```
import numpy as np
import pandas as pd
```

```
df=pd.read_csv('/content/stress.csv')
df.head()
```

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

5 rows × 116 columns

```
df.describe()
```

	id	label	confidence	social_timestamp	social_karma	syntax_ari	lex_liwc_WC
count	2185.000000	2185.000000	2185.000000	2.185000e+03	2185.000000	2185.000000	2185.000000
mean	13970.466362	0.518078	0.806893	1.517997e+09	18.452174	4.700834	86.035240
std	17509.350310	0.499787	0.176911	1.546128e+07	78.752148	3.315674	32.883955
min	4.000000	0.000000	0.500000	1.483274e+09	0.000000	-6.620000	5.000000
25%	951.000000	0.000000	0.600000	1.510162e+09	2.000000	2.495056	65.000000
50%	1940.000000	1.000000	0.800000	1.516987e+09	5.000000	4.309623	81.000000
75%	25827.000000	1.000000	1.000000	1.530755e+09	11.000000	6.481267	101.000000
max	55757.000000	1.000000	1.000000	1.542592e+09	1435.000000	24.074231	310.000000

8 rows × 112 columns

```
df.isnull().sum()
```

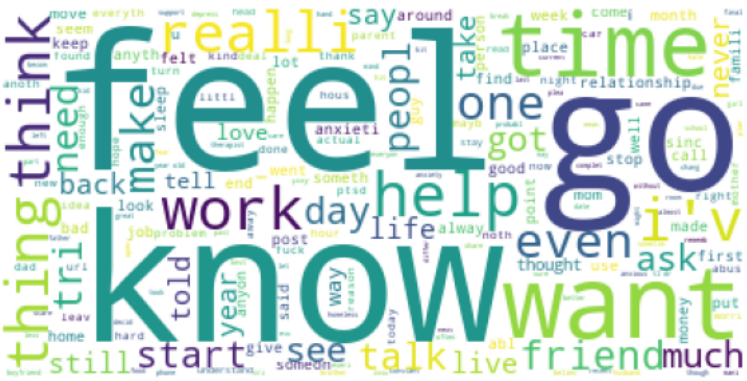
```
subreddit                  0
post_id                    0
sentence_range             0
text                       0
id                         0
                          ..
lex_dal_avg_pleasantness   1
social_upvote_ratio        1
social_num_comments        1
syntax_fk_grade            1
sentiment                  1
Length: 116, dtype: int64
```

```
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. Symbols and Numbers are ignored.
    text = re. sub('\[.*?\]',' ',text)  #substring and returns a string with replaced values.
    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 punctuation from string
    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 stopword]  #removing stopwords
    text =" ". join(text)
    text = [stemmer . stem(word) for word in text. split(' ') ]#remove morphological affixes from words
    text = " ". join(text)
    return text
df [ "text"] = df["text"]. apply(clean)
```

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

```
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, 10) )
plt. imshow(wordcloud )
plt. axis("off")
plt. show( )
```




```
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)
```

```
  (0, 6558)     1
  (0, 2894)     1
  (0, 8381)     1
  (0, 758)      1
  (0, 7415)     1
  (0, 3315)     1
  (0, 6388)     1
  (0, 7915)     1
  (0, 253)      1
  (0, 8650)     1
  (0, 3812)     1
  (0, 4459)     1
  (0, 4714)     1
  (0, 1929)     1
  (0, 4534)     1
  (0, 2882)     1
  (0, 2291)     3
  (0, 3711)     1
  (0, 4706)     1
```

```
(0, 3269)     1
(0, 7396)     1
(0, 6078)     1
(0, 3675)     1
(0, 4581)     1
(0, 1610)     1
:     :
(2184, 2334)  1
(2184, 7792)  1
(2184, 4256)  1
(2184, 7715)  1
(2184, 3617)  2
(2184, 652)   2
(2184, 5096)  1
(2184, 1888)  1
(2184, 410)   1
(2184, 1622)  1
(2184, 6922)  2
(2184, 1611)  2
(2184, 2500)  1
(2184, 8514)  1
(2184, 4564)  1
(2184, 5308)  2
(2184, 6555)  1
(2184, 780)   1
(2184, 1064)  1
(2184, 402)   1
(2184, 7057)  1
(2184, 8582)  1
(2184, 5898)  1
(2184, 7046)  1
(2184, 5251)  1
```

```
from sklearn.naive_bayes import BernoulliNB
model=BernoulliNB()
model.fit(xtrain,ytrain)
```

```
BernoulliNB()
```

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

```
Enter the texti am feeling happy
[1]
```

✓ 5s completed at 06:39

Literature Survey:

Sr. No	Title of Paper	Name of Authors	Published Year	Remarks
1	Automatic Stress Detection Using Wearable Sensors and Machine Learning	Shruti Gedam, Sanchita Paul	July 1-3, 2020	In our day to day life there are many stressors like **Physical, Psychological, Environmental, and Psychosocial** Detecting the stress we can use many technology in this project we will be use **Wearable Sensors** And for extracted the features using various **machine learning algorithms** features extracted using **Heart rate**, **Heart rate skin conductance** are more useful in prediction of stress level **Random forest** and **K-Nearest Neighbor** this algorithms are used
2	A Decision Tree Optimised SVM Model for Stress Detection using Biosignals	Alana Paul Cruz, Aravind Pradeep, Kavali Riya Sivasankar and Krishnaveni K.S	2020	In this project for stress detection **SVM** model is use **Linear** 88.9% **Quadratic** 92.6% **Cubic** 96.3% this is a type of svm model. Following component are used heart rate,pr interval, OR interval, Qt interval, ST segment In our work we selected **ECG** as the bio signal and extracted its features. **EDR** (ECG Derived Respiration) feature can be easily derived without any extra sensors. Among those unique features we chose **ECG** derived Respiration, Respiration Rate,
3	Stress Detection with Machine Learning and Deep Learning using Multimodal Physiological Data	Pramod Bobade, Vani M	2020	stress detection on individuals using multimodal dataset recorded from **wearable physiological** and **motion sensors**, which can prevent a person from various stress□related health problems. Data of sensor modalities like three□axis acceleration (**ACC**), electrocardiogram (**ECG**), blood volume pulse (**BVP**), body temperature (**TEMP**), respiration (**RESP**), electromyogram (**EMG**) and electrodermal activity (**EDA**) are for three physiological conditions
4	Machine Learning and IoT for Prediction and Detection of Stress	Mr. Purnendu Shekhar Pandey	2017	We applied **SVM a**nd **Logistic Regression** which show considerable improvement over **VF - 15** and **Naive Bayes** without any external weights provided. In the weighted VF - 15 the accuracy was comparable to SVM Logistic Regression 100 % 66 % SVM 97 % 68 %. Components Used are **Node MCU & Pulse Sensor**