ML PROJECT:Stress Detection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr NO | Title of Paper | Name of Author | Published Year | Remarks |
| 1 | Automatic Stress Detection Using Wearable Sensors and Machine Learning | Shruti Gedam  Sanchita Paul | 2020 | Support Vector Machines (SVM), Logistic regression, K-Nearest Neighbor, Decision tree and Random forest |
| 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 | Optimised Support Vector Machines (SVM) using decision trees |
| 3 | Stress Detection with Machine Learning and Deep Learning using Multimodal Physiological Data | Pramod Bobade  Vani M. | 2020 | machine learning and deep learning classification methods |
| 4 | Stress detection using deep neural networks | Russell Li1 and Zhandong Liu | 2020 | two deep neural networks: a deep 1D convolutional neural network and a deep multilayer perceptron neural network |

import numpy as np

import pandas as pd

import seaborn as sns

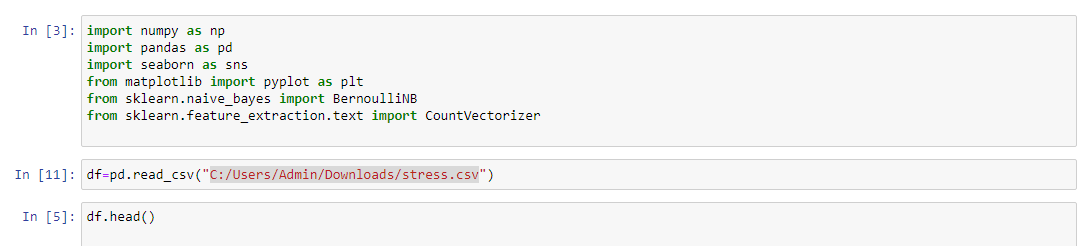
from matplotlib import pyplot as plt

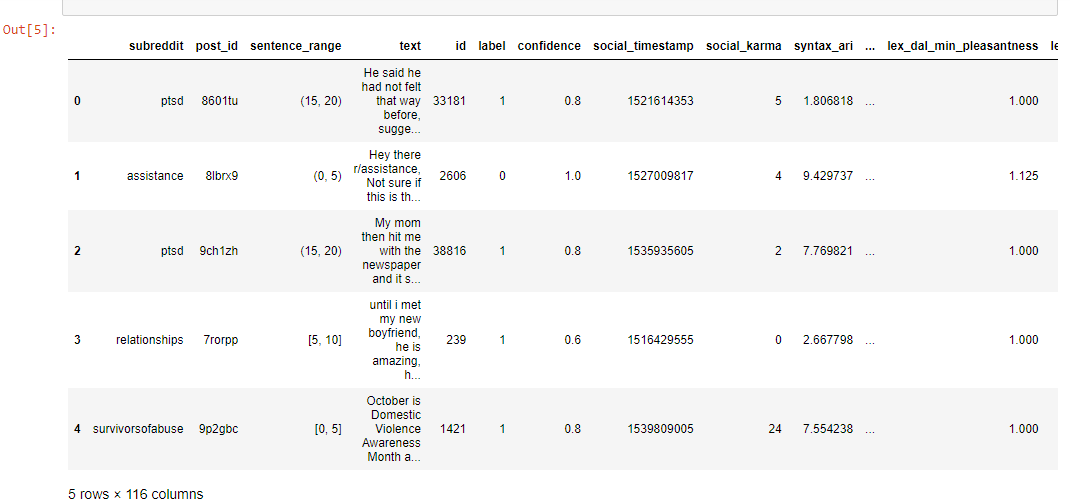
from sklearn.naive\_bayes import BernoulliNB

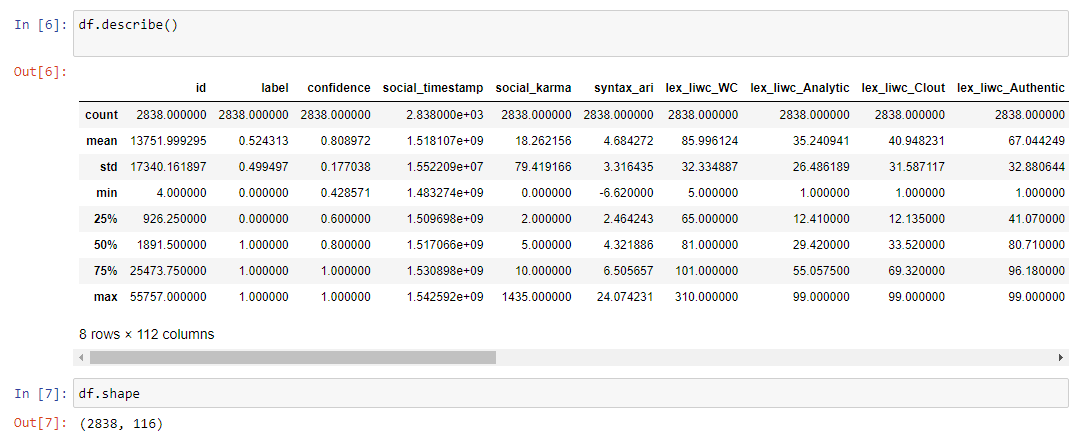
from sklearn.feature\_extraction.text import CountVectorizer

df=pd.read\_csv("C:/Users/Admin/Downloads/stress.csv")

df.head()









