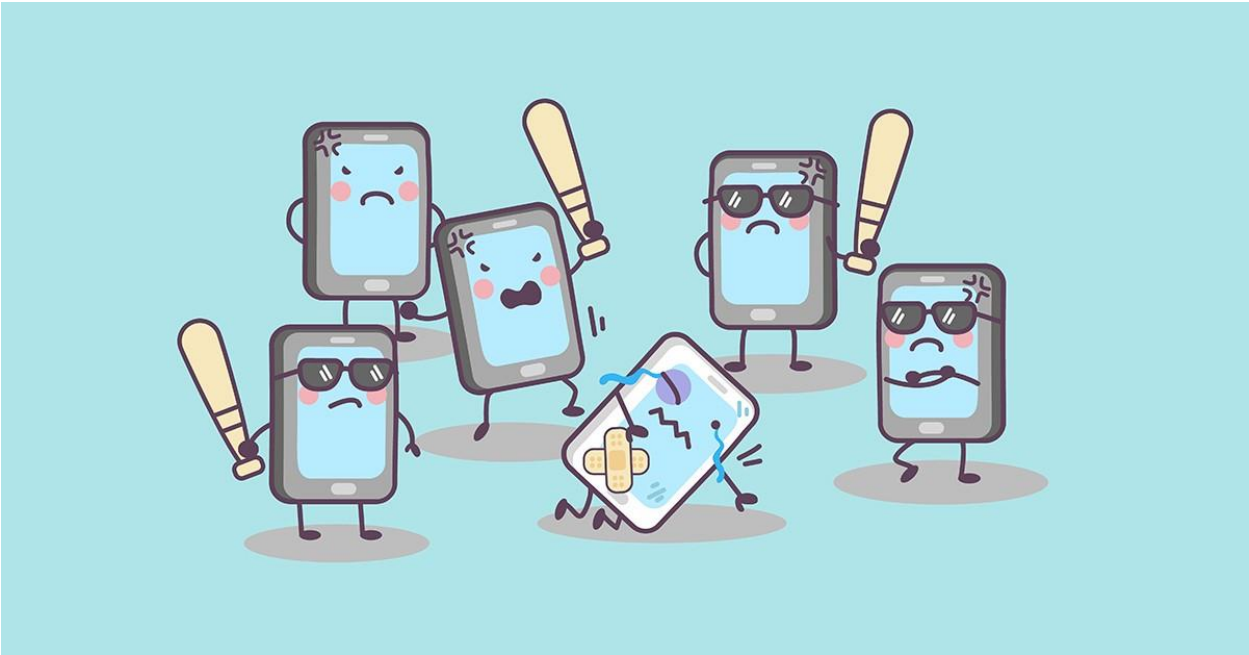
FLIP ROBO

MALIGNANT COMMENTS CLASSIFIER



SUBMITTED BY

RAJASEKAR S

ACKNOWLEDGEMENT

This project includes the professional reference of much external reference analysis done by the individuals. Such references are mentioned below,

1. Raushan Kumar
2. ArunKumar

These individuals professionals and their researches helped and guided me in completion of this project.

TABLE OF CONTENTS

1. PROBLEM STATEMENT
2. BUSINESS GOAL
3. DATA-SET DESCRIPTION
4. DATA PREPROCESSING
5. DATA CLEANING
6. DATA VISUALIZATION
7. FEATURE EXTRACTION
8. MODEL BUILDING

PROBLEM STATEMENT

- The proliferation of social media enables people to express their opinions widely online. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users. Although researchers have found that hate is a problem across multiple platforms, there is a lack of models for online hate detection.
- Online hate, described as abusive language, aggression, cyberbullying, hatefulness and many others has been identified as a major threat on online social media platforms. Social media platforms are the most prominent grounds for such toxic behaviour.
- There has been a remarkable increase in the cases of cyberbullying and trolls on various social media platforms. Many celebrities and influences are facing backlashes from people and have to come across hateful and offensive comments. This can take a toll on anyone and affect them mentally leading to depression, mental illness, self-hatred and suicidal thoughts.
- Internet comments are bastions of hatred and vitriol. While online anonymity has provided a new outlet for aggression and hate speech, machine learning can be used to fight it. The problem we sought to solve was the tagging of internet comments that are aggressive towards other users. This means that insults to third parties such as celebrities will be tagged as unoffensive, but “u are an idiot” is clearly offensive.

BUSINESS GOAL

Our goal is to build a prototype of online hate and abuse comment classifier which can used to classify hate and offensive comments so that it can be controlled and restricted from spreading hatred and cyberbullying.

DATA-SET DESCRIPTIONS

The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains nearly 1,53,000 samples. All the data samples contain 8 fields which includes 'Id', 'Comments', 'Malignant', 'Highly malignant', 'Rude', 'Threat', 'Abuse' and 'Loathe'.

The data-set contains:-

- **Malignant:** It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.
- **Highly Malignant:** It denotes comments that are highly malignant and hurtful.
- **Rude:** It denotes comments that are very rude and offensive.
- **Threat:** It contains indication of the comments that are giving any threat to someone.
- **Abuse:** It is for comments that are abusive in nature.
- **Loathe:** It describes the comments which are hateful and loathing in nature.
- **ID:** It includes unique Ids associated with each comment text given.
- **Comment text:** This column contains the comments extracted from various social media platforms.

ANALYTICAL PROBLEM FRAMING

1. What is Analytical problem framing :-

Analytic problem framing involves translating the business problem into terms that can be addressed analytically via data and modelling. It's at this stage that you work backwards From the results / outputs you want to the data/inputs you're going to need, where you identify potential drivers and

hypotheses to test, and where you nail down your assumptions. Analytic problem framing is the antithesis of merely working with the ready-to-hand data and seeing what comes of it, hoping for something insightful. Typically, the process moves on from here to data collection, cleansing and transformation, Methodology selection and model building, never to return. But if you're willing to borrow and use a concept from complex adaptive systems – maps and models – you can make repeat use of this stage to improve your overall outcome.

2. Hardware requirements:-

A mid level computer that runs on Intel i3/i5/i7 or A10/A11/M1 or ryzen 3/5 or any other equivalent chipset and a suitable processor.

3. Software requirements :-

Windows / Linux /Mac OS

4. TOOLS, Libraries and packages used :-

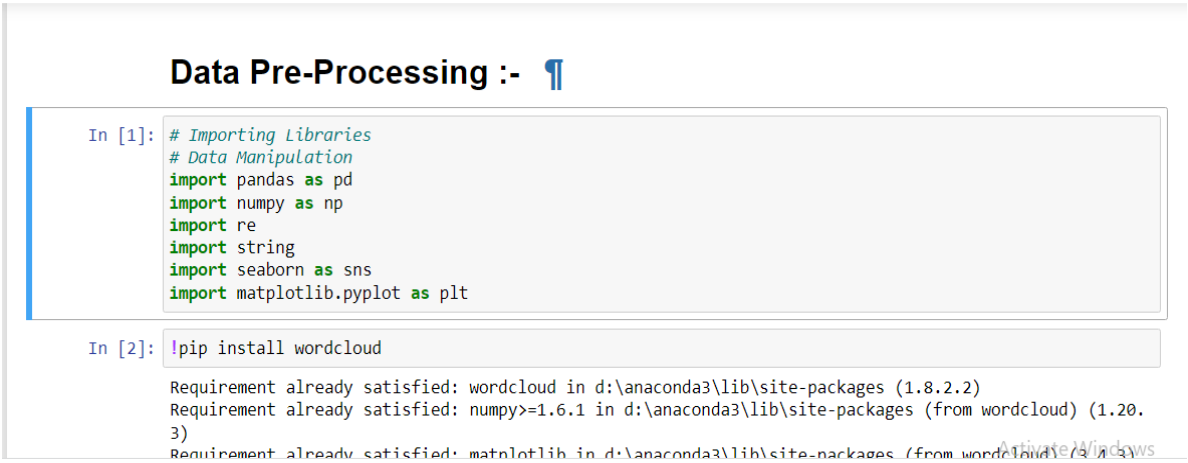
Tool: 1.Anaconda Navigator

2. JupyterNotebook

Libraries and Packages:

- **Numpy**
- **Pandas**
- **Matplotlib**
- **Seaborn**

5. DATA-PREPROCESSING :-



Data Pre-Processing :- ¶

In [1]:
```
# Importing Libraries
# Data Manipulation
import pandas as pd
import numpy as np
import re
import string
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in d:\anaconda3\lib\site-packages (1.8.2.2)
Requirement already satisfied: numpy>=1.6.1 in d:\anaconda3\lib\site-packages (from wordcloud) (1.20.
3)
Requirement already satisfied: matplotlib in d:\anaconda3\lib\site-packages (from wordcloud) (3.4.3)
```

In [3]:
```
# Machine Learning and Text Processing Libraries
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from wordcloud import WordCloud
```

In [4]:
```
!pip install neattext
```

```
Requirement already satisfied: neattext in d:\anaconda3\lib\site-packages (0.1.3)
```

In [5]:
```
import neattext as nt
import neattext.functions as nfx
```

In [6]:
```
# Libraries used for csv file
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
dftrain=pd.read_csv(r'D:\Malignant Comments Classifier Project\malignant_comments_classifier_train.csv'
dftrain.head(10)
```

Out[6]:

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

In [7]:
```
dftrain.info()
```

```
<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   malignant         159571 non-null  int64
 3   highly_malignant  159571 non-null  int64
 4   rude              159571 non-null  int64
 5   threat            159571 non-null  int64
 6   abuse             159571 non-null  int64
 7   loathe            159571 non-null  int64
dtypes: int64(6), object(2)
memory usage: 9.7+ MB
```

In [8]:
```
dftrain.shape
```

In [8]:
```
dftrain.shape
```

Out[8]:
```
(159571, 8)
```

In [9]:
```
# Creating the new column for comments_length
dftrain['comments_length']=dftrain['comment_text'].str.len()
dftrain.head(10)
```

Out[9]:

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	comments_length
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	264
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	112
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	233
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0	622

In [10]:
```
dftrain['comment_text'].apply(lambda x:nt.TextFrame(x).noise_scan()) #noise present in the comment text
```

