Assignment Date	29-10-2022
Student Name	KAVITHASRI R
Student Roll Number	921819104016
Maximum Marks	2 MARKS

Question 1:

Import the necessary libraries

Solution:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
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
```

Question2:

Download the Dataset

Solution:

Dataset Downloaded and uploaded to drive https://www.kaggle.com/code/kredy10/simple-lstm-for-textclassification/data

Question3:

Read dataset and do pre-processing

Solution:

Read dataset

```
In [21]:  df = pd.read_csv('/content/drive/MyDrive/spam.csv',delimiter=',',encoding='latin-1')
          df.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

Pre-processing the Dataset

```
In [22]:
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],axis=1,inplace=True)
df.info()

RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   v1      5572 non-null   object
 1   v2      5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

```
In [23]:
X = df.v2
Y = df.v1
le = LabelEncoder()
Y = le.fit_transform(Y)
Y = Y.reshape(-1,1)
```

```
In [24]:
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.15)
```

```
In [25]:
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)
```

```
In [26]:
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)

model.summary()
```

Hoael. -mdel 1'

LCyer (type)	Output Shape	Paraa 0
inputs (1nputLayer)	[(None, 150)]	0
embedding_1 (Embedding)	(None, 150, 50)	50000
Lstn l (LSTH)	(None, 64)	29440
FC1 (Dense)	(scene, 2S6)	16668
acttvacxon_2 (Act*vatton)	(lime, 256)	0
dropout_1 (Dropout)	(Mone, 256)	0
out_layer (Dense)	(f¥one, 1)	D57
activation_3 (Activation)	(None, 1)	0

T e1 paraas : 96,337
Tra1neb1e paras: 96,337
Nan-tra1nab1e paraas: 8

```
mdaz.caqi11e(loss•'b1nary crossantropy',egtlatzar•9tbprog( ),aetrtcs•( 'accuracy' ])
```

```
sequences_matrix,Y_train,batch_size=128,epochs=10,
validation_split=0.2)
```

Epoch 1/16
3a/3a [=——— -] - 10s 264ms/step - loss: 0.3182 - accuracy: 0.8788 - val_loss: 0.1571 - val_accuracy: 0.9715
£pech 7/18
30/30 [=——————) - 7s 2'+7as/step - Less i 8.0805 - ac‹urscy: B. 97¥£ - val_\#sr: 9.B7€2 - val accuracy: 9.9778
rpoch 1/18
30/30 [=- ———————-j • 7s I37ssf rtep - loss: 6.4443 - ac‹ursry: 9.9881 • val lot i: B.8679 • vat accuracy: 8.9gtl
30/30 [=- ==- --———-j 7s 24Susf xtep • loss: 8.8171 ac en rery. 9,9929 val loss. 8. B486 *sha tcurecy . 8. 9778
30/30 [*••••••••••••-   •••• •••••] - 7s 242ms/step - loss: 0.0220 - accuracy: 0.9937 - val_loss: 0.0820 - val_accuracy: 0.9800
78/M {•••••••••••••••••••••••••] - 7s 24M6/8ttp • loss' e,0178 - ¥‹tur¥cy. 8.r9ss • vol tons. e.e887 - ¥el_a¢¢urscy. e.9789
Cpo¢h 7/je
3a⁄0 [•••••••••••••••••••••••••] - 7s Z43ss/sttp • loss 8.01S8 - eccvrscy. a.9sSa - ve1_\ass. 8.e969 - vel_accurecy: e.9566
36/30 {•-——— ==- ==] - 7s 241ms/step - loss: 0.0162 - accuracy: 0.9958 - val_loss: 0.0901 - val_accuracy: 0.9768
3¥/39 {•——••——— ==] - 7s 246ms/step - loss: 0.0099 - accuracy: 0.9968 - val_loss: 0.1284 - val_accuracy: 0.9789
Egoch IB/tg
30/30 [————————————) - 7s 247•s/sttg - loss: 8.83S5 - accuracy' 8.'—fiñ - va1 loss: e.1?6't - vat accuracy: 8.97f6

Save The Model

```
ciodcl.sive ( 'sn5 t lass ifi cr. hS ' I
```

ipc ngthaTenom

```
test_sequences • tok.texts_to_sequences(X_test I
test_sequences_atrix • pa6_sequences(test_seqvences,aax1en•m4x_len)
```

Testing the Medcl

```
acer • aode1.eveluete'test_sequentes_aetrix,Y_test )
```

```
27/27 [••••••••••••  ••••••••••••••••] • 1s 7&re/step • To:s: 0.0886 • 0ccurecy: 6.9821
```

```
print('Test :et ,n  ros•: ':8.3T}'n  ¢ccvrac\ .  :9.?f' '.fomat(eccr(0),accr[I ] ))
```

```
Test :et
  toi$: 0.689
  4«uraty : 0.982
```