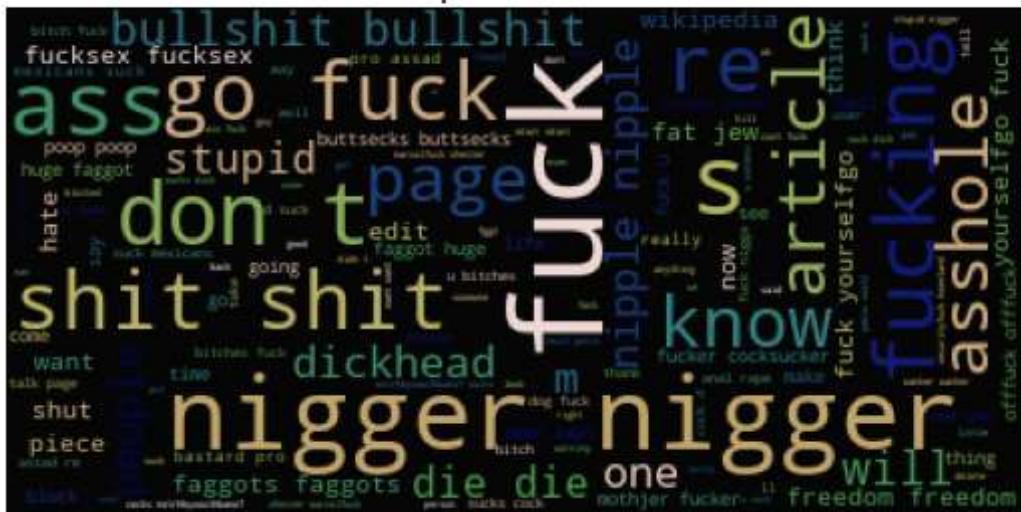
```
wordcloud(Severetoxic_comment_df,'severe_toxic')
```

#### Words frequented in severe toxic



```
In [36]: wordcloud(Obscene_comment_df, 'obscene')
```

#### Words frequented in obscene



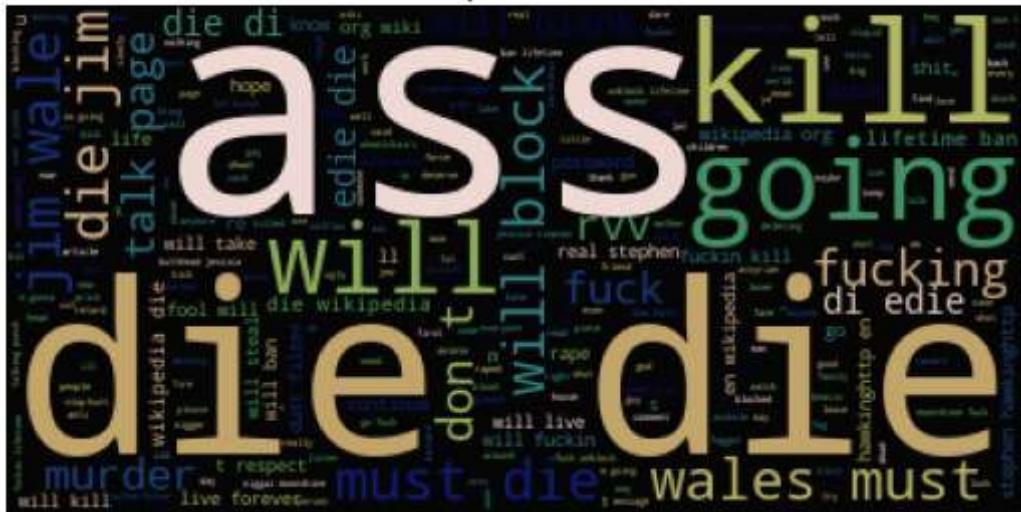
```
In [37]: wordcloud(Toxic comment df, 'toxic')
```

### Words frequented in toxic



```
In [38]: wordcloud(Threatening comment df, 'threat')
```

Words frequented in threat



```
In [39]: wordcloud(Insulting_comment_df, 'insult')
```

Words frequented in insult



```
In [40]: wordcloud(IdentityHate_comment_df, 'identity_hate')
```

