

```
In [2]: import numpy as np
import pandas as pd
```

```
In [4]: df=pd.read_csv("C:/Users/HP/Downloads/stress.csv")
df.head()
```

Out[4]:

	subreddit	post_id	sentence_range	text	id	label	confidence	social_timestamp	score
0	ptsd	8601tu	(15, 20)	He said he had not felt that way before, suggest...	33181	1	0.8	1521614353	1
1	assistance	8lbrx9	(0, 5)	Hey there r/assistance, Not sure if this is th...	2606	0	1.0	1527009817	1
2	ptsd	9ch1zh	(15, 20)	My mom then hit me with the newspaper and it s...	38816	1	0.8	1535935605	1
3	relationships	7rorpp	[5, 10]	until i met my new boyfriend, he is amazing, h...	239	1	0.6	1516429555	1
4	survivorsofabuse	9p2gbc	[0, 5]	October is Domestic Violence Awareness Month a...	1421	1	0.8	1539809005	1

5 rows × 116 columns



```
In [5]: df.describe()
```

Out[5]:

	id	label	confidence	social_timestamp	social_karma	syntax_ari	lex_liwc_W
count	2838.000000	2838.000000	2838.000000	2.838000e+03	2838.000000	2838.000000	2838.000000
mean	13751.999295	0.524313	0.808972	1.518107e+09	18.262156	4.684272	85.99612
std	17340.161897	0.499497	0.177038	1.552209e+07	79.419166	3.316435	32.33488
min	4.000000	0.000000	0.428571	1.483274e+09	0.000000	-6.620000	5.00000
25%	926.250000	0.000000	0.600000	1.509698e+09	2.000000	2.464243	65.00000
50%	1891.500000	1.000000	0.800000	1.517066e+09	5.000000	4.321886	81.00000

	id	label	confidence	social_timestamp	social_karma	syntax_ari	lex_liwc_W
75%	25473.750000	1.000000	1.000000	1.530898e+09	10.000000	6.505657	101.00000
max	55757.000000	1.000000	1.000000	1.542592e+09	1435.000000	24.074231	310.00000

8 rows × 112 columns



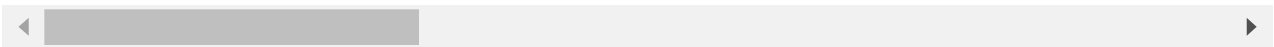
In [6]:

df.isnull()

Out[6]:

	subreddit	post_id	sentence_range	text	id	label	confidence	social_timestamp	social_karma
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
2833	False	False	False	False	False	False	False	False	False
2834	False	False	False	False	False	False	False	False	False
2835	False	False	False	False	False	False	False	False	False
2836	False	False	False	False	False	False	False	False	False
2837	False	False	False	False	False	False	False	False	False

2838 rows × 116 columns



In [7]:

df.isnull().sum()

Out[7]:

subreddit	0
post_id	0
sentence_range	0
text	0
id	0
..	
lex_dal_avg_pleasantness	0
social_upvote_ratio	0
social_num_comments	0
syntax_fk_grade	0
sentiment	0
Length: 116, dtype: int64	

In [8]:

import nltk  
import re  
from nltk. corpus import stopwords  
import string  
nltk. download( 'stopwords' )

