Date:17 Nov 2022

Team Id:PNT2022TMID36441

Project Name: Digital Naturalist-

Assignment-4

Download the Dataset :

```
dataCsv = "/content/spam.csv"
```

Importing The Required Libraries:

```
import pandas as pd
import nltk
import re
import numpy as np
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.translate.ribes_score import word_rank_alignment
from numpy.lib.shape_base import split
from sklearn import preprocessing
from sklearn.feature_extraction.text import CountVectorizer
from tensorflow.keras.models import Sequential
from sklearn.model_selection import train_test_split
from keras.layers import LSTM,Dense,Dropout,Input,Embedding,Activation,Flatten
from keras.models import Model
import nltk
```

Data Reading And Preprocessing :

```
data = pd.read_csv(dataCsv,encoding = "ISO-8859-1")
```

```
data.drop(["Unnamed: 2","Unnamed: 3","Unnamed: 4"],axis = 1,inplace = True)
data.head()
```

	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...

```
nltk.download('stopwords',quiet=True)
nltk.download('all',quiet=True)
```

```
True
```

```
p=PorterStemmer()
input = []
```

```
for i in range(0,5572):
  v2 = data['v2'][i]

  #removing punctuation
  v2 = re.sub('[^a-zA-Z]',' ',v2)

  #converting to lower case
  v2 = v2.lower()

  #splitting the sentence
  v2 = v2.split()

  #removing the stopwords and stemming
  v2 = [p.stem(word) for word in v2 if not word in set(stopwords.words('english'))]

  v2 = ' '.join(v2)

  input.append(v2)
```

```
#creating document term matrix
cv = CountVectorizer(max_features=2000)
x = cv.fit_transform(input).toarray()
x.shape
```

```
(5572, 2000)
```

```
le = preprocessing.LabelEncoder()
data['v1'] = le.fit_transform(data['v1'])
data['v1'].unique()
```

```
array([0, 1])
```

```
y = data['v1'].values
```

```
y = y.reshape(-1,1)
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.4)
```

Create Model And Adding Layers:

```
model = Sequential()
model.add(Dense(1565,activation = "relu"))
model.add(Dense(3000,activation = "relu"))
model.add(Dense(1,activation = "sigmoid"))
model.add(Flatten())
```

Compile Fit And Save The Model:

```
model.compile(optimizer = "adam",loss = "binary_crossentropy", metrics = ["accuracy"])
model.fit(x_train,y_train,epochs = 15)
model.save("spam-message-classifier.h5")
```

```
Epoch 1/15
105/105 [==============================] - 5s 40ms/step - loss: 0.1318 - accuracy: 0.962
Epoch 2/15
105/105 [==============================] - 4s 40ms/step - loss: 0.0121 - accuracy: 0.997
Epoch 3/15
105/105 [==============================] - 4s 40ms/step - loss: 0.0028 - accuracy: 0.999
Epoch 4/15
105/105 [==============================] - 4s 39ms/step - loss: 0.0022 - accuracy: 0.999
Epoch 5/15
105/105 [==============================] - 4s 39ms/step - loss: 0.0019 - accuracy: 0.999
Epoch 6/15
105/105 [==============================] - 4s 40ms/step - loss: 0.0019 - accuracy: 0.999
Epoch 7/15
105/105 [==============================] - 4s 39ms/step - loss: 0.0017 - accuracy: 0.999
Epoch 8/15
105/105 [==============================] - 5s 47ms/step - loss: 0.0016 - accuracy: 0.999
Epoch 9/15
105/105 [==============================] - 4s 40ms/step - loss: 0.0016 - accuracy: 0.999
Epoch 10/15
105/105 [==============================] - 5s 50ms/step - loss: 0.0016 - accuracy: 0.999
Epoch 11/15
105/105 [==============================] - 5s 46ms/step - loss: 0.0017 - accuracy: 0.999
Epoch 12/15
```

```
105/105 [==============================] - 5s 51ms/step - loss: 0.0016 - accuracy: 0.999
Epoch 13/15
105/105 [==============================] - 6s 53ms/step - loss: 0.0015 - accuracy: 0.999
Epoch 14/15
105/105 [==============================] - 4s 41ms/step - loss: 0.0016 - accuracy: 0.999
Epoch 15/15
105/105 [==============================] - 4s 42ms/step - loss: 0.0015 - accuracy: 0.999
```

Testing The Model:

```
test_sentence= "Explore%%The**Features 9in ++Natures"
message = re.sub('[^a-zA-Z]',' ',test_sentence)
message
```

```
'Explore  The  Features  in   Natures'
```

Double-click (or enter) to edit

Double-click (or enter) to edit