Words frequented in identity\_hate



## Balancing the target column in the dataset

```
In [41]: Toxic_comment_balanced_1 = Toxic_comment_df[Toxic_comment_df['toxic'] == 1].iloc[0:5
# Selecting only 5000 toxic comments
```

```
In [42]: Toxic_comment_balanced_0 = Toxic_comment_df[Toxic_comment_df['toxic'] == 0].iloc[0:5
# Selecting only 5000 non toxic comments
```

```
In [43]: Toxic_comment_balanced_1.shape
# Shape of Toxic_comment_balanced_1
```

```
Out[43]: (5000, 3)
```

```
In [44]: Toxic_comment_balanced_0.shape
# Shape of Toxic_comment_balanced_0
```

```
Out[44]: (5000, 3)
```

```
In [45]: Toxic_comment_balanced_1['toxic'].value_counts()
# Value_counts of Toxic_comment_balanced_1
```

```
Out[45]: 1    5000
Name: toxic, dtype: int64
```

```
In [46]: Toxic_comment_balanced_0['toxic'].value_counts()
# Value_counts of Toxic_comment_balanced_0
```

```
Out[46]: 0    5000
Name: toxic, dtype: int64
```

```
In [47]: Toxic_comment_balanced=pd.concat([Toxic_comment_balanced_1,Toxic_comment_balanced_0]
## concatenating toxic and non toxic comments
```

```
In [48]: Toxic_comment_balanced['toxic'].value_counts()
# Balanced column
```

```
Out[48]: 0    5000
1    5000
Name: toxic, dtype: int64
```

## Repeating the steps for other subset datasets

```
In [49]: Severetoxic_comment_df['severe_toxic'].value_counts()
# value counts of Severetoxic_comment_df
```

```
Out[49]: 0    157976
1     1595
Name: severe_toxic, dtype: int64
```

```
In [50]: Severetoxic_comment_df_1 = Severetoxic_comment_df[Severetoxic_comment_df['severe_tox
# selecting 1595 values of Severetoxic_comment_df_1
```

```
In [51]: Severetoxic_comment_df_0 = Severetoxic_comment_df[Severetoxic_comment_df['severe_tox']
# selecting 1595 values of Severetoxic_comment_df_0
```

```
In [52]: Severe_toxic_comment_balanced=pd.concat([Severetoxic_comment_df_1,Severetoxic_comment_df_0])
# Concatenating Severetoxic_comment_df_1 and Severetoxic_comment_df_0
```

```
In [53]: Severe_toxic_comment_balanced['severe_toxic'].value_counts()
# Final value counts of the Severetoxic_comment_balanced
```

```
Out[53]: 0    1595
1    1595
Name: severe_toxic, dtype: int64
```

## Repeating the same for obscene comment data frame

```
In [54]: Obscene_comment_df['obscene'].value_counts()
# Value counts of the obscene_comment_df
```

```
Out[54]: 0    151122
1     8449
Name: obscene, dtype: int64
```

```
In [55]: Obscene_comment_df_1 = Obscene_comment_df[Obscene_comment_df['obscene'] == 1].iloc[0:5000]
```

```
In [56]: Obscene_comment_df_0 = Obscene_comment_df[Obscene_comment_df['obscene'] == 0].iloc[0:5000]
```

```
In [57]: Obscene_comment_balanced = pd.concat([Obscene_comment_df_1,Obscene_comment_df_0])
```

```
In [58]: Obscene_comment_balanced['obscene'].value_counts()
```

```
Out[58]: 0    5000
1    5000
Name: obscene, dtype: int64
```

```
In [59]: ### Repeating the same for Threatening comment data frame
```

```
In [60]: Threatening_comment_df
```

```
Out[60]:
```

	<b>id</b>	<b>comment_text</b>	<b>threat</b>
<b>0</b>	0000997932d777bf	explanation why the edits made under my userna...	0
<b>1</b>	000103f0d9cfb60f	d aww he matches this background colour i m s...	0
<b>2</b>	000113f07ec002fd	hey man i m really not trying to edit war it...	0
<b>3</b>	0001b41b1c6bb37e	more i can t make any real suggestions on im...	0
<b>4</b>	0001d958c54c6e35	you sir are my hero any chance you remember...	0
...	...	...	...
<b>159566</b>	ffe987279560d7ff	and for the second time of asking when ...	0

	<b>id</b>	<b>comment_text</b>	<b>threat</b>
159567	ffea4adeee384e90	you should be ashamed of yourself that is a ...	0
159568	ffee36eab5c267c9	spitzer umm theres no actual article for pr...	0
159569	fff125370e4aaaf3	and it looks like it was actually you who put ...	0
159570	fff46fc426af1f9a	and i really don t think you understand ...	0

159571 rows × 3 columns

```
In [61]: Threatening_comment_df['threat'].value_counts()
```

```
Out[61]: 0    159093
          1     478
          Name: threat, dtype: int64
```

```
In [62]: Threatening_comment_df_1 = Threatening_comment_df[Threatening_comment_df['threat'] =
```

```
In [63]: Threatening_comment_df_0 = Threatening_comment_df[Threatening_comment_df['threat'] =
```

```
In [64]: Threatening_comment_balanced = pd.concat([Threatening_comment_df_1,Threatening_comme
```

```
In [65]: Threatening_comment_balanced['threat'].value_counts()
```

```
Out[65]: 0    478
          1     478
          Name: threat, dtype: int64
```

```
In [66]: Threatening_comment_balanced
```

