Predict Password Strength with Natural Language Processing by Patrick BENIE

March 12, 2021

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[3]:
```
data=pd.read_csv('/Users/patrickslearningprogams/Desktop/Python Projects/
 ↪Password Strength Predictor/data.csv',error_bad_lines=False)
data.head()
```

```
b'Skipping line 2810: expected 2 fields, saw 5\nSkipping line 4641: expected 2
fields, saw 5\nSkipping line 7171: expected 2 fields, saw 5\nSkipping line
11220: expected 2 fields, saw 5\nSkipping line 13809: expected 2 fields, saw
5\nSkipping line 14132: expected 2 fields, saw 5\nSkipping line 14293: expected
2 fields, saw 5\nSkipping line 14865: expected 2 fields, saw 5\nSkipping line
17419: expected 2 fields, saw 5\nSkipping line 22801: expected 2 fields, saw
5\nSkipping line 25001: expected 2 fields, saw 5\nSkipping line 26603: expected
2 fields, saw 5\nSkipping line 26742: expected 2 fields, saw 5\nSkipping line
29702: expected 2 fields, saw 5\nSkipping line 32767: expected 2 fields, saw
5\nSkipping line 32878: expected 2 fields, saw 5\nSkipping line 35643: expected
2 fields, saw 5\nSkipping line 36550: expected 2 fields, saw 5\nSkipping line
38732: expected 2 fields, saw 5\nSkipping line 40567: expected 2 fields, saw
5\nSkipping line 40576: expected 2 fields, saw 5\nSkipping line 41864: expected
2 fields, saw 5\nSkipping line 46861: expected 2 fields, saw 5\nSkipping line
47939: expected 2 fields, saw 5\nSkipping line 48628: expected 2 fields, saw
5\nSkipping line 48908: expected 2 fields, saw 5\nSkipping line 57582: expected
2 fields, saw 5\nSkipping line 58782: expected 2 fields, saw 5\nSkipping line
58984: expected 2 fields, saw 5\nSkipping line 61518: expected 2 fields, saw
5\nSkipping line 63451: expected 2 fields, saw 5\nSkipping line 68141: expected
2 fields, saw 5\nSkipping line 72083: expected 2 fields, saw 5\nSkipping line
74027: expected 2 fields, saw 5\nSkipping line 77811: expected 2 fields, saw
5\nSkipping line 83958: expected 2 fields, saw 5\nSkipping line 85295: expected
2 fields, saw 5\nSkipping line 88665: expected 2 fields, saw 5\nSkipping line
89198: expected 2 fields, saw 5\nSkipping line 92499: expected 2 fields, saw
5\nSkipping line 92751: expected 2 fields, saw 5\nSkipping line 93689: expected
2 fields, saw 5\nSkipping line 94776: expected 2 fields, saw 5\nSkipping line
```

```
97334: expected 2 fields, saw 5\nSkipping line 102316: expected 2 fields, saw
5\nSkipping line 103421: expected 2 fields, saw 5\nSkipping line 106872:
expected 2 fields, saw 5\nSkipping line 109363: expected 2 fields, saw
5\nSkipping line 110117: expected 2 fields, saw 5\nSkipping line 110465:
expected 2 fields, saw 5\nSkipping line 113843: expected 2 fields, saw
5\nSkipping line 115634: expected 2 fields, saw 5\nSkipping line 121518:
expected 2 fields, saw 5\nSkipping line 123692: expected 2 fields, saw
5\nSkipping line 124708: expected 2 fields, saw 5\nSkipping line 129608:
