#Import required libraries

```
import pandas as pd
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

```
df = pd.read_csv(
    'sample_data/spam.csv',
    delimiter = ',',
    encoding = 'latin-1'
)
df.drop(
    ['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],
    axis = 1,
    inplace = True
)
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 = 0.15
)
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)
```

#Add Layers (LSTM, Dense-(Hidden Layers), Output)

```
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
```

#Create model

```
model = RNN()
```

#Compile the model

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

#Fit the model

```
model.fit(
    sequences_matrix,
    Y_train,
    batch_size = 128,
    epochs=10,
    validation_split = 0.2,
    callbacks=[EarlyStopping(monitor = 'val_loss', min_delta =
0.0001)])
```

```
Epoch 1/10
30/30 [==============================] - 12s 296ms/step - loss: 0.3357
- accuracy: 0.8672 - val_loss: 0.1516 - val_accuracy: 0.9757
Epoch 2/10
30/30 [==============================] - 8s 272ms/step - loss: 0.0924
- accuracy: 0.9762 - val_loss: 0.0530 - val_accuracy: 0.9873
```

```
<keras.callbacks.History at 0x7fdb374d5250>
```

#Save the model

```
model.save('./spam.h5')
```

#Test the model

```
test_sequences = tok.texts_to_sequences(X_test)
test_sequences_matrix = pad_sequences(test_sequences, maxlen =
max_len)
accr = model.evaluate(test_sequences_matrix, Y_test)
print('Test set\n  Loss: {:0.3f}\n  Accuracy:
{:0.3f}'.format(accr[0],accr[1]))
```

```
27/27 [==============================] - 1s 26ms/step - loss: 0.0550 -
accuracy: 0.9880
```

```
Test set
  Loss: 0.055
  Accuracy: 0.988
```