Out[10]:
```
0         {'text_noise': 11.742424242424242, 'text_lengt...
1         {'text_noise': 12.5, 'text_length': 112, 'nois...
2         {'text_noise': 12.017167381974248, 'text_lengt...
3         {'text_noise': 13.183279742765272, 'text_lengt...
4         {'text_noise': 14.925373134328357, 'text_lengt...
5         {'text_noise': 12.307692307692308, 'text_lengt...
6         {'text_noise': 11.363636363636363, 'text_lengt...
7         {'text_noise': 14.782608695652174, 'text_lengt...
8         {'text_noise': 14.40677966101695, 'text_length...
9         {'text_noise': 11.428571428571429, 'text_lengt...
10        {'text_noise': 11.68695652173913, 'text_length...
11        {'text_noise': 12.5, 'text_length': 56, 'noise...
12        {'text_noise': 14.733542319749215, 'text_lengt...
13        {'text_noise': 11.477411477411477, 'text_lengt...
14        {'text_noise': 13.698630136986301, 'text_lengt
```


In [11]:
```
print(dftrain.isnull().sum())
```

```
id                0
comment_text      0
malignant         0
highly_malignant  0
rude              0
threat            0
abuse             0
loathe            0
comments_length   0
dtype: int64
```

In [12]:
```
dftrain['comment_text_noise']=dftrain['comment_text'].apply(lambda x:nt.TextExtractor(x).extract_stopwo
dftrain['comment_text_noise'] # extracted the stopwords from the comment text
```

Out[12]:
```
0         [why, the, made, under, my, were, they, weren'
```

```
Out[12]: 0         [why, the, made, under, my, were, they, weren'...
         1                                              [he, this]
         2         [really, not, to, it's, just, that, this, is, ...
         3         [more, i, make, any, on, i, if, the, should, b...
         4                                 [are, my, any, you, what]
         5                                      [from, me, as, the]
         6                               [before, you, around, on, my]
         7         [your, to, the, has, been, please, don't, do, ...
         8         [if, the, was, to, not, to, anything, in, the,...
         9               [on, this, and, which, are, to, those, of]
         10        [for, for, i, the, that, the, is, being, used,...
         11                                   [be, a, and, over, the]
         12        [what, is, what, is, an, of, some, are, at, wh...
         13        [before, you, and, at, the, isn't, to, your, i...
         14        [and, the, above, her, with, she, her, where, ...
```


DATA CLEANING

```
In [13]: dftrain['comment_textnew']=dftrain['comment_text'].apply(nfx.remove_stopwords) #removing stopwords
```

```
In [14]: # Removing punctuations Special characters and URL'S
         dftrain['comment_textnew']=dftrain['comment_textnew'].apply(nfx.remove_punctuations)
         dftrain['comment_textnew']=dftrain['comment_textnew'].apply(nfx.remove_special_characters)
         dftrain['comment_textnew']=dftrain['comment_textnew'].apply(nfx.remove_phone_numbers)
         dftrain['comment_textnew']=dftrain['comment_textnew'].apply(nfx.remove_urls)
```

```
In [15]: dftrain['clean_length']=dftrain.comment_textnew.str.len()
         dftrain.head(10)
```

Out[15]:

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	comments_length	comment
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	264	[why under, my
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	112	
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	233	[really just, tha
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0	622	[more, i on, i, if,

OBSERVATIONS :-

We could see that data has been cleaned and is ready to build a ML model.But before that let’s sense the word traffic in the given data-set. This helps in understanding the dataset and also helps us in choosing.

- **DATA VISUALIZATION :-**

Data Visualization :-

We will now create a column integrating all the target.

In [16]:
```
target=['malignant','highly_malignant','loathe','rude','abuse','threat']

for i in target:
    print(i)
    print(dftrain[i].value_counts())
    print("------------------------------------")
```

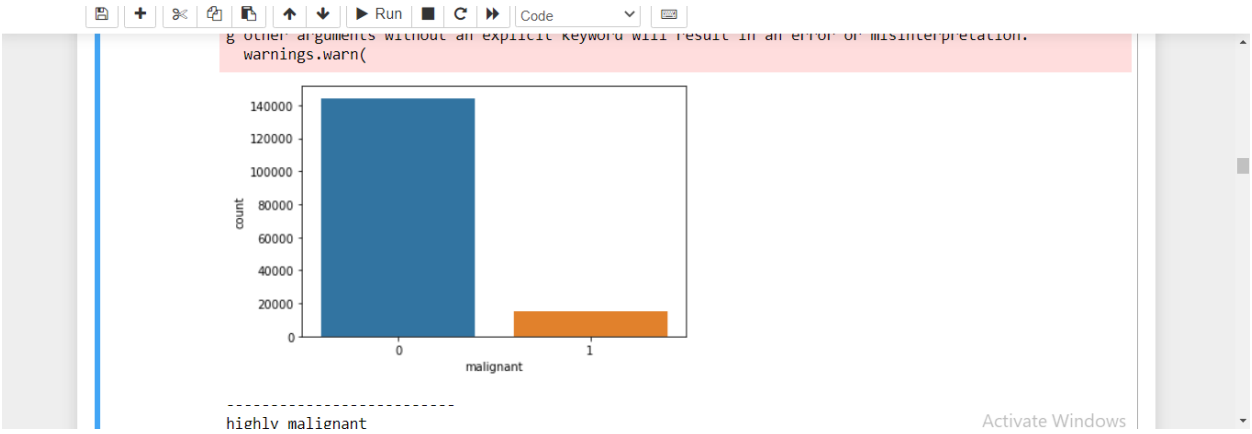
```
malignant
0    144277
1     15294
Name: malignant, dtype: int64
------------------------------------
highly_malignant
```

