In [36]:
```
#Load the dataset

import pandas as pd



dataset = pd.read_csv(r"C:\Users\Vaish\Desktop\NLP(AD)\hate_speech.csv")

dataset.head()
```

Out[36]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation

In [38]:
```
dataset.shape
```

Out[38]:
```
(5242, 3)
```

In [40]:
```
dataset.label.value_counts()
```

Out[40]:
```
label
0    3000
1    2242
Name: count, dtype: int64
```

In [42]:
```
for index, tweet in enumerate(dataset["tweet"][10:15]):

    print(index+1,"-",tweet)
```

```
1 -  â #ireland consumer price index (mom) climbed from previous 0.2% to 0.5% in m
ay   #blog #silver #gold #forex
2 - we are so selfish. #orlando #standwithorlando #pulseshooting #orlandoshooting #b
iggerproblems #selfish #heabreaking   #values #love #
3 - i get to see my daddy today!!   #80days #gettingfed
4 - ouch...junior is angryð#got7 #junior #yugyoem   #omg
5 - i am thankful for having a paner. #thankful #positive
```

In [44]:
```
import re

#Clean text from noise

def clean_text(text):

    #Filter to allow only alphabets

    text = re.sub(r'[^a-zA-Z\']', ' ', text)
    #Remove Unicode characters

    text = re.sub(r'[^\x00-\x7F]+', ' ', text)

    #Convert to lowercase to maintain consistency

    text = text.lower()

    return text
```

In [46]:
```
dataset['clean_text'] = dataset.tweet.apply(lambda x: clean_text(x))
```

In [48]:
```
dataset['clean_text'] = dataset.tweet.apply(lambda x: clean_text(x))
```

In [50]:
```
dataset.head(10)
```

Out[50]:

	id	label	tweet	clean_text
0	1	0	@user when a father is dysfunctional and is s...	user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...	user user thanks for lyft credit i can't us...
2	3	0	bihday your majesty	bihday your majesty
3	4	0	#model i love u take with u all the time in ...	model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation	factsguide society now motivation
5	6	0	[2/2] huge fan fare and big talking before the...	huge fan fare and big talking before the...
6	7	0	@user camping tomorrow @user @user @user @use...	user camping tomorrow user user user use...
7	8	0	the next school year is the year for exams.ð□□...	the next school year is the year for exams ...
8	9	0	we won!!! love the land!!! #allin #cavs #champ...	we won love the land allin cavs champ...
9	10	0	@user @user welcome here ! i'm it's so #gr...	user user welcome here i'm it's so gr...

In [52]:
```
from nltk.corpus import stopwords
len(stopwords.words('english'))
```

Out[52]:
```
179
```

In [54]:
```
stop = stopwords.words('english')
```

In [56]:
```
#Generate word frequency
```

```
def gen_freq(text):

    #Will store the list of words

    word_list = []

    #Loop over all the tweets and extract words into word_list

    for tw_words in text.split():

        word_list.extend(tw_words)

    #Create word frequencies using word_list

    word_freq = pd.Series(word_list).value_counts()

    #Drop the stopwords during the frequency calculation

    word_freq = word_freq.drop(stop, errors='ignore')

    return word_freq
```

In [58]:
```
#Check whether a negation term is present in the text

def any_neg(words):

    for word in words:

        if word in ['n', 'no', 'non', 'not'] or re.search(r"\wn't", word):

            return 1

        else:

            return 0
```

In [60]:
```
def any_rare(words,rare_100):
    for word in words:
        if word in rare_100:
            return 1
        else:
            return 0
```

In [62]:
```
#Check whether prompt words are present

def is_question(words):

    for word in words:

        if word in ['when', 'what', 'how', 'why', 'who', 'where']:

            return 1

        else:
            return 0
```