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

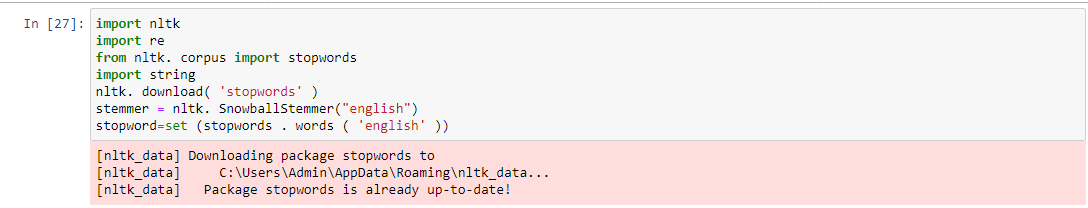
text = [word for word in text. split(' ') if word not in stopword] #removing stopwords

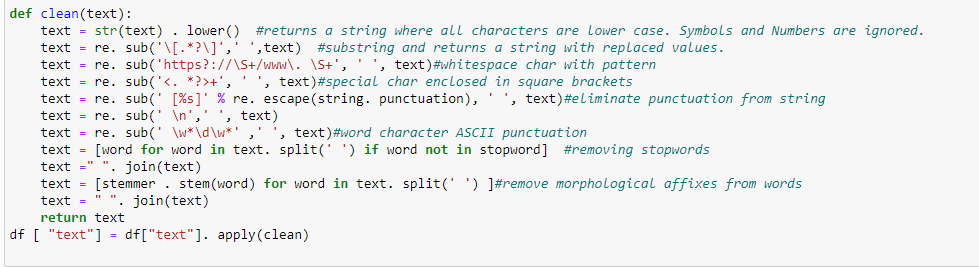
text =" ". join(text)

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)



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"])

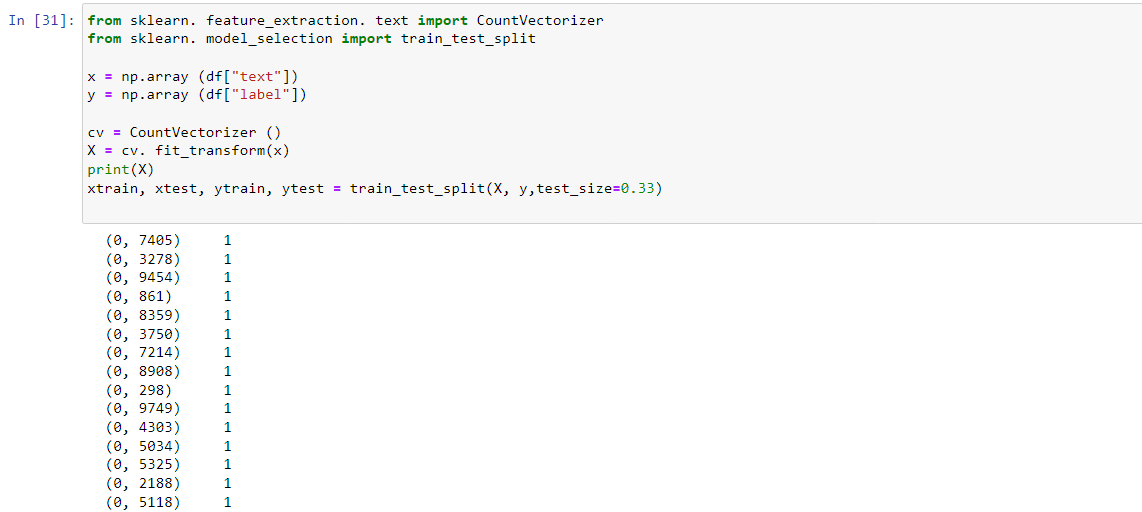
y = np.array (df["label"])

cv = CountVectorizer ()

X = cv. fit\_transform(x)

print(X)

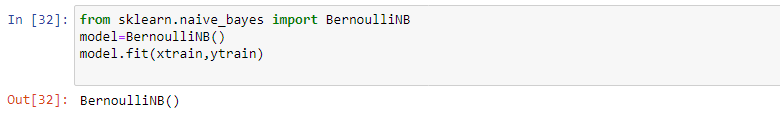
xtrain, xtest, ytrain, ytest = train\_test\_split(X, y,test\_size=0.33)



from sklearn.naive\_bayes import BernoulliNB

model=BernoulliNB()

model.fit(xtrain,ytrain)



user=input("Enter the text")

data=cv.transform([user]).toarray()

output=model.predict(data)

print(output)