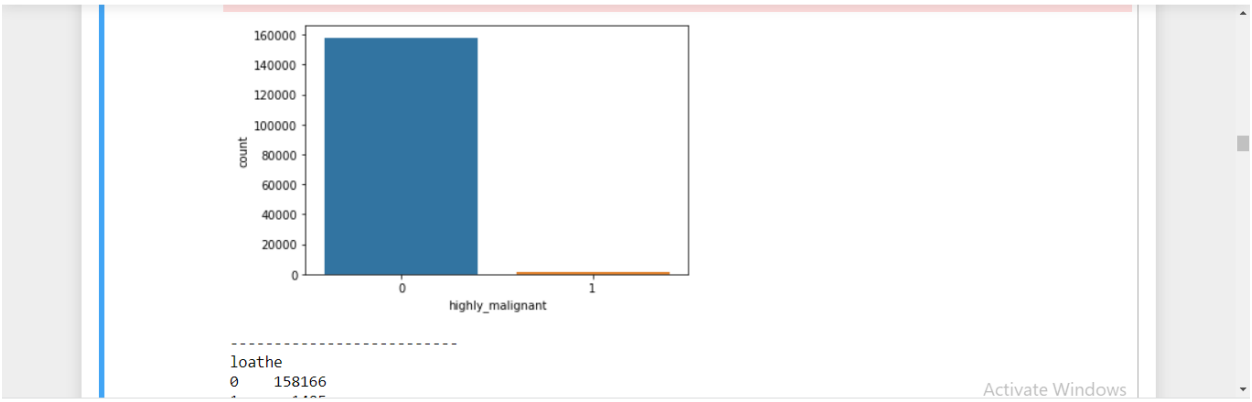


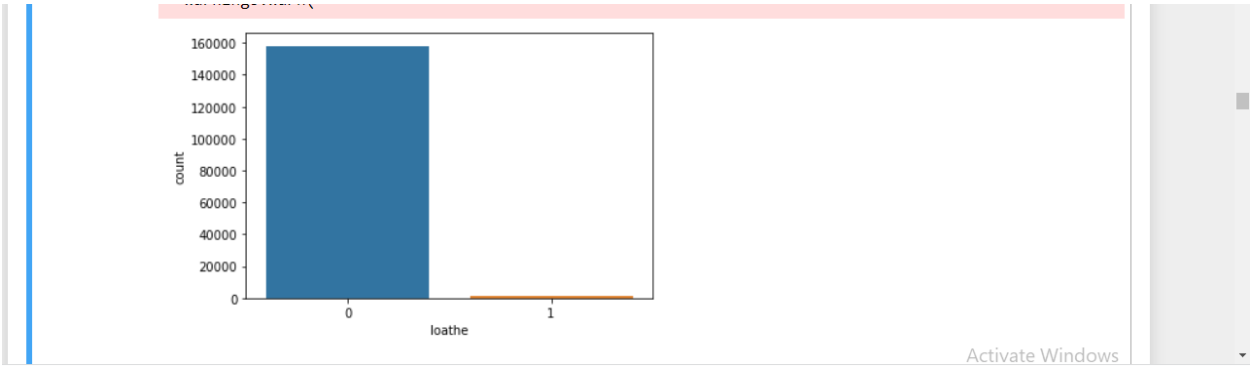
```
loathe
0    158166
1      1405
Name: loathe, dtype: int64
------------------------------------
rude
0    151122
1      8449
Name: rude, dtype: int64
------------------------------------
abuse
0    151694
1      7877
Name: abuse, dtype: int64
------------------------------------
threat
0    159093
1       478
Name: threat, dtype: int64
------------------------------------
```


In [17]:
```
target=['malignant','highly_malignant','loathe','rude','abuse','threat']

for i in target:
    print(i)
    print(dftrain[i].value_counts())
    sns.countplot(dftrain[i])
    plt.show()
    print("--------------------------")
```

