```
# Import required lib

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
import tensorflow
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Read csv file
df = pd.read_csv(r"C:\Users\prasa\OneDrive\Desktop\nalaiyathiran\
spam.csv", encoding="ISO-8859-1")

df.head(10)
```

```
     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
5  spam  FreeMsg Hey there darling it's been 3 week's n...        NaN
NaN        NaN
6   ham  Even my brother is not like to speak with me. ...        NaN
NaN        NaN
7   ham  As per your request 'Melle Melle (Oru Minnamin...        NaN
NaN        NaN
8  spam  WINNER!! As a valued network customer you have...        NaN
NaN        NaN
9  spam  Had your mobile 11 months or more? U R entitle...        NaN
NaN        NaN
```

```
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed:
4'],axis=1,inplace=True)
df.head(10)
```

```
     v1                                                  v2
0   ham  Go until jurong point, crazy.. Available only ...
1   ham                      Ok lar... Joking wif u oni...
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...
3   ham  U dun say so early hor... U c already then say...
4   ham  Nah I don't think he goes to usf, he lives aro...
5  spam  FreeMsg Hey there darling it's been 3 week's n...
6   ham  Even my brother is not like to speak with me. ...
7   ham  As per your request 'Melle Melle (Oru Minnamin...
8  spam  WINNER!! As a valued network customer you have...
9  spam  Had your mobile 11 months or more? U R entitle...
```

```
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=0.30,
random_state=7)

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)

def RNN_model():
    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

model = RNN_model()
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['ac
curacy'])

model.summary()
```

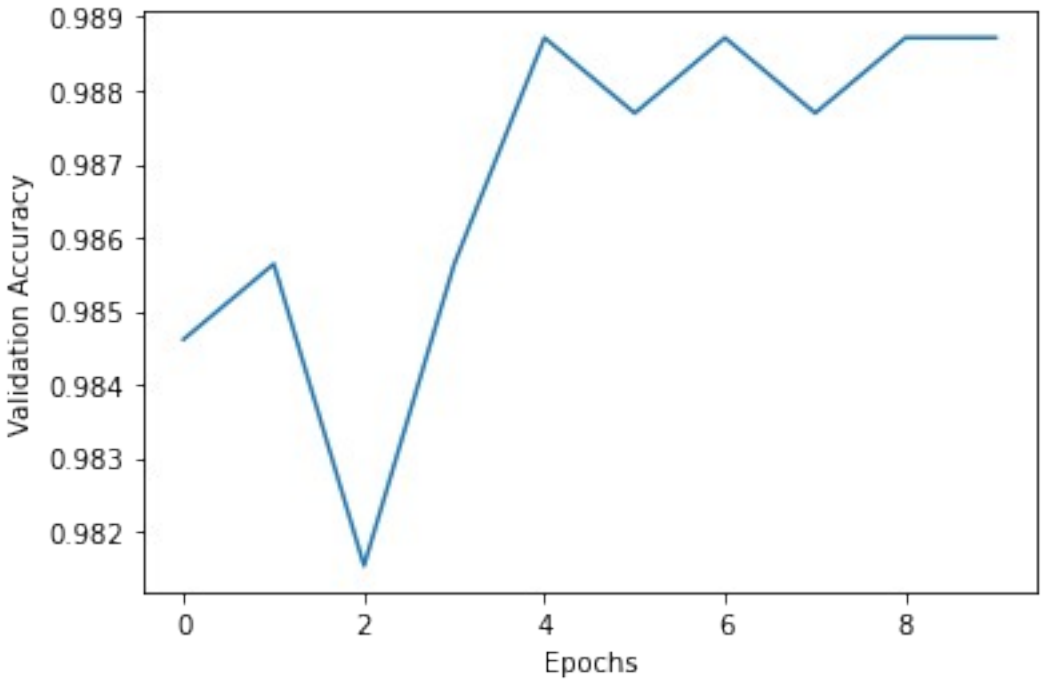
```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inputs (InputLayer)         [(None, 150)]             0
```

```
 embedding_1 (Embedding)     (None, 150, 50)           50000

 lstm_1 (LSTM)               (None, 64)                29440

 FC1 (Dense)                 (None, 256)               16640

 activation_2 (Activation)   (None, 256)               0

 dropout_1 (Dropout)         (None, 256)               0

 out_layer (Dense)           (None, 1)                 257

 activation_3 (Activation)   (None, 1)                 0

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

data =
model.fit(sequences_matrix,Y_train,batch_size=16,epochs=10,validation_
split=0.25)

Epoch 1/10
183/183 [==============================] - 13s 54ms/step - loss:
0.1779 - accuracy: 0.9419 - val_loss: 0.0689 - val_accuracy: 0.9846
Epoch 2/10
183/183 [==============================] - 9s 50ms/step - loss: 0.0358
- accuracy: 0.9901 - val_loss: 0.0631 - val_accuracy: 0.9856
Epoch 3/10
183/183 [==============================] - 9s 49ms/step - loss: 0.0173
- accuracy: 0.9949 - val_loss: 0.0982 - val_accuracy: 0.9815
Epoch 4/10
183/183 [==============================] - 9s 49ms/step - loss: 0.0101
- accuracy: 0.9969 - val_loss: 0.0889 - val_accuracy: 0.9856
Epoch 5/10
183/183 [==============================] - 9s 49ms/step - loss: 0.0024
- accuracy: 0.9993 - val_loss: 0.0763 - val_accuracy: 0.9887
Epoch 6/10
183/183 [==============================] - 9s 48ms/step - loss: 0.0011
- accuracy: 0.9993 - val_loss: 0.0741 - val_accuracy: 0.9877
Epoch 7/10
183/183 [==============================] - 9s 49ms/step - loss: 0.0036
- accuracy: 0.9993 - val_loss: 0.0938 - val_accuracy: 0.9887
Epoch 8/10
183/183 [==============================] - 9s 49ms/step - loss:
3.5366e-04 - accuracy: 1.0000 - val_loss: 0.0960 - val_accuracy:
0.9877
Epoch 9/10
183/183 [==============================] - 9s 49ms/step - loss:
```

```
1.7132e-04 - accuracy: 1.0000 - val_loss: 0.1246 - val_accuracy:
0.9887
Epoch 10/10
183/183 [==============================] - 9s 49ms/step - loss:
5.0427e-05 - accuracy: 1.0000 - val_loss: 0.1172 - val_accuracy:
0.9887
```

