**ASSIGNMENT- 4**

|  |  |
| --- | --- |
| **Register Number** | **921319104212** |
| **Date** | **15th October ,2022** |
| **Student Name** | **K. Soundarya** |
| **Maximum Marks** | **2 Marks** |

# **#Import required libraries**

import pandas as pd import seaborn as sns

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 RMSprop

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

# **#Read dataset and do pre-processing**

df = pd.read\_csv( 'sample\_data/spam.csv', delimiter = ',', encoding = 'latin-1'

)

df.drop(

['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],

axis = 1, inplace = True

)

X = df.v2 Y = df.v1

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 = 0.15

)

max\_words = 1000

max\_len = 150

tok = Tokenizer(num\_words = max\_words) tok.fit\_on\_texts(X\_train)

sequences = tok.texts\_to\_sequences(X\_train)

sequences\_matrix = pad\_sequences(sequences, maxlen = max\_len)

# **#Add Layers (LSTM, Dense-(Hidden Layers), Output)**

**def** RNN():

inputs = Input(name = 'inputs', shape = [max\_len])

layer = Embedding(max\_words, 50, input\_length = max\_len)(inputs)

layer = LSTM(64)(layer)

layer = Dense(256,name = 'FC1')(layer) layer = Activation('relu')(layer) layer = Dropout(0.5)(layer)

layer = Dense(1,name = 'out\_layer')(layer) layer = Activation('sigmoid')(layer)

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

**return** model #Create model = RNN() #Compile the model

model.compile(loss = 'binary\_crossentropy', optimizer = RMSprop(), metrics = ['accuracy'])

# **#Fit the model**

model.fit(

sequences\_matrix, Y\_train, batch\_size = 128, epochs=10,

validation\_split = 0.2,

callbacks=[EarlyStopping(monitor = 'val\_loss', min\_delta = 0.0001)])

Epoch 1/10

30/30 [==============================] - 12s 296ms/step - loss: 0.3357

* accuracy: 0.8672 - val\_loss: 0.1516 - val\_accuracy: 0.9757 Epoch 2/10

30/30 [==============================] - 8s 272ms/step - loss: 0.0924

* accuracy: 0.9762 - val\_loss: 0.0530 - val\_accuracy: 0.9873

<keras.callbacks.History at 0x7fdb374d5250>

#Save the model model.save('./spam.h5') #Test the model

test\_sequences = tok.texts\_to\_sequences(X\_test) test\_sequences\_matrix = pad\_sequences(test\_sequences, maxlen = max\_len)

accr = model.evaluate(test\_sequences\_matrix, Y\_test) print('Test set\n Loss: {:0.3f}\n Accuracy:

{:0.3f}'.format(accr[0],accr[1]))

27/27 [==============================] - 1s 26ms/step - loss: 0.0550 -

accuracy: 0.9880

Test set Loss: 0.055

Accuracy: 0.9