expected 2 fields, saw 5\nSkipping line 133176: expected 2 fields, saw
5\nSkipping line 135532: expected 2 fields, saw 5\nSkipping line 138042:
expected 2 fields, saw 5\nSkipping line 139485: expected 2 fields, saw
5\nSkipping line 140401: expected 2 fields, saw 5\nSkipping line 144093:
expected 2 fields, saw 5\nSkipping line 149850: expected 2 fields, saw
5\nSkipping line 151831: expected 2 fields, saw 5\nSkipping line 158014:
expected 2 fields, saw 5\nSkipping line 162047: expected 2 fields, saw
5\nSkipping line 164515: expected 2 fields, saw 5\nSkipping line 170313:
expected 2 fields, saw 5\nSkipping line 171325: expected 2 fields, saw
5\nSkipping line 171424: expected 2 fields, saw 5\nSkipping line 175920:
expected 2 fields, saw 5\nSkipping line 176210: expected 2 fields, saw
5\nSkipping line 183603: expected 2 fields, saw 5\nSkipping line 190264:
expected 2 fields, saw 5\nSkipping line 191683: expected 2 fields, saw
5\nSkipping line 191988: expected 2 fields, saw 5\nSkipping line 195450:
expected 2 fields, saw 5\nSkipping line 195754: expected 2 fields, saw
5\nSkipping line 197124: expected 2 fields, saw 5\nSkipping line 199263:
expected 2 fields, saw 5\nSkipping line 202603: expected 2 fields, saw
5\nSkipping line 209960: expected 2 fields, saw 5\nSkipping line 213218:
expected 2 fields, saw 5\nSkipping line 217060: expected 2 fields, saw
5\nSkipping line 220121: expected 2 fields, saw 5\nSkipping line 223518:
expected 2 fields, saw 5\nSkipping line 226293: expected 2 fields, saw
5\nSkipping line 227035: expected 2 fields, saw 7\nSkipping line 227341:
expected 2 fields, saw 5\nSkipping line 227808: expected 2 fields, saw
5\nSkipping line 228516: expected 2 fields, saw 5\nSkipping line 228733:
expected 2 fields, saw 5\nSkipping line 232043: expected 2 fields, saw
5\nSkipping line 232426: expected 2 fields, saw 5\nSkipping line 234490:
expected 2 fields, saw 5\nSkipping line 239626: expected 2 fields, saw
5\nSkipping line 240461: expected 2 fields, saw 5\nSkipping line 244518:
expected 2 fields, saw 5\nSkipping line 245395: expected 2 fields, saw
5\nSkipping line 246168: expected 2 fields, saw 5\nSkipping line 246655:
expected 2 fields, saw 5\nSkipping line 246752: expected 2 fields, saw
5\nSkipping line 247189: expected 2 fields, saw 5\nSkipping line 250276:
expected 2 fields, saw 5\nSkipping line 255327: expected 2 fields, saw
5\nSkipping line 257094: expected 2 fields, saw 5\n'
b'Skipping line 264626: expected 2 fields, saw 5\nSkipping line 265028: expected
2 fields, saw 5\nSkipping line 269150: expected 2 fields, saw 5\nSkipping line
271360: expected 2 fields, saw 5\nSkipping line 273975: expected 2 fields, saw
5\nSkipping line 274742: expected 2 fields, saw 5\nSkipping line 276227:
expected 2 fields, saw 5\nSkipping line 279807: expected 2 fields, saw
5\nSkipping line 283425: expected 2 fields, saw 5\nSkipping line 287468:
```

