1.Spam dataset downloaded - [https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset?resource=download (https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset?resource=download)](https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset?resource=download)

2.Required libararies are imported

In [1]:

```
import numpy as np
import pandas as pd
import keras

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
#from keras.preprocessing.sequence import pad_sequences
%matplotlib inline
```

3.Read dataset and pre processing

In [2]:

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

Out[2]:

	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

In [3]:

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

In [4]:

```
df.shape
```

Out[4]:

```
(5572, 2)
```

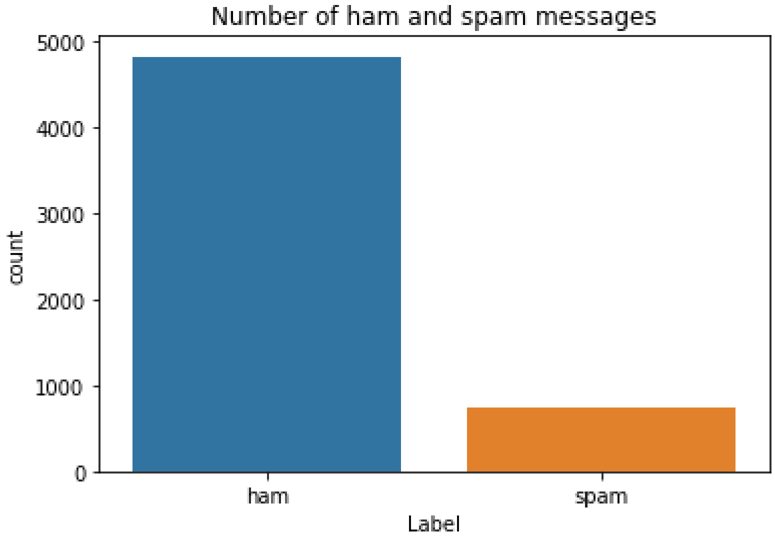
In [5]:

```
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: Pas
s the following variable as a keyword arg: x. From version 0.12, the only valid posi
tional argument will be `data`, and passing other arguments without an explicit keyw
ord will result in an error or misinterpretation.
  FutureWarning
```

Out[5]:

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



In [6]:

```
X = df.v2
Y = df.v1
#label encoding for Y
le = LabelEncoder()
Y = le.fit_transform(Y)
Y = Y.reshape(-1,1)
```

In [7]:

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.20)
```

In [8]:

```
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 = keras.utils.pad_sequences(sequences,maxlen=max_len)
```

4.Create LSTM model

5.Add layers

In [9]:

```
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)
```

6.Compile the model

In [10]:

```
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
_________________________________________________________________
```

7.Fit the model

In [11]:

```
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
28/28 [==============================] - 12s 289ms/step - loss: 0.3506 - accuracy:
0.8623 - val_loss: 0.2481 - val_accuracy: 0.9753
Epoch 2/10
28/28 [==============================] - 7s 269ms/step - loss: 0.1098 - accuracy: 0.
9742 - val_loss: 0.0550 - val_accuracy: 0.9798
```

Out[11]:

```
<keras.callbacks.History at 0x7f705d2aa550>
```

8. Save the model

In [12]:

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

9.Test the model

In [13]:

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

In [14]:

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

```
35/35 [==============================] - 1s 25ms/step - loss: 0.0575 - accuracy: 0.9
857
Test set
  Loss: 0.058
  Accuracy: 0.986
```