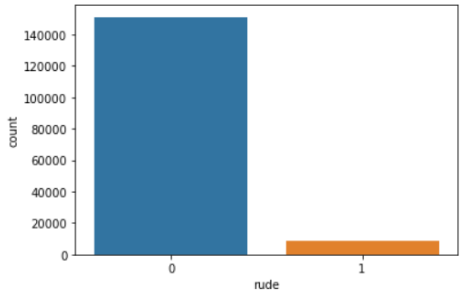
```
malignant
0    144277
1     15294
Name: malignant, dtype: int64

D:\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```


Run
Code
warnings.warn(
140000
120000
100000
80000
60000
40000
20000
0
count
0
1
malignant
----
highly_malignant

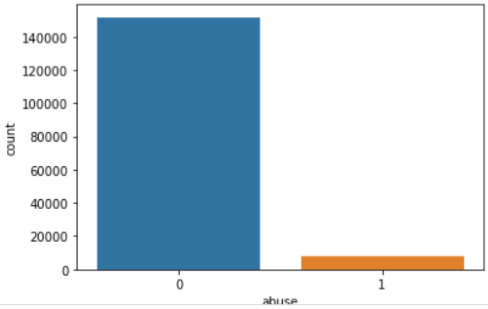

160000
140000
120000
100000
80000
60000
40000
20000
0
count
0
1
highly_malignant
----
loathe
0 158166

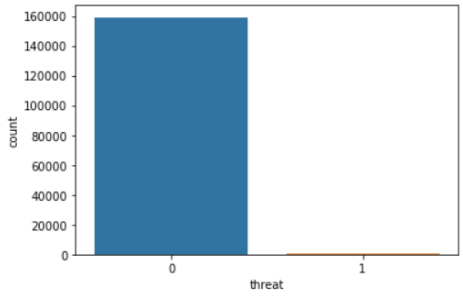

160000
140000
120000
100000
80000
60000
40000
20000
0
count
0
1
loathe

```
  warnings.warn(
```




```
---------------------------
abuse
```



In [18]:
```
# Getting sense of words in malignant
plot=dftrain['comment_textnew'][dftrain['malignant']==1]
print(len(plot))
plot_cloud=WordCloud(width=700,height=500,background_color='white',max_words=200).generate(' '.join(plo
plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(plot_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

15294

15294

Malignant image.jpeg

Show all

28°C ENG 20:22 16-09-2022

In [19]:
```
# Getting Sense of Words in Highly_Malignant
plot=dftrain['comment_textnew'][dftrain['rude']==1]
print(len(plot))
plot_cloud=WordCloud(width=700,height=500,background_color='white',max_words=200).generate(' '.join(plo
plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(plot_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

In [20]:
```
# Getting Sense of words in Highly-Malignant
plot=dftrain['comment_textnew'][dftrain['threat']==1]
print(len(plot))
plot_cloud=WordCloud(width=700,height=500,background_color='white',max_words=200).generate(' '.join(plo
plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(plot_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

In [21]:
```
# Getting sense of same words in Highly_Malignant
plot=dftrain['comment_textnew'][dftrain['abuse']==1]
print(len(plot))
plot_cloud=WordCloud(width=700,height=500,background_color='white',max_words=200).generate(' '.join(plo
plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(plot_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

gnant image.jpeg

In [22]:
```
#Getting sense of words in Highly_malignant
plot=dftrain['comment_textnew'][dftrain['loathe']==1]
print(len(plot))
plot_cloud=WordCloud(width=700,height=500,background_color='white',max_words=200).generate(' '.join(plo
plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(plot_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

1405

In [23]:
```
dftrain[target].sum()
```

Out[23]:
```
malignant          15294
highly_malignant    1595
loathe              1405
rude                8449
abuse               7877
threat               478
dtype: int64
```

In [24]:
```
dftrain.shape
```

Out[24]:
```
(159571, 12)
```

In [25]:
```
# Adding a column representing the comments with all the characteristics
dftrain['Target']=dftrain[target].sum(axis=1)
dftrain.head(5)
```


In [25]:
```
# Adding a column representing the comments with all the characteristics
dftrain['Target']=dftrain[target].sum(axis=1)
dftrain.head(5)
```

Out[25]:

ıt	highly_malignant	rude	threat	abuse	loathe	comments_length	comment_text_noise	comment_textnew	clean_length	Target
0	0	0	0	0	0	264	[why, the, made, under, my, were, they, weren'...	Explanation edits username Hardcore Metallica ...	156	0
0	0	0	0	0	0	112	[he, this]	Daww matches background colour Im seemingly st...	91	0
0	0	0	0	0	0	233	[really, not, to, it's, just, that, this, is, ...	Hey man Im trying edit war guy constantly remo...	131	0
							[more, i, make, any,	cant real		

Out[25]:

ıt	highly_malignant	rude	threat	abuse	loathe	comments_length	comment_text_noise	comment_textnew	clean_length	Target
0	0	0	0	0	0	264	[why, the, made, under, my, were, they, weren'...	Explanation edits username Hardcore Metallica ...	156	0
0	0	0	0	0	0	112	[he, this]	Daww matches background colour Im seemingly st...	91	0
0	0	0	0	0	0	233	[really, not, to, it's, just, that, this, is, ...	Hey man Im trying edit war guy constantly remo...	131	0
0	0	0	0	0	0	622	[more, i, make, any, on, i, if, the, should, b...	cant real suggestions improvement wondered s...	361	0
0	0	0	0	0	0	67	[are, my, any, you, what]	You sir hero chance remember	42	0

In [26]:
```
dftrain['Target'].unique()
```

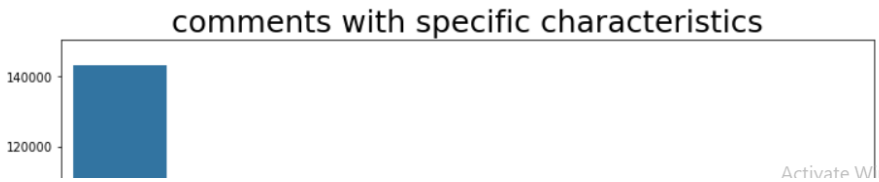
Out[26]:
```
array([0, 4, 1, 3, 2, 5, 6], dtype=int64)
```