	<b>id</b>	<b>comment_text</b>	<b>threat</b>
79	003217c3eb469ba9	hi i am back again last warning stop undoin...	1
176	006b94add72ed61c	i think that your a fagget get a oife and burn...	1
600	0199d6af27b715f3	i m also a sock puppet of this account supri...	1
802	02230885017a50c5	fuck you smith please have me notified when ...	1
1017	02c6e41e4b317ac3	wouldn t be the first time bitch fuck you i l...	1
...	...	...	...
475	01389dc7e054bfe5	hello if you re interested we could re...	0
476	013ad5246f4b953b	f k all rides at worlds of fun i hate all ...	0
477	013bd808a6d3d69b	as to job for you sure thing as soon as i ...	0
478	013c1a43411c5f9a	oh ok i just wanted to see what people though...	0
479	013d17a8b342f501	wrong the capital is podgorica as it has alwa...	0

956 rows × 3 columns

In [67]: *### Repeating the same for Insulting\_comment\_data frame*

In [68]: `Insulting_comment_df['insult'].value_counts()`

Out[68]:

0	151694
1	7877
Name: insult, dtype: int64	

In [69]: `Insulting_comment_df_1 = Insulting_comment_df[Insulting_comment_df['insult'] == 1].i`

In [70]: `Insulting_comment_df_0 = Insulting_comment_df[Insulting_comment_df['insult'] == 0].i`

In [71]: `Insulting_comment_balanced = pd.concat([Insulting_comment_df_1, Insulting_comment_df_`

In [72]: `Insulting_comment_balanced['insult'].value_counts()`

Out[72]:

0	5000
1	5000
Name: insult, dtype: int64	

In [73]: *### Repeating the same for IdentityHate\_comment\_df*

In [74]: `IdentityHate_comment_df['identity_hate'].value_counts()`

Out[74]:

0	158166
1	1405
Name: identity_hate, dtype: int64	

In [75]: `IdentityHate_comment_df_1 = IdentityHate_comment_df[IdentityHate_comment_df['identit`

In [76]: `IdentityHate_comment_df_0 = IdentityHate_comment_df[IdentityHate_comment_df['identit`

In [77]: `IdentityHate_comment_balanced = pd.concat([IdentityHate_comment_df_1, IdentityHate_co`

In [78]: `IdentityHate_comment_balanced['identity_hate'].value_counts()`

Out[78]:

0	1405
1	1405
Name: identity_hate, dtype: int64	

## Machine learning

In [79]:

```
from sklearn import preprocessing
from sklearn.feature_selection import SelectFromModel
```

```
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.metrics import f1_score, precision_score, recall_score, precision_recall_curve
from sklearn.metrics import roc_auc_score, roc_curve
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB, BernoulliNB
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier

from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from nltk import ngrams, bigrams, trigrams
```

In [81]:

```
def cv_tf_train_test(dataframe,label,vectorizer,ngram):

    # Split the data into X and y data sets
    X = dataframe.comment_text
    y = dataframe[label]

    # Split our data into training and test data
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_

    # Using vectorizer and removing stopwords
    cv1 = vectorizer(ngram_range=(ngram), stop_words='english')

    # Transforming x-train and x-test
    X_train_cv1 = cv1.fit_transform(X_train)
    X_test_cv1 = cv1.transform(X_test)

    ## Machine Learning models

    ## Logistic regression
    lr = LogisticRegression()
    lr.fit(X_train_cv1, y_train)

    ## k-nearest neighbours
    knn = KNeighborsClassifier(n_neighbors=5)
    knn.fit(X_train_cv1, y_train)

    ## Naive Bayes
    bnb = BernoulliNB()
    bnb.fit(X_train_cv1, y_train)

    ## Multinomial naive bayes
    mnb = MultinomialNB()
    mnb.fit(X_train_cv1, y_train)

    ## Support vector machine
    svm_model = LinearSVC()
    svm_model.fit(X_train_cv1, y_train)

    ## Random Forest
    randomforest = RandomForestClassifier(n_estimators=100, random_state=50)
    randomforest.fit(X_train_cv1, y_train)

    f1_score_data = {'F1 Score':[f1_score(lr.predict(X_test_cv1), y_test), f1_score(
        bnb.predict(X_test_cv1), y_test), f1_score(
        mnb.predict(X_test_cv1), y_test), f1_
    ## Saving f1 score results into a dataframe
    df_f1 = pd.DataFrame(f1_score_data, index=[ 'Log Regression', 'KNN', 'BernoulliNB'])

    return df_f1
```

