Import the libraries

```
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
```

Read dataset and do pre-processing

Read dataset

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

Saved successfully! ×

		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

Preprocessing the Dataset

```
ag.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
 ---  ------      --------------  -----
  0   v1          5572 non-null   object
  1   v2          5572 non-null   object
  2   Unnamed: 2  50 non-null     object
  3   Unnamed: 3  12 non-null     object
  4   Unnamed: 4  6 non-null      object
 dtypes: object(5)
 memory usage: 217.8+ KB
```

```
X = ag.v2
Y = ag.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)
```

Saved successfully!

Dense-(Hidden Layers), Output)

```
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()
```

```
 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
 _________________________________________________________________
```

Compile the Model

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

Train and Fit the Model

```
model.fit(sequences_matrix, Y_train,batch_size=128,epochs=10,validation_split=0.2)
```

Saved successfully!

```
Epoch 1/10
30/30 [==============================] - 8s 31ms/step - loss: 0.3214 - accuracy: 0.8812
                               =======] - 0s 13ms/step - loss: 0.0801 - accuracy: 0.9826
30/30 [==============================] - 0s 14ms/step - loss: 0.0440 - accuracy: 0.9889
Epoch 4/10
30/30 [==============================] - 0s 14ms/step - loss: 0.0340 - accuracy: 0.9908
Epoch 5/10
30/30 [==============================] - 0s 14ms/step - loss: 0.0253 - accuracy: 0.9929
Epoch 6/10
30/30 [==============================] - 0s 14ms/step - loss: 0.0196 - accuracy: 0.9950
Epoch 7/10
30/30 [==============================] - 0s 14ms/step - loss: 0.0140 - accuracy: 0.9960
Epoch 8/10
30/30 [==============================] - 0s 14ms/step - loss: 0.0126 - accuracy: 0.9963
Epoch 9/10
30/30 [==============================] - 0s 14ms/step - loss: 0.0086 - accuracy: 0.9979
Epoch 10/10
30/30 [==============================] - 0s 14ms/step - loss: 0.0063 - accuracy: 0.9987
<keras.callbacks.History at 0x7fe7105341d0>
```

Save The Model

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

Preprocessing the Test Dataset

```
test_sequences = tok.texts_to_sequences(X_test)
test_sequences_matrix = pad_sequences(test_sequences,maxlen=max_len)
```

Testing the Model

```
accr = model.evaluate(test_sequences_matrix,Y_test)
```

```
27/27 [==============================] - 0s 6ms/step - loss: 0.0478 - accuracy: 0.9892
```

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

```
Test set
 Loss: 0.048
 Accuracy: 0.989
```

Saved successfully! ×

✓ 0s completed at 9:22 PM

Saved successfully!