expected 2 fields, saw 5\nSkipping line 292995: expected 2 fields, saw
5\nSkipping line 293496: expected 2 fields, saw 5\nSkipping line 293735:
expected 2 fields, saw 5\nSkipping line 295060: expected 2 fields, saw
5\nSkipping line 296643: expected 2 fields, saw 5\nSkipping line 296848:
expected 2 fields, saw 5\nSkipping line 308926: expected 2 fields, saw
5\nSkipping line 310360: expected 2 fields, saw 5\nSkipping line 317004:
expected 2 fields, saw 5\nSkipping line 318207: expected 2 fields, saw
5\nSkipping line 331783: expected 2 fields, saw 5\nSkipping line 333864:
expected 2 fields, saw 5\nSkipping line 335958: expected 2 fields, saw
5\nSkipping line 336290: expected 2 fields, saw 5\nSkipping line 343526:
expected 2 fields, saw 5\nSkipping line 343857: expected 2 fields, saw
5\nSkipping line 344059: expected 2 fields, saw 5\nSkipping line 348691:
expected 2 fields, saw 5\nSkipping line 353446: expected 2 fields, saw
5\nSkipping line 357073: expected 2 fields, saw 5\nSkipping line 359753:
expected 2 fields, saw 5\nSkipping line 359974: expected 2 fields, saw
5\nSkipping line 366534: expected 2 fields, saw 5\nSkipping line 369514:
expected 2 fields, saw 5\nSkipping line 377759: expected 2 fields, saw
5\nSkipping line 379327: expected 2 fields, saw 5\nSkipping line 380769:
expected 2 fields, saw 5\nSkipping line 381073: expected 2 fields, saw
5\nSkipping line 381489: expected 2 fields, saw 5\nSkipping line 386304:
expected 2 fields, saw 5\nSkipping line 387635: expected 2 fields, saw
5\nSkipping line 389613: expected 2 fields, saw 5\nSkipping line 392604:
expected 2 fields, saw 5\nSkipping line 393184: expected 2 fields, saw
5\nSkipping line 395530: expected 2 fields, saw 5\nSkipping line 396939:
expected 2 fields, saw 5\nSkipping line 397385: expected 2 fields, saw
5\nSkipping line 397509: expected 2 fields, saw 5\nSkipping line 402902:
expected 2 fields, saw 5\nSkipping line 405187: expected 2 fields, saw
5\nSkipping line 408412: expected 2 fields, saw 5\nSkipping line 419423:
expected 2 fields, saw 5\nSkipping line 420962: expected 2 fields, saw
5\nSkipping line 425965: expected 2 fields, saw 5\nSkipping line 427496:
expected 2 fields, saw 5\nSkipping line 438881: expected 2 fields, saw
5\nSkipping line 439776: expected 2 fields, saw 5\nSkipping line 440345:
expected 2 fields, saw 5\nSkipping line 445507: expected 2 fields, saw
5\nSkipping line 445548: expected 2 fields, saw 5\nSkipping line 447184:
expected 2 fields, saw 5\nSkipping line 448603: expected 2 fields, saw
5\nSkipping line 451732: expected 2 fields, saw 5\nSkipping line 458249:
expected 2 fields, saw 5\nSkipping line 460274: expected 2 fields, saw
5\nSkipping line 467630: expected 2 fields, saw 5\nSkipping line 473961:
expected 2 fields, saw 5\nSkipping line 476281: expected 2 fields, saw
5\nSkipping line 478010: expected 2 fields, saw 5\nSkipping line 478322:
expected 2 fields, saw 5\nSkipping line 479999: expected 2 fields, saw
5\nSkipping line 480898: expected 2 fields, saw 5\nSkipping line 481688:
expected 2 fields, saw 5\nSkipping line 485193: expected 2 fields, saw
5\nSkipping line 485519: expected 2 fields, saw 5\nSkipping line 486000:
expected 2 fields, saw 5\nSkipping line 489063: expected 2 fields, saw
5\nSkipping line 494525: expected 2 fields, saw 5\nSkipping line 495009:
expected 2 fields, saw 5\nSkipping line 501954: expected 2 fields, saw
5\nSkipping line 508035: expected 2 fields, saw 5\nSkipping line 508828:

```
expected 2 fields, saw 5\nSkipping line 509833: expected 2 fields, saw
5\nSkipping line 510410: expected 2 fields, saw 5\nSkipping line 518229:
expected 2 fields, saw 5\nSkipping line 520302: expected 2 fields, saw
5\nSkipping line 520340: expected 2 fields, saw 5\n'
b'Skipping line 525174: expected 2 fields, saw 5\nSkipping line 526251: expected
2 fields, saw 5\nSkipping line 529611: expected 2 fields, saw 5\nSkipping line
531398: expected 2 fields, saw 5\nSkipping line 534146: expected 2 fields, saw
5\nSkipping line 544954: expected 2 fields, saw 5\nSkipping line 553002:
expected 2 fields, saw 5\nSkipping line 553883: expected 2 fields, saw
5\nSkipping line 553887: expected 2 fields, saw 5\nSkipping line 553915:
expected 2 fields, saw 5\nSkipping line 554172: expected 2 fields, saw
5\nSkipping line 563534: expected 2 fields, saw 5\nSkipping line 565191:
expected 2 fields, saw 5\nSkipping line 574108: expected 2 fields, saw
5\nSkipping line 574412: expected 2 fields, saw 5\nSkipping line 575985:
expected 2 fields, saw 5\nSkipping line 580091: expected 2 fields, saw
5\nSkipping line 582682: expected 2 fields, saw 5\nSkipping line 585885:
expected 2 fields, saw 5\nSkipping line 590171: expected 2 fields, saw
5\nSkipping line 591924: expected 2 fields, saw 5\nSkipping line 592515:
expected 2 fields, saw 5\nSkipping line 593888: expected 2 fields, saw
5\nSkipping line 596245: expected 2 fields, saw 5\nSkipping line 607344:
expected 2 fields, saw 5\nSkipping line 607633: expected 2 fields, saw
5\nSkipping line 610939: expected 2 fields, saw 5\nSkipping line 613638:
expected 2 fields, saw 5\nSkipping line 615643: expected 2 fields, saw
5\nSkipping line 615901: expected 2 fields, saw 5\nSkipping line 617389:
expected 2 fields, saw 5\nSkipping line 634641: expected 2 fields, saw
5\nSkipping line 635755: expected 2 fields, saw 5\nSkipping line 646243:
expected 2 fields, saw 5\nSkipping line 647165: expected 2 fields, saw
5\nSkipping line 648610: expected 2 fields, saw 5\nSkipping line 648772:
expected 2 fields, saw 5\nSkipping line 651833: expected 2 fields, saw
5\nSkipping line 653663: expected 2 fields, saw 5\nSkipping line 656233:
expected 2 fields, saw 5\nSkipping line 656694: expected 2 fields, saw
5\nSkipping line 659783: expected 2 fields, saw 5\nSkipping line 660478:
expected 2 fields, saw 5\nSkipping line 661133: expected 2 fields, saw
5\nSkipping line 661736: expected 2 fields, saw 5\nSkipping line 669827:
expected 2 fields, saw 5\n'
```

[3]:

	password	strength
0	kzde5577	1
1	kino3434	1
2	visi7k1yr	1
3	megzy123	1
4	lamborghin1	1

[4]:
```
#Check the unique value in the Strength column
data['strength'].unique()
```

[4]:
```
array([1, 2, 0])
```

[ ]:

[5]:
```
#check for missing values
data.isna().sum()
```

[5]:
```
password    1
strength    0
dtype: int64
```

[6]:
```
#check what entry has the missing value
data[data['password'].isnull()]
```

[6]:
```
       password  strength
367579      NaN         0
```

[7]:
```
#drop that entry
data.dropna(inplace=True)
```

[8]:
```
data.isna().sum()
```

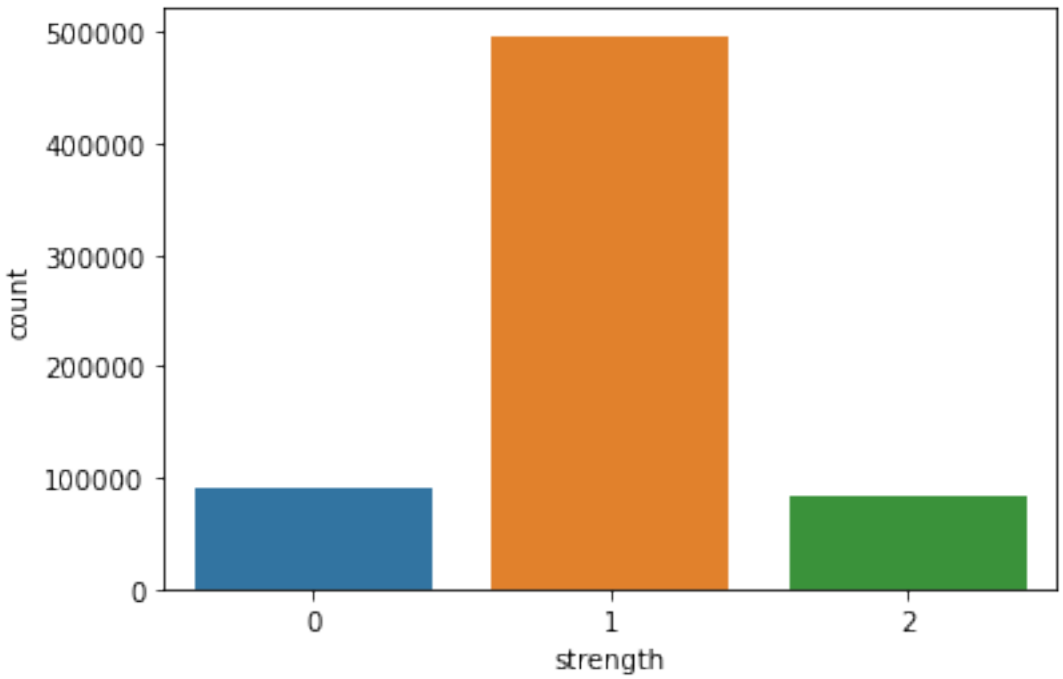
[8]:
```
password    0
strength    0
dtype: int64
```

[ ]:

[9]:
```
#visualize the different categories in my strength column
sns.countplot(data['strength'])

#we can see a huge flactuation in data between the 1's and the 0's, 2's
#if we create a ML using strength 1 it will be bias because it has already the␣
 ↪highest count (inbalance dataset)
```

[9]:
```
<AxesSubplot:xlabel='strength', ylabel='count'>
```



[10]:
```
#separate independent(password) to dependent(strength) features
#convert my entire dataset in a form of an array
password_tuple=np.array(data)
password_tuple
```

[10]:
```
array([['kzde5577', 1],
       ['kino3434', 1],
       ['visi7k1yr', 1],
       …,
       ['184520socram', 1],
       ['marken22a', 1],
       ['fxx4pw4g', 1]], dtype=object)
```

[ ]:

[11]:
```
#shuffle the data to provide robustness to the model
import random
random.shuffle(password_tuple)
```

[12]:
```
#extract dependent and independent features from the shuffled array
x=[labels[0] for labels in password_tuple]
y=[labels[1] for labels in password_tuple]
```

[ ]:

[22]:
```
#words to characters
def word_divide_char(inputs):
    character=[]
    for i in inputs:
        character.append(i)
    return character
```

[23]:
```
word_divide_char('kzde5577')
```

[23]:
```
['k', 'z', 'd', 'e', '5', '5', '7', '7']
```

[24]:
```
#import TF-IDF vectorizer to convert String data into numerical data
from sklearn.feature_extraction.text import TfidfVectorizer
```