In [86]:
```
word_freq = gen_freq(dataset.clean_text.str)

#100 most rare words in the dataset

rare_100 = word_freq[-100:] # last 100 rows/words

#Number of words in a tweet

dataset['word_count'] = dataset.clean_text.str.split().apply(lambda x: len(x))

#Negation present or not

dataset['any_neg'] = dataset.clean_text.str.split().apply(lambda x: any_neg(x))

#Prompt present or not

dataset['is_question'] = dataset.clean_text.str.split().apply(lambda x: is_question

#Any of the most 100 rare words present or not

dataset['any_rare'] = dataset.clean_text.str.split().apply(lambda x: any_rare(x, ra

#Character count of the tweet

dataset['char_count'] = dataset.clean_text.apply(lambda x: len(x))
```

In [67]:
```
dataset.head(10)
```

Out[67]:

	id	label	tweet	clean_text	word_count	any_neg	is_question	any_rare	char_
0	1	0	@user when a father is dysfunctional and is s...	user when a father is dysfunctional and is s...	18	0	0	0	
1	2	0	@user @user thanks for #lyft credit i can't us...	user user thanks for lyft credit i can't us...	19	0	0	0	
2	3	0	bihday your majesty	bihday your majesty	3	0	0	0	
3	4	0	#model i love u take with u all the time in ...	model i love u take with u all the time in ...	12	0	0	0	
4	5	0	factsguide: society now #motivation	factsguide society now motivation	4	0	0	0	
5	6	0	[2/2] huge fan fare and big talking before the...	huge fan fare and big talking before the...	18	0	0	0	
6	7	0	@user camping tomorrow @user @user @user @use...	user camping tomorrow user user user use...	11	0	0	0	
7	8	0	the next school year is the year for exams.ð□□...	the next school year is the year for exams ...	20	0	0	0	
8	9	0	we won!!! love the land!!! #allin #cavs #champ...	we won love the land allin cavs champ...	10	0	0	0	
9	10	0	@user @user welcome here ! i'm it's so #gr...	user user welcome here i'm it's so gr...	8	0	0	0	

In [71]:
```
from sklearn.model_selection import train_test_split
X = dataset[['word_count', 'any_neg', 'any_rare', 'char_count', 'is_question']]
y = dataset.label
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,

                                                    random_state=0)
```

In [76]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model= model.fit(X_train,y_train)
pred=model.predict(X_test)
```

In [78]:
```
model.predict(X_test[5:10])
```

Out[78]:
```
array([0, 1, 1, 1, 1], dtype=int64)
```

In [80]:
```
from sklearn.metrics import accuracy_score



print("Accuracy:", accuracy_score(y_test, pred)*100, "%")
```

```
Accuracy: 42.99332697807436 %
```

In [82]:
```
from sklearn.ensemble import RandomForestClassifier
clf_rf=RandomForestClassifier()
clf_rf.fit(X_train,y_train)
rf_pred=clf_rf.predict(X_test).astype(int)
```

In [90]:
```
from sklearn.metrics import classification_report,confusion_matrix,\
accuracy_score
print(confusion_matrix(y_test,rf_pred))
print(classification_report(y_test,rf_pred))
print("Accuracy:",accuracy_score(y_test,rf_pred))
```

```
[[399 200]
 [234 216]]
              precision    recall  f1-score   support

           0       0.63      0.67      0.65       599
           1       0.52      0.48      0.50       450

    accuracy                           0.59      1049
   macro avg       0.57      0.57      0.57      1049
weighted avg       0.58      0.59      0.58      1049

Accuracy: 0.5862726406101049
```

In [92]:
```
from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression(class_weight='balanced')

logreg.fit(X_train, y_train)
```

Out[92]:

```
▾ LogisticRegression

LogisticRegression(class_weight='balanced')
```

In [94]:

```
y_pred = logreg.predict(X_test) #predicting the values
```

In [96]:

```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.63      0.58      0.61       599
           1       0.49      0.54      0.52       450

    accuracy                           0.57      1049
   macro avg       0.56      0.56      0.56      1049
weighted avg       0.57      0.57      0.57      1049
```

In [ ]: