```
import pandas as pd
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df = pd.read_csv('/content/drive/MyDrive/twitter_csv')
```

```
df
```

	Unnamed: 0	Unnamed: 0.1	Date	User	Content
0	0	0.0	2022-12-30 09:23:04+00:00	virendersehwag	Poora deekhn banta hai , but from 4:25 mins is...
1	1	1.0	2022-12-30 03:37:02+00:00	virendersehwag	Wishing dear @RishabhPant17 a super speedy rec...
2	2	2.0	2022-12-29 19:37:55+00:00	virendersehwag	A Magician on the field and one of the greates...
3	3	3.0	2022-12-25 05:36:30+00:00	virendersehwag	The scientist did it. Somehow got this one. Br...
4	4	4.0	2022-12-18 17:58:27+00:00	virendersehwag	One of the greatest World Cup games of all tim...
...	...	...	...	...	...
39781	39781	NaN	2018-01-02 11:15:30+00:00	TechnicalGuruji	What should WE AIM for in 2018? Another GOLD? ...
39782	39782	NaN	2018-01-02 09:37:42+00:00	TechnicalGuruji	Login with Facebook? Login with Google? OAuth?...

```
import re
```

```
df['Content']
```

```
0        Poora deekhn banta hai , but from 4:25 mins is...
1        Wishing dear @RishabhPant17 a super speedy rec...
2        A Magician on the field and one of the greates...
3        The scientist did it. Somehow got this one. Br...
4        One of the greatest World Cup games of all tim...
                              ...
39781    What should WE AIM for in 2018? Another GOLD? ...
39782    Login with Facebook? Login with Google? OAuth?...
39783          कार्य प्रगति पे है। https://t.co/3SsnmXLtSa
39784    Tech Predictions for 2018 - Good and Bad? http...
```

```
39785    Happy New Year - Light Up 2018 Dubai - World R...
Name: Content, Length: 39786, dtype: object
```

```
def remove_emoji(string):
    emoji_pattern = re.compile("["
                           u"\U0001F600-\U0001F64F"
                           u"\U0001F300-\U0001F5FF"
                           u"\U0001F680-\U0001F6FF"
                           u"\U0001F1E0-\U0001F1FF"
                           u"\U00002702-\U000027B0"
                           u"\U000024C2-\U0001F251"
                           "]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', string)
```

```
df['Content'] = df['Content'].apply(remove_emoji)
```

```
def remove_urls (tweet):
    tweet = re.sub(r'(https|http)?:\/\/(\w|\.|\/|\?|\=|\&|\%)*\b', '', tweet, flags=re.MULTIL
    return(tweet)
```

```
df['Content'] = df['Content'].apply(remove_urls)
```

```
df['Content']
```

```
0        Poora deekhn banta hai , but from 4:25 mins is...
1        Wishing dear @RishabhPant17 a super speedy rec...
2        A Magician on the field and one of the greates...
3        The scientist did it. Somehow got this one. Br...
4        One of the greatest World Cup games of all tim...
                              ...
39781       What should WE AIM for in 2018? Another GOLD?
39782    Login with Facebook? Login with Google? OAuth?...
39783                                  कार्य प्रगति पे है।
39784    Tech Predictions for 2018 - Good and Bad?  via...
39785    Happy New Year - Light Up 2018 Dubai - World R...
Name: Content, Length: 39786, dtype: object
```

```
def get_tweet(tweets):
    tweet = re.sub(r"[-()\"#/@;:<>{}`+=~|.!?,]", "", tweets)
    return tweet
```

```
df['Content'] = df['Content'].apply(get_tweet)
```

```
df['Content']
```

```
0        Poora deekhn banta hai  but from 425 mins is v...
1        Wishing dear RishabhPant17 a super speedy reco...
```

```
2        A Magician on the field and one of the greates...
3        The scientist did it Somehow got this one Bril...
4        One of the greatest World Cup games of all tim...
                              ...
39781         What should WE AIM for in 2018 Another GOLD
39782    Login with Facebook Login with Google OAuth Sa...
39783                                 कार्य प्रगति पे है।
39784    Tech Predictions for 2018  Good and Bad  via Y...
39785    Happy New Year  Light Up 2018 Dubai  World Rec...
Name: Content, Length: 39786, dtype: object
```

```
df['Content'][10000]
```

```
'Agra police raid spa center arrest foreign girls for prostitution  squreshiagra\n\nRea
d full story  '
```

```
def remove_num(tweets):
  tweet=re.sub("(\s\d+)","",tweets)
  return tweet
```

```
df['Content'] = df['Content'].apply(remove_num)
```

```
lines = []
for i in df['Content']:
    lines.append(i)
print(lines[-1])
```

```
 Happy New Year  Light Up Dubai  World Record Celebrations  via YouTubeIndia
```

```
data = ""

for i in lines:
    data = ' '. join(lines)

data = data.replace('\n', '').replace('\r', '').replace('\ufeff', '')
```

```
z = []

for i in data.split():
    if i not in z:
        z.append(i)

data = ' '.join(z)
data[:500]
```

```
'Poora deekhn banta hai but from mins is very special stuff May you heal and get well s
```

```
from tensorflow.keras.preprocessing.text import Tokenizer
import pickle
import numpy as np


tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts([data])

pickle.dump(tokenizer, open('tokenizer1.pkl', 'wb'))

sequence_data = tokenizer.texts_to_sequences([data])[0]
sequence_data[:10]
```

```
[1441, 8740, 1442, 408, 84, 35, 1443, 11, 1444, 409]
```

```
print(tokenizer.index_word)
```

```
{1: 'india', 2: 'the', 3: 'amp', 4: 'in', 5: 's', 6: 'i', 7: 'a', 8: 'm', 9: 'its', 10:
```

```
sequence = []
for i in range(3,len(sequence_data)):
  words = sequence_data[i-3:i+1]
  sequence.append(words)
print(len(sequence))
sequence = np.array(sequence)
sequence[:10]
```

```
21085
array([[1441, 8740, 1442,  408],
       [8740, 1442,  408,   84],
       [1442,  408,   84,   35],
       [ 408,   84,   35, 1443],
       [  84,   35, 1443,   11],
       [  35, 1443,   11, 1444],
       [1443,   11, 1444,  409],
       [  11, 1444,  409, 1445],
       [1444,  409, 1445, 1446],
       [ 409, 1445, 1446,  166]])
```

```
X = []
y = []

for i in sequence:
  X.append(i[:3])
  y.append(i[3])

X = np.array(X)
y = np.array(y)


print(X[:10])
print(y[:10])
```

```
[[1441 8740 1442]
 [8740 1442  408]
 [1442  408   84]
 [ 408   84   35]
 [  84   35 1443]
 [  35 1443   11]
 [1443   11 1444]
 [  11 1444  409]
 [1444  409 1445]
 [ 409 1445 1446]]
[ 408   84   35 1443   11 1444  409 1445 1446  166]
```

```
from tensorflow.keras.utils import to_categorical


y = to_categorical(y,num_classes=10000)


y[:5]
```

```
array([[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.]], dtype=float32)
```

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.layers import Embedding, LSTM, Dense
from tensorflow.keras.models import Sequential


model = Sequential()
model.add(Embedding(10000,64,input_length=3))
model.add(LSTM(1000, return_sequences=True))
model.add(LSTM(1000))
model.add(Dense(1000, activation="relu"))
model.add(Dense(10000, activation="softmax"))
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 3, 64)             640000

 lstm (LSTM)                 (None, 3, 1000)           4260000

 lstm_1 (LSTM)               (None, 1000)              8004000

 dense (Dense)               (None, 1000)              1001000

 dense_1 (Dense)             (None, 10000)             10010000

=================================================================
Total params: 23,915,000
Trainable params: 23,915,000
Non-trainable params: 0
_________________________________________________________________
```

```
model.compile(loss='categorical_crossentropy')
```

```
model.fit(X,y,epochs=100,batch_size=128)
```

```
Epoch 1/100
165/165 [==============================] - 15s 35ms/step - loss: 9.2132
Epoch 2/100
165/165 [==============================] - 3s 21ms/step - loss: 9.2103
Epoch 3/100
165/165 [==============================] - 4s 22ms/step - loss: 9.2075
Epoch 4/100
165/165 [==============================] - 3s 21ms/step - loss: 9.2041
Epoch 5/100
165/165 [==============================] - 3s 21ms/step - loss: 9.1981
Epoch 6/100
165/165 [==============================] - 4s 21ms/step - loss: 9.1917
Epoch 7/100
165/165 [==============================] - 3s 20ms/step - loss: 9.1867
Epoch 8/100
165/165 [==============================] - 3s 21ms/step - loss: 9.1833
Epoch 9/100
165/165 [==============================] - 3s 21ms/step - loss: 9.1806
Epoch 10/100
165/165 [==============================] - 3s 20ms/step - loss: 9.1790
Epoch 11/100
165/165 [==============================] - 4s 21ms/step - loss: 9.1773
Epoch 12/100
165/165 [==============================] - 3s 21ms/step - loss: 9.1753
Epoch 13/100
165/165 [==============================] - 3s 20ms/step - loss: 9.1747
Epoch 14/100
165/165 [==============================] - 3s 21ms/step - loss: 9.1733
```

```
Epoch 15/100
165/165 [==============================] - 3s 21ms/step - loss: 9.1718
Epoch 16/100
165/165 [==============================] - 4s 22ms/step - loss: 9.1641
Epoch 17/100
165/165 [==============================] - 3s 20ms/step - loss: 9.1195
Epoch 18/100
165/165 [==============================] - 3s 20ms/step - loss: 9.0773
Epoch 19/100
165/165 [==============================] - 4s 22ms/step - loss: 9.0629
Epoch 20/100
165/165 [==============================] - 3s 20ms/step - loss: 9.0621
Epoch 21/100
165/165 [==============================] - 3s 20ms/step - loss: 9.0476
Epoch 22/100
165/165 [==============================] - 3s 20ms/step - loss: 9.0390
Epoch 23/100
165/165 [==============================] - 4s 22ms/step - loss: 9.0320
Epoch 24/100
165/165 [==============================] - 4s 22ms/step - loss: 9.0199
Epoch 25/100
165/165 [==============================] - 3s 20ms/step - loss: 8.9995
Epoch 26/100
165/165 [==============================] - 3s 20ms/step - loss: 8.9525
Epoch 27/100
165/165 [==============================] - 3s 21ms/step - loss: 8.8936
Epoch 28/100
165/165 [==============================] - 3s 21ms/step - loss: 8.8407
Epoch 29/100
165/165 [==============================] - 3s 20ms/step - loss: 8.7923
```

```
model.save('next_word.h5')
```