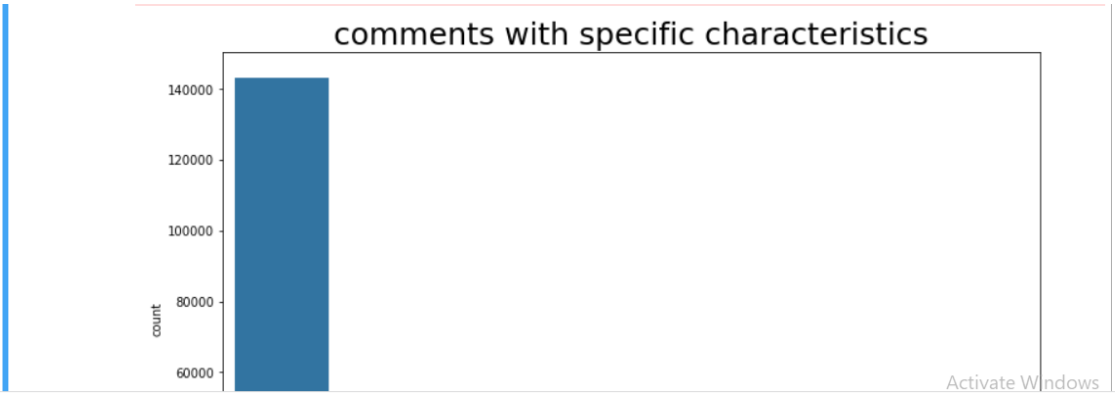
In [27]:
```
for i in range(0,7):
    print(f'For value:{i}')
    print(dftrain['Target']==(i)).sum()
    print("--------------------------------")
```

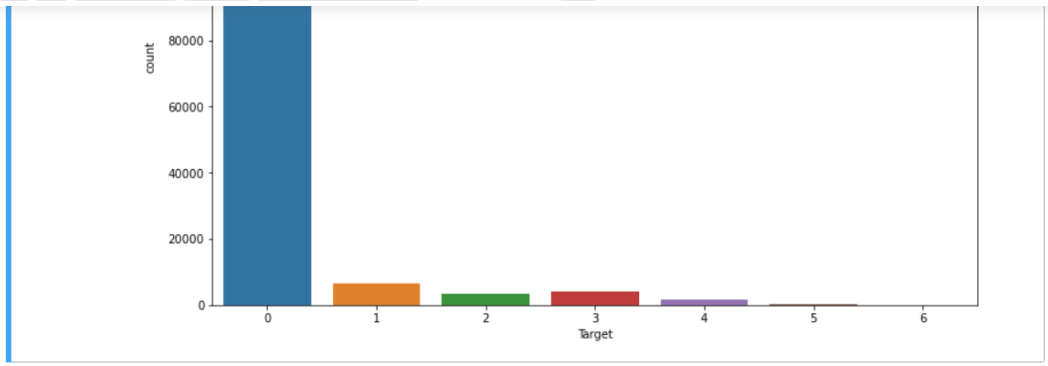
```
For value:0
0          True
1          True
2          True
3          True
4          True
5          True
```


In [28]:
```
# Comments with all the characteristics
plt.figure(figsize=(12,8))
sns.countplot(dftrain['Target'])
plt.title("comments with specific characteristics",fontsize=25)
plt.show()
```

```
D:\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```







In [29]:
```
dftrain['Target']=dftrain['Target']>0
dftrain['Target']=dftrain['Target'].astype(int)
print(dftrain['Target'].value_counts())
```

```
0    143346
1     16225
Name: Target, dtype: int64
```

In [30]:
```
dftrain.head(10)
```

Out[30]:

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	comments_length	comment
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	264	[why under, my
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	112	

Out[30]:

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	comments_length	comment
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	264	[why, under, my
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	112	
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	233	[really, just, tha
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0	622	[more, i, on, i, if,
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0	67	[are, r

In [31]:
```
dftrain=dftrain.drop(['id'],axis=1)
```

In [32]:
```
dftrain.head(10)
```

Out[32]:

	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	comments_length	comment_text_noise	comme
0	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	264	[why, the, made, under, my, were, they, weren'...	Expla Hardcc
1	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	112	[he, this]	Dav backgr Im se
2	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	233	[really, not, to, it's, just, that, this, is, ...	Hey m consta
3	"\nMore\nI can't make any real	0	0	0	0	0	0	622	[more, i, make, any, on, i, if, the, should,	ir

	the edits made under my usern...	0	0	0	0	0	0	264	under, my, were, they, weren'...	Hardcc
1	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	112	[he, this]	Dav backgr Im se
2	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	233	[really, not, to, it's, just, that, this, is, ...	Hey m consta
3	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0	622	[more, i, make, any, on, i, if, the, should, b...	ir w
4	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0	67	[are, my, any, you, what]	chanc pa
5	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0	65	[from, me, as, the]	Con well us
6	COCKSUCKER BEFORE YOU	1	1	1	0	1	0	44	[before, you, around,	COC

In [33]:
```
dftrain.Target.unique()
```

Out[33]:
```
array([0, 1])
```

- FEATURE EXTRACTION :-

Feature Extraction

In [34]:
```
tfidf=TfidfVectorizer(max_features=2000)
x=tfidf.fit_transform(dftrain['comment_textnew'])
y=dftrain[['malignant','highly_malignant','rude','threat','abuse','loathe']]
```

In [35]:
```
#Creating train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=42,test_size=0.30)
```

In [36]:
```
x.shape,y.shape
```

Out[36]: ((159571, 2000), (159571, 6))

In [37]:
```
x_train.shape , y_train.shape
```

Out[37]: ((111699, 2000), (111699, 6))

- MODEL BUILDING :-

MODEL BUILDING:-

In this dataset we have 6 target values, which means this is a Multi-classification problems.
So we will use Multi-Label classification libraries to build the ML model.

In [38]:
```
# Importing all the model library
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
```

In [39]:
```
# Importing Error Metrics
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score,roc_curve,auc,f1_scor
```

In [40]:
```
!pip install scikit-multilearn
```

```
Requirement already satisfied: scikit-multilearn in d:\anaconda3\lib\site-packages (0.2.0)
```

In [41]:
```
# Importing Multilabel classification Libraries
import skmultilearn
from skmultilearn.problem_transform import BinaryRelevance
from skmultilearn.problem_transform import ClassifierChain
```

In [42]:
```
#Declaring models
LR=LogisticRegression()
SVC=LinearSVC(random_state=0)
MNB=MultinomialNB()
```

In [43]:
```
#appending models
models=[]
models.append(LR)
models.append(MNB)
models.append(SVC)

#Appending estimators
estimators=[]
estimators.append(BinaryRelevance)
estimators.append(ClassifierChain)
```

In [44]:
```
#Creating emptylist
Model=[]
Estimator=[]
F1_score=[]
#Creating a loop to run the data through the models
for model in models:

    for estimator in estimators:
        #model fitting
        clf=estimator(model)
        Model.append(model)
        Estimator.append(estimator)
        clf.fit(x_train,y_train)
        clf_pred=clf.predict(x_test)
        #F1_Score
        F1Score=f1_score(clf_pred,y_test,average='micro')
        F1_score.append(F1Score*100)
        #acc=accuracy_score(y_test,clf_pred)
        #ll=log_los(y_test,clf_pred.toaray())
        #print('accuracy',acc)
        #print('log_loss',ll)
```


In [45]:
```
#Finalizing the Results
scores=pd.DataFrame({'Model':Model,'Estimator':Estimator,'F1_Score':F1_score})
scores
```

Out[45]:

	Model	Estimator	F1_Score
0	LogisticRegression()	<class 'skmultilearn.problem_transform.br.Bina...	66.313028
1	LogisticRegression()	<class 'skmultilearn.problem_transform.cc.Clas...	67.614694
2	MultinomialNB()	<class 'skmultilearn.problem_transform.br.Bina...	59.639007
3	MultinomialNB()	<class 'skmultilearn.problem_transform.cc.Clas...	57.976950
4	LinearSVC(random_state=0)	<class 'skmultilearn.problem_transform.br.Bina...	67.653981
5	LinearSVC(random_state=0)	<class 'skmultilearn.problem_transform.cc.Clas...	68.053542

- HYPERPARAMETER TUNING :-

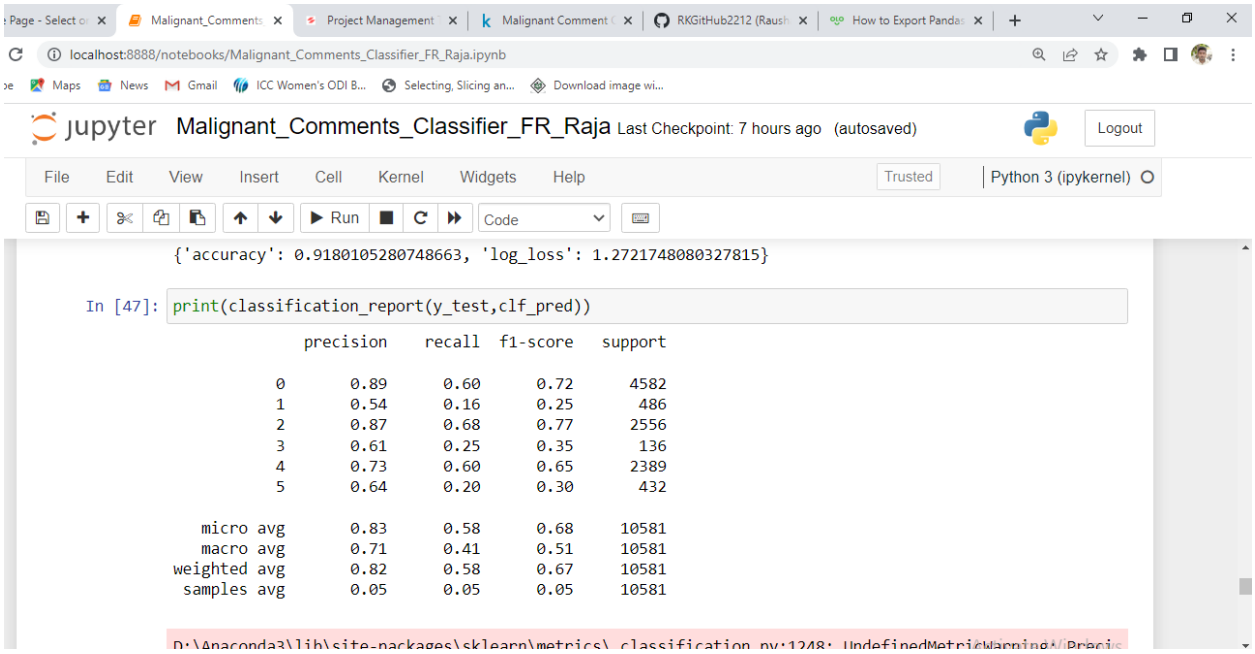
HYPERPARAMETER TUNING

In [46]:
```
clf=ClassifierChain(LinearSVC(random_state=0))
clf.fit(x_train,y_train)
clf_pred=clf.predict(x_test)
acc=accuracy_score(y_test,clf_pred)
ll=log_loss(y_test,clf_pred.toarray())
print({'accuracy':acc,'log_loss':ll})
```

