In [1]:

```python
#Load the dataset

import pandas as pd



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

```python
dataset.shape
```

Out[2]:

```
(5242, 3)
```

In [3]:

```python
dataset.label.value_counts()
```

Out[3]:

```
label
0    3000
1    2242
Name: count, dtype: int64
```

In [4]:

```python
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]:

```python
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 [11]:
```
dataset['clean_text'] = dataset.tweet.apply(lambda x: clean_text(x))
```

In [13]:
```
dataset['clean_text'] = dataset.tweet.apply(lambda x: clean_text(x))
```

In [15]:
```
dataset.head(10)
```

Out[15]:

	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 [17]:
```
from nltk.corpus import stopwords
len(stopwords.words('english'))
```

Out[17]:
```
179
```

In [19]:
```
stop = stopwords.words('english')
```

In [25]:
```
#Generate word frequency
```

```python
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 [27]:

```python
#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 [29]:

```python
def any_rare(words,rare_100):
    for word in words:
        if word in rare_100:
            return 1
        else:
            return 0
```

In [31]:

```python
#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 [33]:

```
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 [ ]: