import string

string.punctuation

nltk.download('stopwords')

from nltk.corpus import stopwords

stopwords.words('english')

train\_df1['type'] = 'train'

test\_df1['type'] = 'test'

df1 = pd.concat([train\_df1,test\_df1],axis = 0)

def cleaning\_text(msg1):

## removing punctuations

punc\_removed = [char for char in msg1 if char not in string.punctuation]

punc\_removed2 = ''.join(punc\_removed)

## remove stop words

stop\_word\_removed1 = [word.lower() for word in punc\_removed2.split() if word.lower() not in stopwords.words('english')]

return stop\_word\_removed1

df1['cleaned\_text'] = df1.text.apply(cleaning\_text)

from sklearn.feature\_extraction.text import CountVectorizer

vectorizer1 = CountVectorizer(analyzer=cleaning\_text,dtype = np.uint8)

tweets1\_countvec = vectorizer1.fit\_transform(df1.text)

x = pd.DataFrame(tweets1\_countvec.toarray())

x.columns = vectorizer1.get\_feature\_names()

df2 = df1.reset\_index().merge(x,left\_index = True,right\_index= True, how = 'left')

df2.rename(columns = {

'id\_y' :'id' ,

'index\_y' : 'index',

'location\_y' : 'location',

'target\_y' : 'target',

'text\_y' : 'text',

'type\_y':'type'

},inplace = True)

model\_data\_train = df2[df2.type\_x == 'train']

model\_data\_test = df2[df2.type\_x == 'test']

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

from sklearn.naive\_bayes import MultinomialNB

nb\_cls = MultinomialNB()

train1,test1 = train\_test\_split(model\_data\_train,test\_size = 0.2)

from sklearn.metrics import classification\_report, confusion\_matrix

y\_predict\_test = nb\_cls.predict(test1[vectorizer1.get\_feature\_names()])

cm = confusion\_matrix(test1.target\_x,y\_predict\_test)

print(classification\_report(test1.target\_x,y\_predict\_test))

nb\_cls.fit(model\_data\_train[vectorizer1.get\_feature\_names()],model\_data\_train.target\_x)

y\_predict\_test = nb\_cls.predict(model\_data\_test[vectorizer1.get\_feature\_names()])

test\_df1['target'] = y\_predict\_test

test\_df1[['id','target']]