```
{'accuracy': 0.9180105280748663, 'log_loss': 1.2721748080327815}
```

In [47]:
```
print(classification_report(y_test,clf_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.60      0.72      4582
           1       0.54      0.16      0.25       486
           2       0.87      0.68      0.77      2556
           3       0.61      0.25      0.35       136
```



- **PREDICTIONS ON TESTING DATA :-**

In [50]:
```
dftest.shape
```

Out[50]:
```
(153164, 2)
```

In [51]:
```
dftest.drop('id',axis=1,inplace=True)
dftest.head(5)
```

Out[51]:

	comment_text
0	Yo bitch Ja Rule is more succesful then you'll...
1	== From RfC == \n\n The title is fine as it is...
2	" \n\n == Sources == \n\n * Zawe Ashton on Lap...
3	:If you have a look back at the source, the in...
4	I don't anonymously edit articles at all.

DATA CLEANING :-

In [52]:
```
dftest['comment_textnew']=dftest['comment_text'].apply(nfx.remove_stopwords) #removing stopwords
```

In [53]:
```
#Removing punctuations, special character and URL'S
dftest['comment_textnew']=dftest['comment_textnew'].apply(nfx.remove_punctuations)
dftest['comment_textnew']=dftest['comment_textnew'].apply(nfx.remove_special_characters)
dftest['comment_textnew']=dftest['comment_textnew'].apply(nfx.remove_phone_numbers)
dftest['comment_textnew']=dftest['comment_textnew'].apply(nfx.remove_urls)
```

In [54]:
```
dftest['length']=dftest.comment_text.str.len()
dftest['clean_length']=dftest.comment_textnew.str.len()
dftest.head()
```

Out[54]:

	comment_text	comment_textnew	length	clean_length
0	Yo bitch Ja Rule is more succesful then you'll...	Yo bitch Ja Rule succesful whats hating sad mo...	367	234

Out[54]:

	comment_text	comment_textnew	length	clean_length
0	Yo bitch Ja Rule is more succesful then you'll...	Yo bitch Ja Rule succesful whats hating sad mo...	367	234
1	== From RfC == \n\n The title is fine as it is...	RfC title fine is IMO	50	23
2	" \n\n == Sources == \n\n * Zawe Ashton on Lap...	Sources Zawe Ashton Lapland	54	34
3	:If you have a look back at the source, the in...	If look source information updated correct for...	205	107
4	I don't anonymously edit articles at all.	anonymously edit articles all	41	29

Test Feature Extraction

In [55]:
```
#Vectorization
xtest=tfidf.fit_transform(dftest['comment_textnew'])
```

In [56]:
```
#Loading the model
test_model=joblib.load('Malignant_Comments_Classifier_FR.pkl')
```

TEST DATA PREDICTIONS

In [57]:
```
dftest_pred=test_model.predict(xtest)
dftest_predict=dftest_pred.toarray()
Malignant_Comment_predictions=pd.DataFrame(dftest_predict)
```

In [58]:
```
Malignant_Comment_predictions.sample(10)
```

Out[58]:

	0	1	2	3	4	5
131229	0.0	0.0	0.0	0.0	0.0	0.0
70869	0.0	0.0	0.0	0.0	0.0	0.0
142639	0.0	0.0	0.0	0.0	0.0	0.0
41137	0.0	0.0	0.0	0.0	0.0	0.0
1378	0.0	0.0	0.0	0.0	0.0	0.0

Saving the test output in csv format

In [64]:
```
Malignant_Comment_predictions.to_csv(r'D:\Malignant Comments Classifier Project\Malignant_test.csv',inc
```

Conclusion

We can see that the Linear SVC performs well with 'accuracy':0.919 to the given multi-classification dataset

In [ ]:
```
```

END