In [1]:
```
import pandas as pd
dataset = pd.read_csv('hate_speech.csv')
dataset.head()
```

Out[1]:

	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 [2]:
```
dataset.label.value_counts()
```

Out[2]:
```
0    3000
1    2242
Name: label, dtype: int64
```

In [3]:
```
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 may   #blog #silver #gold #forex
2 - we are so selfish. #orlando #standwithorlando #pulseshooting #orlandoshooting #biggerproblems #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 [5]:
```
import re
def clean_text(text):
    text = re.sub(r'[^a-zA-Z\']','', text)
    text = re.sub(r'[^\x00-\x7F]+','', text)
    text = text.lower()
    return text
```

In [6]:
```
dataset['clean_text'] = dataset.tweet.apply(lambda x: clean_text(x))
```

In [7]:
```
from nltk.corpus import stopwords
len(stopwords.words('english'))
```

Out[7]:
```
179
```

In [8]:
```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\nihar\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[8]:
```
True
```

In [21]:
```
def gen_freq(text):
    word_list = []
    for tw_words in text.split():
        word_list.extend(tw_words)
    word_freq = pd.Series(word_list).value_counts()
    stop = stopwords.words('english')
    word_freq = word_freq.drop(stop, errors='ignore')
    return word_freq
```

In [22]:
```
def any_neg(words):
    for word in words:
        if word in ['n', 'no', 'non', 'not'] or re.search(r"\wn't", wor
            return 1
        else:
            return 0
```

In [23]:
```
def any_rare(words,rare_100):
      for word in words:
        if word in rare_100:
              return 1
        else:
            return 0
```

In [24]:
```
def is_question(words):
      for word in words:
        if word in ["when","what","how","why","who"]:
              return 1
        else:
              return 0
```

In [25]:
```
word_freq=gen_freq(dataset.clean_text.str)
rare_100=word_freq[-100:]
dataset['word_count']=dataset.clean_text.str.split().apply(lambda x:len
dataset['any_neg']=dataset.clean_text.str.split().apply(lambda x:any_ne
dataset['is_question']=dataset.clean_text.str.split().apply(lambda x:is
dataset['any_rare']=dataset.clean_text.str.split().apply(lambda x:any_r
dataset['char_count']=dataset.clean_text.apply(lambda x:len(x))
```

In [ ]:
```

```