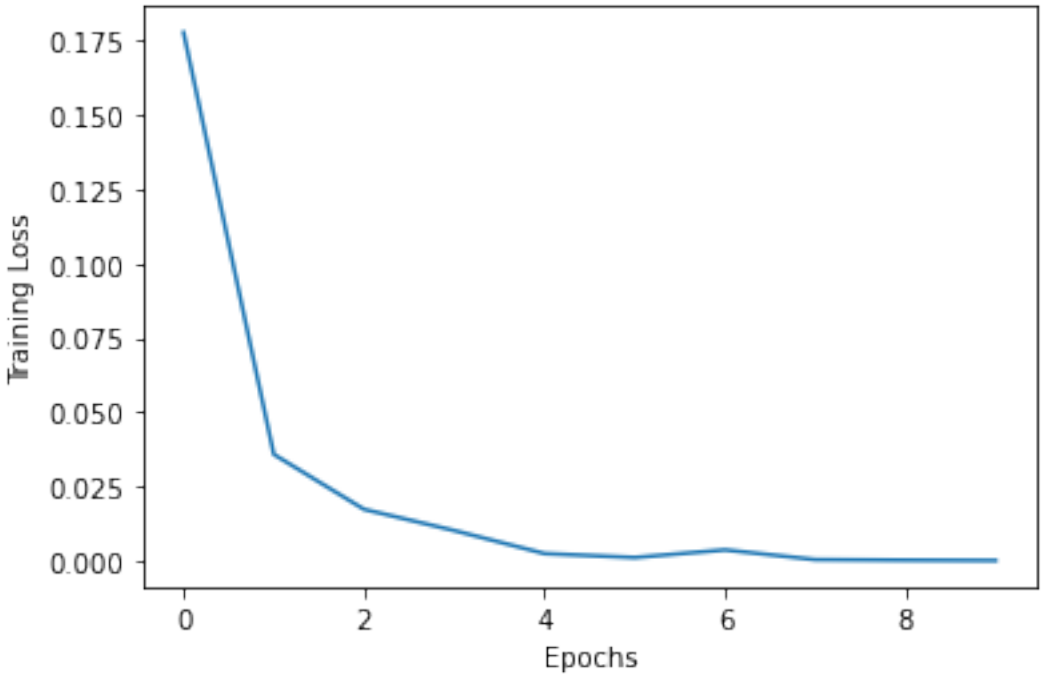
```
plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Validation Accuracy')
plt.plot(data.epoch,data.history['val_accuracy'])
```

```
[<matplotlib.lines.Line2D at 0x1dd06e79d00>]
```




```
plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Training Loss')
plt.plot(data.epoch,data.history['loss'])
```

```
[<matplotlib.lines.Line2D at 0x1dd7e556e20>]
```




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

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

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

```
53/53 [==============================] - 1s 19ms/step - loss: 0.1611 -
accuracy: 0.9809
```

```
model.metrics_names
```

```
['loss', 'accuracy']
```

```
print('Test Loss: {:0.4f} and Test Accuracy: {:0.2f}
%'.format(test_accuracy[0],test_accuracy[1]*100))
```

```
Test Loss: 0.1611 and Test Accuracy: 98.09%
```