# Assignment-Ⅳ

**AI-POWERED NUTRITION ANALYSER FOR FITNESS ENTHUSIAST**

|  |  |
| --- | --- |
| Date | 15 November 2022 |
| Student name | Nallapaneni Bala Thripura sundari |
| Team ID | PNT2022TMID36299 |
| Maximum marks | 2 marks |

importpandasaspdimportnumpyasnpimport matplotlib.pyplotaspltimportseabornassnsfrom sklearn.model\_selectionimporttrain\_test\_split fromsklearn.preprocessingimportLabelEncoderfrom tensorflow.keras.modelsimportModel

fromtensorflow.keras.layersimportLSTM,Activation,Dense,Dropout,Input,

Embedding

fromtensorflow.keras.optimizersimportRMSpropfrom tensorflow.keras.preprocessing.textimportTokenizer fromtensorflow.keras.preprocessingimportsequencefrom tensorflow.keras.utilsimportto\_categoricalfrom tensorflow.keras.callbacksimportEarlyStopping

%matplotlibinlineimport csv

**with**open('/spam.csv','r')ascsvfile: reader

=csv.reader(csvfile)df= pd.read\_csv(r'/spam.csv',encoding='latin-1') df.head()

v1 v2Unnamed:2 \0

ham Gountiljurongpoint,crazy..Availableonly... NaN

1. ham Oklar...Jokingwifuoni... NaN
2. spam Freeentryin2awklycomptowinFACupfina... NaN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | ham  ham | Udunsaysoearlyhor...Ucalreadythen  NahIdon'tthinkhegoestousf,helives | | say...  aro... | NaN  NaN | 4 |
|  | Unnamed:3 | | Unnamed:4 | | | |
| 0 | NaN | | NaN | | | |
| 1 | NaN | | NaN | | | |
| 2 | NaN | | NaN | | | |

1. NaN NaN4 NaN NaNdf.drop(['Unnamed:2', 'Unnamed:3','Unnamed:4'],axis=1,inplace=True)df.info()

<class'pandas.core.frame.DataFrame'> RangeIndex:5572entries,0to5571 Datacolumns(total2columns):

# Column Non-NullCount Dtype

-

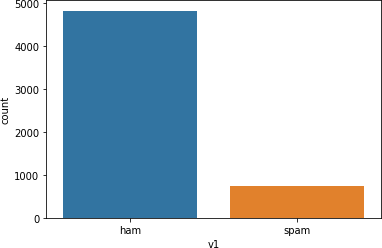
0 v1 5572non-null object

1 v2 5572non-null objectdtypes:object(2)memoryusage:

87.2+KBsns.countplot(df.v1)

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning:Passthefollowingvariableasakeywordarg:x.Fromversion 0.12,theonlyvalidpositionalargumentwillbe`data`,andpassingother arguments without an explicit keyword will result in an error or misinterpretation. FutureWarning

<matplotlib.axes.\_subplots.AxesSubplotat0x7f5197dac250>



X=df.v2Y=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=0.20)

max\_words=1000max\_len

=150

tok=Tokenizer(num\_words=max\_words)tok.fit\_on\_texts(X\_train) sequences=tok.texts\_to\_sequences(X\_train)sequences\_matrix= sequence.pad\_sequences(sequences,maxlen=max\_len)

**def**RNN():

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

layer=Embedding(max\_words,50,input\_length=max\_len)(inputs) layer=LSTM(128)(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('tanh')(layer) model= Model(inputs=inputs,outputs=layer) **return**model

model=RNN()model.summary() model.compile(loss='binary\_crossentropy',optimizer=RMSprop(),metrics=['accura cy','mse','mae'])

Model:"model"

Layer(type) OutputShape Param#

=================================================================

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| inputs(InputLayer) | [(None,150)] | | | 0 | |
| embedding(Embedding) | (None,150,50) | | | 50000 | |
| lstm(LSTM) FC1(Dense) | (None, (None,256) | | 128) | 33024 | 91648 |
| activation(Activation) | (None, | 256) | 0 | | |
| dropout(Dropout) | (None, | 256) | 0 | | |

out\_layer(Dense) (None,1) 257

activation\_1(Activation) (None,1) 0

=================================================================

Totalparams:174,929

Trainableparams:174,929

Non-trainableparams:0

model.fit(sequences\_matrix,Y\_train,batch\_size=128,epochs=10,

validation\_split=0.2,callbacks=[EarlyStopping(monitor='val\_loss',min\_delta=0. 0001)])

Epoch1/10

28/28[==============================]-17s486ms/step-loss:0.2960-

accuracy:0.8819-mse:0.0821-mae:0.1563-val\_loss:0.1341- val\_accuracy:0.9675-val\_mse:0.0344-val\_mae:0.1237Epoch2/10 28/28[==============================]-13s462ms/step-loss:0.1149-

accuracy:0.9764-mse:0.0381-mae:0.1538-val\_loss:0.1321-

val\_accuracy:0.9798-val\_mse:0.0437-val\_mae:0.1695

<keras.callbacks.Historyat0x7f5193192590>

test\_sequences=tok.texts\_to\_sequences(X\_test)test\_sequences\_matrix= sequence.pad\_sequences(test\_sequences,maxlen=max\_len)accr= model.evaluate(test\_sequences\_matrix,Y\_test)

35/35[==============================]-3s78ms/step-loss:0.1590-

accuracy:0.9812-mse:0.0451-mae:0.1733

print('Testset\n Loss:{:0.3f}\n Accuracy:

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

Testset

Loss:0.159 Accuracy:0.981 model.save("./assign4model.h5")

fromtensorflow.keras.modelsimportload\_modelm2

=load\_model("./assign4model.h5") m2.evaluate(test\_sequences\_matrix,Y\_test)

35/35[==============================]-3s68ms/step-loss:0.1590-

accuracy:0.9812-mse:0.0451-mae:0.1733

[0.1589982509613037,

0.9811659455299377,

0.04506031796336174,

0.17333826422691345]