```
import pandas as pd
dataset=pd.read_csv("hate_speech.csv")
dataset
```

	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
...	...	...	...
5237	31935	1	lady banned from kentucky mall. @user #jcpenn...
5238	31947	1	@user omfg i'm offended! i'm a mailbox and i'...
5239	31948	1	@user @user you don't have the balls to hashta...
5240	31949	1	makes you ask yourself, who am i? then am i a...
5241	31961	1	@user #sikh #temple vandalised in in #calgary,...

5242 rows × 3 columns

Next steps: Generate code with dataset | View recommended plots | New interactive sheet

```
dataset.shape
```

```
(5242, 3)
```

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

label	count
0	3000
1	2242

```
import pandas as pd
dataset=pd.read_csv("hate_speech.csv")
dataset
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
```

```
import re

# Clean text from noise
def clean_text(text):
    text = re.sub(r'[^a-zA-Z\s]', ' ', text)

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

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

```
from nltk.corpus import stopwords
len(stopwords.words('english'))
```

```
179
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
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
```

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

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

Generated code may be subject to a license | RajarsiGit/Basic_ML_Model_for_Text_Classification

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

Generated code may be subject to a license | RuthNduta/Natural-Language-Processing | RajarsiGit/Basic_ML_Model_for_Text_Classification

```
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(x))
dataset['any_neg']=dataset.clean_text.str.split().apply(lambda x:any_neg(x))
dataset['is_question']=dataset.clean_text.str.split().apply(lambda x:is_question(x))
dataset['any_rare']=dataset.clean_text.str.split().apply(lambda x:any_rare(x,rare_100))
dataset['char_count']=dataset.clean_text.apply(lambda x:len(x))
```

+ Code + Text

Start coding or generate with AI.