

ASSIGNMENT 2

Assignment Date	06 November 2022
Student Name	P.Aswini
Student Roll Number	715319106004
Maximum Marks	2 Marks

Question-1. Import required library

Solution:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from keras.models import Model

from keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding

from keras.optimizers import Adam

from keras.preprocessing.text import Tokenizer

from keras.preprocessing import sequence

from keras.utils import pad_sequences

from keras.utils import to_categorical

from keras.callbacks import EarlyStopping
```



Question-2. Read the Dataset

Solution:

```
df = pd.read_csv('spam.csv',delimiter=',',encoding='latin-1')  
df.head()
```

Question-3. Preprocessing the Dataset

Solution:

```
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],axis=1,inplace=True)  
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator  
X = df.v2  
Y = df.v  
le =LabelEncoder()  
Y =le.fit_transform(Y)  
Y = Y.reshape(-1,1)
```

```

X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.25)

max_words = 1000

max_len = 150

tok = Tokenizer(num_words=max_words,fit_on_texts(X_train))
word_sequences = tok.texts_to_sequences(X_train)

sequences_matrix = pad_sequences(
    (word_sequences,maxlen=max_len)
)

```

Question :4

Solution:

```

inputs = Input(shape=[max_len])

layer = Embedding(max_words,50,input_length=max_len)(inputs)

layer = LSTM(128)(layer)

layer = Dense(128)(layer)

layer = Activation('relu')(layer)

layer = Dropout(0.5)(layer)

layer = Dense(1)(layer)

layer = Activation('sigmoid')(layer)

model = Model(inputs=inputs,outputs=layer)

```

Question-7. Fit the Model

Solution:

```

history = model.fit(sequences_matrix,Y_train,batch_size=20,epochs=15,
    validation_split=0.2)

```

```

metrics = pd.DataFrame(history.history)

metrics.rename(columns ={'loss':'Training_Loss', 'accuracy':'Training_Accuracy', 'val_loss':
'Validation_Loss', 'val_accuracy': 'Validation_Accuracy'}, inplace = True)

def plot_graphs1(var1, var2, string):

metrics[[var1, var2]].plot()

plt.title('Training and Validation ' + string)

plt.xlabel ('Number of epochs')

plt.ylabel(string)

plt.legend([var1, var2])

plot_graphs1('Training_Accuracy', 'Validation_Accuracy', 'Accuracy')

```

Question-8.Save The Model

Solution:

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

Question-9. Test The Model

Solution:

```

test_sequences = tok.texts_to_sequences(X_test)

test_sequences_matrix = pad_sequences(test_sequences,maxlen=max_len)

accuracy1 = model.evaluate(test_sequences_matrix,Y_test)

```