[25]:
```
vectorizer=TfidfVectorizer(tokenizer=word_divide_char)
```

[26]:
```
#apply TF-IDF vectorizer on data
X=vectorizer.fit_transform(x)
```

[27]:
```
X.shape
```

[27]:
```
(669639, 133)
```

[28]:
```
vectorizer.get_feature_names()
```

[28]:
```
['\x05',
 '\x06',
 '\x08',
 '\x0f',
 '\x10',
 '\x11',
 '\x16',
 '\x17',
 '\x19',
 '\x1b',
 '\x1c',
 '\x1e',
 ' ',
 '!',
 '"',
 '#',
 '$',
 '%',
 '&',
 '(',
 ')',
 '*',
```

```
'+',
'-',
'.',
'/',
'0',
'1',
'2',
'3',
'4',
'5',
'6',
'7',
'8',
'9',
';',
'<',
'=',
'>',
'?',
'@',
'[',
'\\',
']',
'^',
'_',
'`',
'a',
'b',
'c',
'd',
'e',
'f',
'g',
'h',
'i',
'j',
'k',
'l',
'm',
'n',
'o',
'p',
'q',
'r',
's',
't',
'u',
```

```
'v',
'w',
'x',
'y',
'z',
'{',
'|',
'}',
'~',
'\x7f',
'\xa0',
'¡',
'¢',
'¦',
'¨',
'«',
'°',
'±',
'²',
'³',
'´',
'µ',
'¶',
'·',
'¹',
'º',
'¼',
'¾',
'¿',
'×',
'ß',
'à',
'á',
'â',
'ä',
'å',
'æ',
'ç',
'è',
'é',
'ê',
'í',
'ï',
'ð',
'ñ',
'ò',
'ó',
```

```
 'ô',
 'õ',
 'ö',
 '÷',
 'ù',
 'ú',
 'û',
 'ü',
 'ý',
 'þ',
 'ÿ',
 'œ',
 '–',
 '‚',
 '‡',
 '‹',
 '›']
```

[29]:
```
first_document_vector=X[0]
first_document_vector
```

[29]:
```
<1x133 sparse matrix of type '<class 'numpy.float64'>'
        with 6 stored elements in Compressed Sparse Row format>
```

[30]:
```
first_document_vector.T.todense()
```

[30]:
```
matrix([[0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
```

```
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.56695815],
[0.        ],
[0.59162557],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.28600303],
[0.22137588],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.29153   ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
```

```
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.33577209],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
[0.        ],
```

```
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.        ]])
```

[31]:
```
#sorting the dataframe in descending order before passing it to the machine
 ↪learning
df=pd.DataFrame(first_document_vector.T.todense(),index=vectorizer.
 ↪get_feature_names(),columns=['TF-IDF'])
df.sort_values(by=['TF-IDF'],ascending=False)
```

[31]:
```
      TF-IDF
7   0.591626
5   0.566958
z   0.335772
k   0.291530
d   0.286003
..       …
>   0.000000
=   0.000000
<   0.000000
;   0.000000
>   0.000000

[133 rows x 1 columns]
```

[32]:
```
#split data into train & test
#train---> To learn the relationship within data,
#test-->  To do predictions, and this testing data will be unseen to my model
from sklearn.model_selection import train_test_split
```

[33]:
```
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.2)
```

[34]:
```
X_train.shape
```

[34]:
```
(535711, 133)
```

[35]:
```
from sklearn.linear_model import LogisticRegression
```

[36]:
```
#Apply Logistic on data as use-case is Classification
#using multinomial because of different classes in the 'strength' column (0,1,2)
clf=LogisticRegression(random_state=0,multi_class='multinomial')
```

[37]:
```
#fit the data
clf.fit(X_train,y_train)
```

[37]:
```
LogisticRegression(multi_class='multinomial', random_state=0)
```

[ ]:

[38]:
```
#doing prediction for specific custom data
dt=np.array(['%@123abcd'])
pred=vectorizer.transform(dt) #transforming the password in some numrical data␣
 ↪for the machine learning al to understand
clf.predict(pred)
```

[38]:
```
array([1])
```

[39]:
```
#doing prediction on X-Test data
y_pred=clf.predict(X_test)
y_pred
```

[39]:
```
array([2, 1, 1, …, 1, 1, 1])
```

[ ]:

[40]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
```

[41]:
```
#check Accuracy of your model using confusion_matrix,accuracy_score
cm=confusion_matrix(y_test,y_pred)
print(cm)
print(accuracy_score(y_test,y_pred))
```

```
[[ 5363 12740    10]
 [ 3872 92826  2540]
 [   23  5241 11313]]
0.8176184218385999
```

[ ]:

[42]:
```
#create report of your model
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.58      0.30      0.39     18113
           1       0.84      0.94      0.88     99238
           2       0.82      0.68      0.74     16577

    accuracy                           0.82    133928
   macro avg       0.74      0.64      0.67    133928
weighted avg       0.80      0.82      0.80    133928
```

[ ]:
```

```