

```

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 tensorflow.keras.models import Model
from tensorflow.keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import EarlyStopping
%matplotlib inline

```

read dataset and preprocessing

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

```
df.head()
```

	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

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

```
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0    v1      5572 non-null    object
1    v2      5572 non-null    object
dtypes: object(2)
memory usage: 87.2+ KB

```

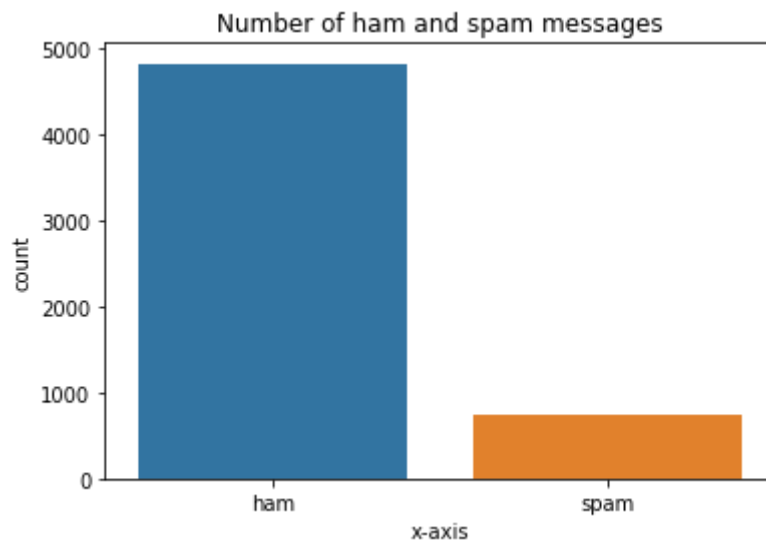
```

sns.countplot(df.v1)
plt.xlabel('x-axis')

```

```
plt.title('Number of ham and spam messages')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: P
FutureWarning
Text(0.5, 1.0, 'Number of ham and spam messages')
```



Create i/p vector and process label

```
X = df.v2
```

```
Y = df.v1
```

```
le = LabelEncoder()
```

```
Y = le.fit_transform(Y)
```

```
Y = Y.reshape(-1,1)
```

Split Training and test data

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

Data processing

```
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 = sequence.pad_sequences(sequences,maxlen=max_len)
```

Create models and add layers

```
def RNN():
    inputs = Input(name='inputs', shape=[max_len])
    layer = Embedding(max_words, 50, input_length=max_len)(inputs)
    layer = LSTM(128)(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('tanh')(layer)
    model = Model(inputs=inputs, outputs=layer)
    return model
```

```
model = RNN()
```

```
model.summary()
```

Model: "model"

Layer (type)	Output Shape	Param #
inputs (InputLayer)	[(None, 150)]	0
embedding (Embedding)	(None, 150, 50)	50000
lstm (LSTM)	(None, 128)	91648
FC1 (Dense)	(None, 256)	33024
activation (Activation)	(None, 256)	0
dropout (Dropout)	(None, 256)	0
out_layer (Dense)	(None, 1)	257
activation_1 (Activation)	(None, 1)	0
Total params: 174,929		
Trainable params: 174,929		
Non-trainable params: 0		

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

Fit the model

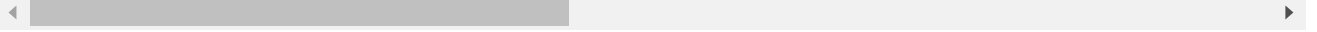
```
model.fit(sequences_matrix, Y_train, batch_size=128, epochs=100,
          validation_split=0.2, callbacks=[EarlyStopping(monitor='val_loss', min_delta=0.000
```

Epoch 1/100

28/28 [=====] - 21s 631ms/step - loss: 0.3445 - accuracy: 0

Epoch 2/100

28/28 [=====] - 16s 583ms/step - loss: 0.1376 - accuracy: 0
<keras.callbacks.History at 0x7f881ca592d0>

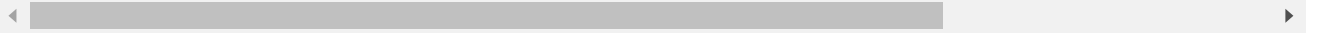


```
test_sequences = tok.texts_to_sequences(X_test)
```

```
test_sequences_matrix = sequence.pad_sequences(test_sequences,maxlen=max_len)
```

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

35/35 [=====] - 3s 77ms/step - loss: 0.1860 - accuracy: 0.98



```
print('Test set\n Loss: {:.3f}\n Accuracy: {:.3f}'.format(accr[0],accr[1]))
```

Test set

Loss: 0.186

Accuracy: 0.981

```
model.save(r"C:\Users\it\Desktop\model_lstm.h5")
```

Run the model

```
from tensorflow.keras.models import load_model
```

```
m2 = load_model(r"C:\Users\it\Desktop\model_lstm.h5")
```

```
m2.evaluate(test_sequences_matrix,Y_test)
```

35/35 [=====] - 3s 80ms/step - loss: 0.1860 - accuracy: 0.98

```
[0.1859765499830246,  
 0.9811659455299377,  
 0.07287588715553284,  
 0.23770132660865784]
```



[Colab paid products](#) - [Cancel contracts here](#)

✓ 3s completed at 2:21 PM