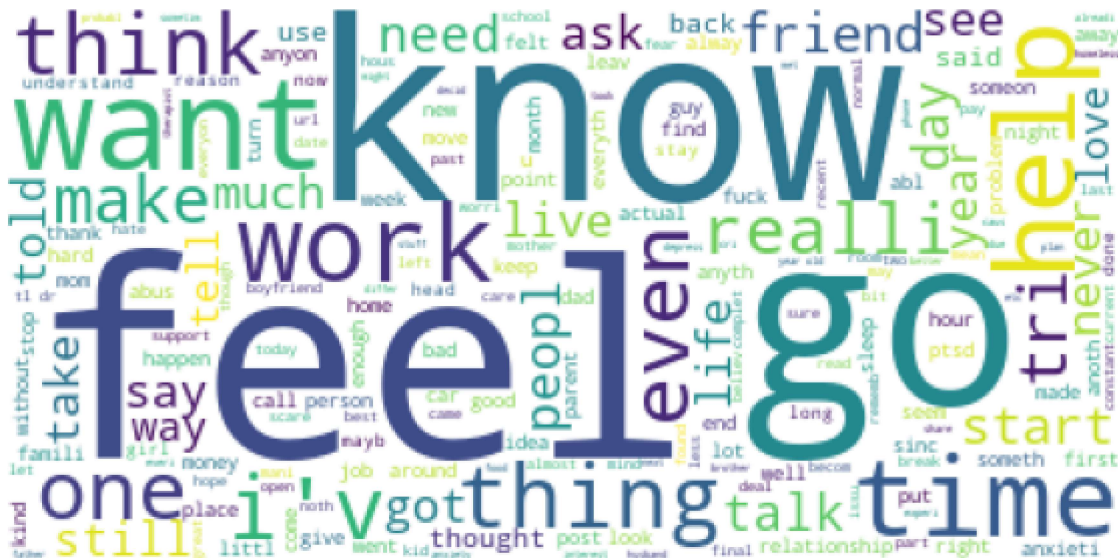
```
stemmer = nltk.SnowballStemmer("english")
stopword=set (stopwords . words ( 'english' ))

def clean(text):
    text = str(text) . lower() #returns a string where all characters are lower case.
    text = re. sub('\[. *? \]', ' ',text) #substring and returns a string with replaced v
    text = re. sub('https?:\/\/\S+\/www\ . \S+', ' ', text)#whitespace char with pattern
    text = re. sub('< . *? >+', ' ', text)#special char enclosed in square brackets
    text = re. sub(' [%s]' % re. escape(string. punctuation), ' ', text)#eliminate punc
    text = re. sub(' \n', ' ', text)
    text = re. sub(' \w*\d\w*', ' ', text)#word character ASCII punctuation
    text = [word for word in text. split(' ') if word not in stopwords] #removing stopw
    text = " ". join(text)
    text = [stemmer . stem(word) for word in text. split(' ') ]#remove morphological af
    text = " ". join(text)
    return text

df [ "text" ] = df["text"]. apply(clean)
```

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

```
In [12]: import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
text = " ".join(i for i in df.text)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, background_color="white") . generate(text)
plt.figure(figsize=(10, 15))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```



```
In [15]: from sklearn. feature_extraction. text import CountVectorizer
         from sklearn. model_selection import train_test_split

         x = np.array (df["text"])
         y = np.array (df["label"])

         cv = CountVectorizer ()
```

```
X = cv.fit_transform(x)
print(X)
xtrain, xtest, ytrain, ytest = train_test_split(X, y, test_size=0.33, random_state=42)
```

```
(0, 7405)      1
(0, 3278)      1
(0, 9454)      1
(0, 861)       1
(0, 8359)      1
(0, 3750)      1
(0, 7214)      1
(0, 8908)      1
(0, 298)       1
(0, 9749)      1
(0, 4303)      1
(0, 5034)      1
(0, 5325)      1
(0, 2188)      1
(0, 5118)      1
(0, 3265)      1
(0, 2593)      3
(0, 4188)      1
(0, 5316)      1
(0, 3697)      1
(0, 8339)      1
(0, 6861)      1
(0, 4150)      1
(0, 5174)      1
(0, 1831)      1
:
(2836, 877)    1
(2836, 4555)   1
(2836, 2928)   1
(2836, 4615)   1
(2836, 4785)   1
(2836, 4511)   1
(2837, 7405)   2
(2837, 3018)   1
(2837, 5533)   2
(2837, 8784)   1
(2837, 8502)   1
(2837, 6770)   1
(2837, 4318)   1
(2837, 9670)   1
(2837, 5569)   1
(2837, 8881)   1
(2837, 5713)   1
(2837, 2587)   1
(2837, 7468)   1
(2837, 2351)   1
(2837, 7804)   1
(2837, 2758)   1
(2837, 8880)   1
(2837, 5459)   1
(2837, 3020)   1
```

```
In [14]: from sklearn.naive_bayes import BernoulliNB
model=BernoulliNB()
model.fit(xtrain,ytrain)
```

Out[14]: BernoulliNB()

```
In [17]: user=input("Enter the text")
data=cv.transform([user]).toarray()
output=model.predict(data)
print(output)
```

Enter the text i think we need to take care of ourselves  
[0]

```
In [19]: user=input("Enter the text")
data=cv.transform([user]).toarray()
output=model.predict(data)
print(output)
```

Enter the text i am sad  
[1]