

## Importing 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 tensorflow.keras.optimizers import RMSprop
from keras.preprocessing.text import Tokenizer
from keras.preprocessing import sequence
from tensorflow.keras.utils import to_categorical
from keras.callbacks import EarlyStopping
%matplotlib inline
```

## Loading the dataset

```
df = pd.read_csv(r'C:/Users/M.DEVENDRAN/Desktop/IBM_Project/ASS
4/spam.csv', delimiter=',', encoding='latin-1')
df.head()
```

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

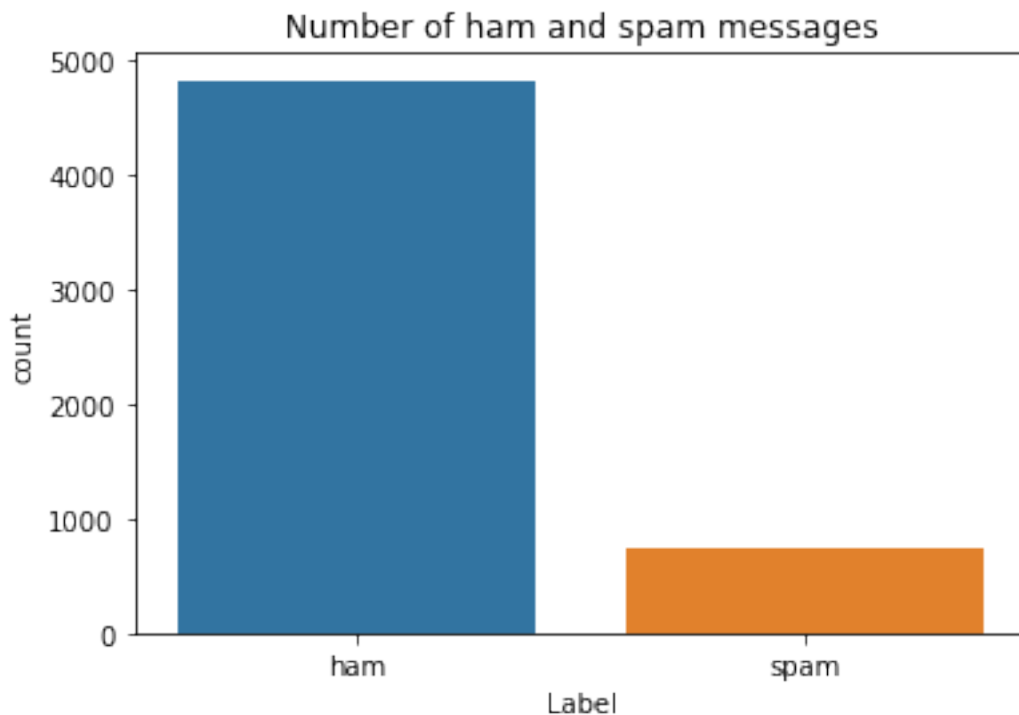
	Unnamed: 3	Unnamed: 4
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN

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

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
```

```
warnings.warn(
```

```
Text(0.5, 1.0, 'Number of ham and spam messages')
```



## Creating model

```
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_word=1000
max_len=150
tok = Tokenizer(num_words=max_word)
tok.fit_on_texts(X_train)
sequences = tok.texts_to_sequences(X_train)
sequences_matrix=sequence.pad_sequences(sequences,maxlen=max_len)
```

## Adding Layers

```
def RNN():
    inputs= Input(name='inputs',shape=[max_len])
    layer = Embedding(max_word,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
```

## Compiling the Model

```
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		

## Fitting and saving 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 [=====] - 14s 300ms/step - loss: 0.3474
- accuracy: 0.8662 - val_loss: 0.1306 - val_accuracy: 0.9800
Epoch 2/10
30/30 [=====] - 7s 237ms/step - loss: 0.0974
- accuracy: 0.9754 - val_loss: 0.0480 - val_accuracy: 0.9863

<keras.callbacks.History at 0x290f04daf10>

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

accr = model.evaluate(test_sequences_matrix,Y_test)

27/27 [=====] - 1s 35ms/step - loss: 0.0493 -
accuracy: 0.9904

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

Test set
Loss:0.049
Accuracy: 0.990
```

## Testing the model

```
Testing_context = ["oh k...i'm watching here:")]

txts= tok.texts_to_sequences(Testing_context)
txts = sequence.pad_sequences(txts, maxlen=max_len)

preds = model.predict(txts)
print(preds)

[[0.00108719]]
```