# Assignment-4

## VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning

| Assignment Date | November 14, 2022 |
|---|---|
| Student Name | SATHEESH RATHNAVEL |
| Student Roll Number | 2127190801073 |
| Maximum Marks | 2 Marks |

## Question-1:

Download the dataset

**Solution:**

**Download the given dataset in the given attached link.**

| | A | B |
|---|---|---|
| 1 | v1 | v2 |
| 2 | ham | Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat... |
| 3 | ham | Ok lar... Joking wif u oni... |
| 4 | spam | Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's |
| 5 | ham | U dun say so early hor... U c already then say... |
| 6 | ham | Nah I don't think he goes to usf, he lives around here though |
| 7 | spam | FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, å£1.50 to rcv |
| 8 | ham | Even my brother is not like to speak with me. They treat me like aids patent. |
| 9 | ham | As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune |
| 10 | spam | WINNER!! As a valued network customer you have been selected to receivea å£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only. |
| 11 | spam | Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030 |
| 12 | ham | I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today. |
| 13 | spam | SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info |
| 14 | spam | URGENT! You have won a 1 week FREE membership in our å£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18 |
| 15 | ham | I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times. |
| 16 | ham | I HAVE A DATE ON SUNDAY WITH WILL!! |
| 17 | spam | XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL |
| 18 | ham | Oh k...i'm watching here:) |
| 19 | ham | Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet. |
| 20 | ham | Fine if thatåÕs the way u feel. ThatåÕs the way its gota b |
| 21 | spam | England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/Ì¼1.20 POBOXox36504W45WQ 16+ |
| 22 | ham | Is that seriously how you spell his name? |
| 23 | ham | I‰Û÷m going to try for 2 months ha ha only joking |
| 24 | ham | So Ì_ pay first lar... Then when is da stock comin... |
| 25 | ham | Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already? |
| 26 | ham | Ffffffffff. Alright no way I can meet up with you sooner? |
| 27 | ham | Just forced myself to eat a slice. I'm really not hungry tho. This sucks. Mark is getting worried. He knows I'm sick when I turn down pizza. Lol |
| 28 | ham | Lol your always so convincing. |
| 29 | ham | Did you catch the bus ? Are you frying an egg ? Did you make a tea? Are you eating your mom's left over dinner ? Do you feel my Love ? |
| 30 | ham | I'm back &amp; we're packing the car now. I'll let you know if there's room |

## Question-2:

**Import required library**

**Solution:**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**import tensorflow as tensorflow**

**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**

```python
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import tensorflow as tensorflow
        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, E
        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
```

## Question 3:

**Read dataset and do pre-processing**

**Solution:**

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

```
In [2]: df = pd.read_csv(r'spam.csv',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)
        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
```

## Question 4:

● **Create Model**

● **Add Layers (LSTM, Dense-(Hidden Layers), Output)**

● **Compile the Model**

● **Fit the Model**

● **Save The Model**

● **Test The Model**

**Solution:**

**sns.countplot(df.v1)**

**plt.xlabel('Label')**

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

**X = df.v2**

**Y = df.v1**

**le = LabelEncoder()**

**Y = le.fit_transform(Y)**

```python
Y = Y.reshape(-1,1)
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.20)
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)
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.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy','mse','mae'])
model.save(r"C:\Users\MAGGIE\model_lSTM.h5")
from tensorflow.keras.models import load_model
m2 = load_model(r"C:\Users\MAGGIE\model_lSTM.h5")
m2.evaluate(test_sequences_matrix,Y_test)
```
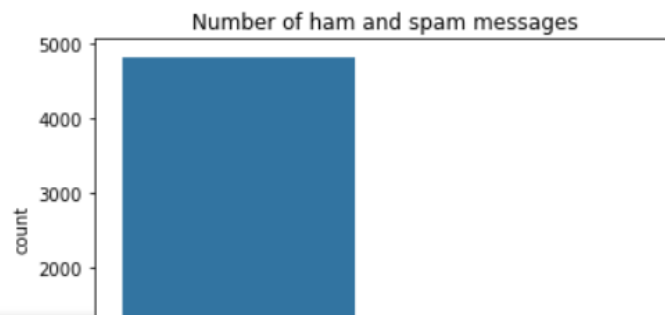
```python
In [4]:  sns.countplot(df.v1)
         plt.xlabel('Label')
         plt.title('Number of ham and spam messages')
```

```
Out[4]:  Text(0.5, 1.0, 'Number of ham and spam messages')
```



```python
In [5]:  X = df.v2
         Y = df.v1
         le = LabelEncoder()
         Y = le.fit_transform(Y)
         Y = Y.reshape(-1,1)
```

```python
In [6]:  X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.20)
```

```python
In [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 = sequence.pad_sequences(sequences,maxlen=max_len)
```

```python
In [8]:  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
```

```
In [9]: model = RNN()
        model.summary()
        model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy','mse','mae'])
```

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
_____

```
In [10]: 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 [==============================] - 18s 512ms/step - loss: 0.2518 - accuracy: 0.9072 - mse: 0.0687 - mae: 0.1437 - val_los
s: 0.1685 - val_accuracy: 0.9675 - val_mse: 0.0306 - val_mae: 0.1025
Epoch 2/10
28/28 [==============================] - 12s 413ms/step - loss: 0.0801 - accuracy: 0.9843 - mse: 0.0195 - mae: 0.0883 - val_los
s: 0.1846 - val_accuracy: 0.9720 - val_mse: 0.0306 - val_mae: 0.1086

Out[10]: <keras.callbacks.History at 0x2ac38ddae80>

```
In [11]: test_sequences = tok.texts_to_sequences(X_test)
         test_sequences_matrix = sequence.pad_sequences(test_sequences,maxlen=max_len)
```

```
In [12]: accr = model.evaluate(test_sequences_matrix,Y_test)
```

35/35 [==============================] - 2s 70ms/step - loss: 0.1396 - accuracy: 0.9794 - mse: 0.0257 - mae: 0.1030

```
In [13]: print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))
```

Test set
  Loss: 0.140
  Accuracy: 0.979

```
In [18]: model.save(r"C:\Users\MAGGIE\model_lSTM.h5")
```

```
In [19]: from tensorflow.keras.models import load_model
         m2 = load_model(r"C:\Users\MAGGIE\model_lSTM.h5")
```

```
In [20]: m2.evaluate(test_sequences_matrix,Y_test)
```

35/35 [==============================] - 3s 57ms/step - loss: 0.1396 - accuracy: 0.9794 - mse: 0.0257 - mae: 0.1030

Out[20]: [0.13960915803909302,
 0.9793722033500671,
 0.02572023682296276,
 0.10301480442285538]

```
In [ ]:
```