** FAKE NEWS DETECTION USING PYTHON PROJECT**

**CODE:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import matplotlib

import seaborn as sns

import itertools

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import PassiveAggressiveClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix

*#default theme*

plt.style.use('ggplot')

sns.color\_palette("tab10")

sns.set(context='notebook', style='darkgrid', font='sans-serif', font\_scale=1, rc=None)

matplotlib.rcParams['figure.figsize'] =[20,8]

matplotlib.rcParams.update({'font.size': 15})

matplotlib.rcParams['font.family'] = 'sans-serif'

*Read the data*

df=pd.read\_csv('../input/textdb3/fake\_or\_real\_news.csv')

*#Get shape and head*

print(df.shape)

df.head()

*#DataFlair - Get the labels*

labels=df.label

labels.head()

target=df.label.value\_counts()

target

sns.countplot(df.label)

plt.title('the number of news fake/real);

*#DataFlair - Split the dataset*

x\_train,x\_test,y\_train,y\_test=train\_test\_split(df['text'], labels, test\_size=0.2, random\_state=7)

*#DataFlair - Initialize a TfidfVectorizer*

tfidf\_vectorizer=TfidfVectorizer(stop\_words='english', max\_df=0.7)

*#DataFlair - Fit and transform train set, transform test set*

tfidf\_train=tfidf\_vectorizer.fit\_transform(x\_train)

tfidf\_test=tfidf\_vectorizer.transform(x\_test)

*#DataFlair - Initialize a PassiveAggressiveClassifier*

pac=PassiveAggressiveClassifier(max\_iter=50)

pac.fit(tfidf\_train,y\_train)

*#DataFlair - Predict on the test set and calculate accuracy*

y\_pred=pac.predict(tfidf\_test)

score=accuracy\_score(y\_test,y\_pred)

print(f'Accuracy: **{**round(score\*100,2)**}**%')

*#DataFlair - Build confusion matrix*

confusion\_matrix(y\_test,y\_pred, labels=['FAKE','REAL'])