ASSIGNMENT – 4

PROBLEM STATEMENT :-SMS SPAM CLASSIFICATION

Assignment Date	25-OCTOBER-2022
Student's Name	KAVIYA A
Student's Roll Number	510919106701
Maximum Marks	2 Marks

QUESTION -1:

DOWNLOAD THE DATASET.

In [ ]:

```
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 to_categorical
from keras.callbacks import EarlyStopping
%matplotlib inline
```

QUESTION -2:

IMPORT REQUIRED LIBRARY

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

Out[ ]:

	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

QUESTION – 3:-

READ DATASET & DO PRE-PROCESSING

In [ ]:
```
df
```

Out[ ]:

	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
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

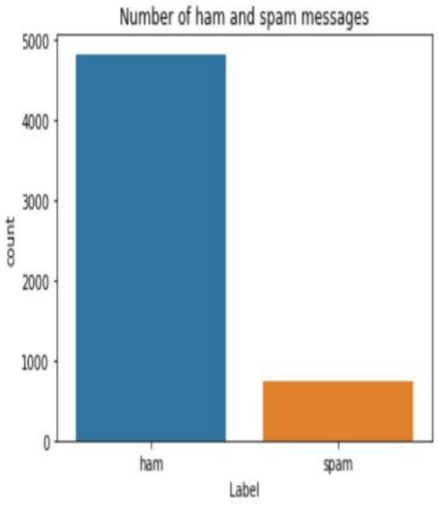
5572 rows × 5 columns

QUESTION -4:- CREATE DATASET

In [ ]:
```
sns.countplot(df.v1)
plt.xlabel('Label')
plt.title('Number of ham and spam messages')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretatio
n.
  FutureWarning
```

Out[ ]:
```
Text(0.5, 1.0, 'Number of ham and spam messages')
```



QUESTION -5:- ADD LAYERS (LSTM,DENSE-(HIDDEN LAYERS),OUTPUT)

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

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

In [ ]:
```
import tensorflow as tf
```

In [ ]:
```
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 = tf.keras.preprocessing.sequence.pad_sequences(sequences,maxlen=max_len)
```

In [ ]:
```
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
```

QUESTION – 6:-

COMPILE THE MODEL

In [ ]:

```
model = RNN()
model.summary()
model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inputs (InputLayer)         [(None, 150)]             0

 embedding (Embedding)       (None, 150, 50)           50000

 lstm (LSTM)                 (None, 64)                29440

 FC1 (Dense)                 (None, 256)               16640

 activation (Activation)     (None, 256)               0

 dropout (Dropout)           (None, 256)               0

 out_layer (Dense)           (None, 1)                 257

 activation_1 (Activation)   (None, 1)                 0

=================================================================
Total params: 96,337
Trainable params: 96,337
Non-trainable params: 0
```

QUESTION – 7:-

FIT THE MODEL

In [ ]:

```
model.fit(sequences_matrix,Y_train,batch_size=128,epochs=20,
          validation_split=0.2,callbacks=[EarlyStopping(monitor='val_loss',min_delta=0.0001)])
```

```
Epoch 1/20
30/30 [==============================] - 11s 380ms/step - loss: 0.0184 - accuracy: 0.9966 - val_loss: 0.0756 - val_accuracy: 0.9810
Epoch 2/20
30/30 [==============================] - 9s 286ms/step - loss: 0.0165 - accuracy: 0.9947 - val_loss: 0.0791 - val_accuracy: 0.9842
```

Out[ ]:

```
<keras.callbacks.History at 0x7f2b6ec39610>
```

QUESTION -8:-

SAVE THE MODEL

In [26]:
```
model.save('Spam.h5')
```

In [28]:
```
test_sequences = tok.texts_to_sequences(X_test)
test_sequences_matrix = tf.keras.preprocessing.sequence.pad_sequences(test_sequences,maxlen=max_len)
```

QUESTION – 9:-

TEST THE MODEL

In [29]:
```
accr = model.evaluate(test_sequences_matrix,Y_test)
```

```
27/27 [==============================] - 1s 23ms/step - loss: 0.0960 - accuracy: 0.9809
```

In [30]:
```
print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))
```

```
Test set
  Loss: 0.096
  Accuracy: 0.981
```