## Evaluating model performance using evaluation metrics.

```
In [82]: severe_toxic_comment_cv = cv_tf_train_test(Severe_toxic_comment_balanced, 'severe_toxic', TfidfVect
severe_toxic_comment_cv.rename(columns={'F1 Score': 'F1 Score(severe_toxic)'}, inplace=True)
severe_toxic_comment_cv
# Multinomial NB has higher F1 score
```

Out[82]: **F1 Score(severe\_toxic)**

<b>Log Regression</b>	0.940282
<b>KNN</b>	0.860192
<b>BernoulliNB</b>	0.790738
<b>MultinomialNB</b>	0.932377
<b>SVM</b>	0.937901
<b>Random Forest</b>	0.941176

```
In [83]: obscene_comment_cv = cv_tf_train_test(Obscene_comment_balanced, 'obscene', TfidfVect
obscene_comment_cv.rename(columns={'F1 Score': 'F1 Score(obscene)'}, inplace=True)
obscene_comment_cv
# Random Forest has higher F1 score
```

Out[83]: **F1 Score(obscene)**

<b>Log Regression</b>	0.901183
<b>KNN</b>	0.625341
<b>BernoulliNB</b>	0.766640
<b>MultinomialNB</b>	0.887496
<b>SVM</b>	0.915613
<b>Random Forest</b>	0.884261

```
In [85]: threatening_comment_cv = cv_tf_train_test(Threatening_comment_balanced, 'threat', TfidfVect
threatening_comment_cv.rename(columns={'F1 Score': 'F1 Score(threat)'}, inplace=True)
threatening_comment_cv
# Random Forest has higher F1 score
```

Out[85]: **F1 Score(threat)**

<b>Log Regression</b>	0.897338
<b>KNN</b>	0.852459
<b>BernoulliNB</b>	0.745205
<b>MultinomialNB</b>	0.902098
<b>SVM</b>	0.894737
<b>Random Forest</b>	0.923077

```
In [86]: insulting_comment_cv = cv_tf_train_test(Insulting_comment_balanced, 'insult', TfidfVect
insulting_comment_cv.rename(columns={'F1 Score': 'F1 Score(insult)'}, inplace=True)
insulting_comment_cv
# SVM has higher F1 score
```

Out[86]:

	F1 Score(insult)
<b>Log Regression</b>	0.901851
<b>KNN</b>	0.320661
<b>BernoulliNB</b>	0.776986
<b>MultinomialNB</b>	0.896299
<b>SVM</b>	0.906218
<b>Random Forest</b>	0.890821

In [87]:

```
identity_hatecomment_cv = cv_tf_train_test(IdentityHate_comment_balanced, 'identity_hatecomment_cv')
identity_hatecomment_cv.rename(columns={'F1 Score': 'F1 Score(identity_hate)'}, inplace=True)
# MultinomialNB has higher F1 score
```

Out[87]:

	F1 Score(identity_hate)
<b>Log Regression</b>	0.905707
<b>KNN</b>	0.820046
<b>BernoulliNB</b>	0.776699
<b>MultinomialNB</b>	0.903302
<b>SVM</b>	0.896806
<b>Random Forest</b>	0.888087

In [88]:

```
X = Toxic_comment_balanced.comment_text
y = Toxic_comment_balanced['toxic']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initiate a TfIdf vectorizer
tfv = TfidfVectorizer(ngram_range=(1,1), stop_words='english')

X_train_fit = tfv.fit_transform(X_train)
X_test_fit = tfv.transform(X_test)
randomforest = RandomForestClassifier(n_estimators=100, random_state=50)

randomforest.fit(X_train_fit, y_train)
randomforest.predict(X_test_fit)
```

Out[88]: array([0, 1, 1, ..., 1, 1, 1], dtype=int64)

In [89]:

```
## Testing the model to check if the given text is toxic or not.
```

In [90]:

```
comment1 = ['i killed an insect and ate it']
comment1_vect = tfv.transform(comment1)
randomforest.predict_proba(comment1_vect)[:,1]
## As seen below the above comment is 73 percent toxic
```

Out[90]: array([0.73519444])

```
In [91]: comment2 = ['Is this sentence a good one']
comment2_vect = tfv.transform(comment2)
randomforest.predict_proba(comment2_vect)[:,1]
## As seen below the above comment is 0.08 percent toxic which says the comment is n
```

```
Out[91]: array([0.08770635])
```

```
In [92]: comment2 = ['truth will prevail']
comment2_vect = tfv.transform(comment2)
randomforest.predict_proba(comment2_vect)[:,1]
## The above comment is 46 percent toxic.
```

```
Out[92]: array([0.46238997])
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```