

Assignment -IV STM for Text Classification

Assignment Date	03 November 2022
Student Name	RENUKA DEVI D
Student Roll Number	9517201903122
Maximum Marks	2

#Import necessary libraries

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
%matplotlib inline
```

```
from sklearn.model_selection import train_test_split
```

```
from keras.layers import Dense , LSTM , Embedding , Dropout , Activation ,  
Flatten
```

```
from sklearn.preprocessing import LabelEncoder
```

```
from keras.preprocessing.text import Tokenizer
```

```
from keras.models import Sequential
```

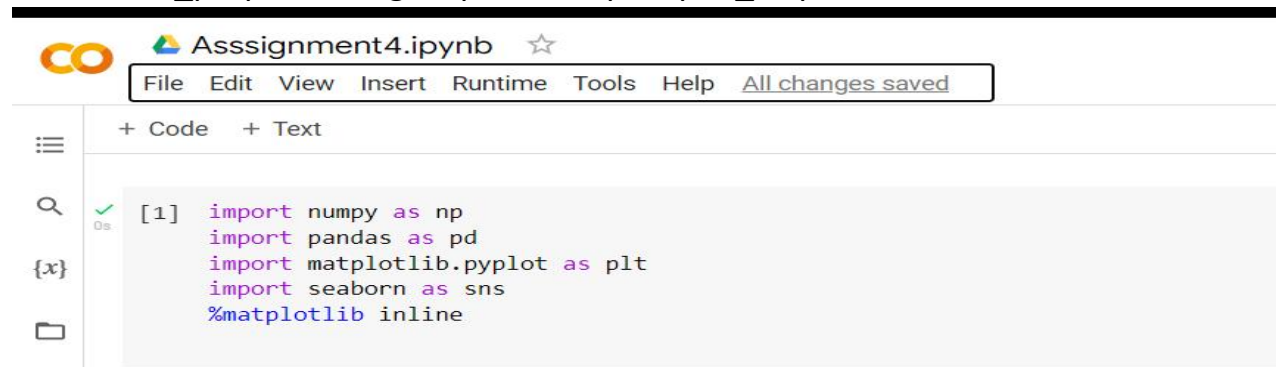
```
from tensorflow.keras.preprocessing import sequence
```


```
from tensorflow.keras.utils import to_categorical
```

```
from keras.callbacks import EarlyStopping
```

```
from tensorflow.keras.optimizers import RMSprop
```

```
from keras_preprocessing.sequence import pad_sequences
```



```
3s  from sklearn.model_selection import train_test_split
from keras.layers import Dense , LSTM , Embedding , Dropout , Activation , Flatten
from sklearn.preprocessing import LabelEncoder
from keras.preprocessing.text import Tokenizer
from keras.models import Sequential
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.utils import to_categorical
from keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import RMSprop
from keras_preprocessing.sequence import pad_sequences
```

#Read dataset and do pre-processing

```
data = pd.read_csv('/content/spam.csv',delimiter=',',encoding='latin-1')
```

```
data
```

```
#Information about dataset
```

```
data.describe().T
```

```
data.shape
```

```
#Check if there is any missing values
```

```
data.isnull().sum()
```

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

```
#Visualize the dataset
```

```
sns.countplot(data.v1)
```

```
#Preprocess using Label Encoding
```

```
X = data.v2
```

```
Y = data.v1
```

```
le = LabelEncoder()
```

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

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

1s

[5] data = pd.read_csv('/content/drive/MyDrive/spam.csv',delimiter=',',encoding='latin-1')
data

<>

v1v2 Unnamed: 2 Unnamed: 3 Unnamed: 4

0hamGo until jurong point, crazy.. Available only ...NaNNaNNaN

1hamOk lar... Joking wif u oni...NaNNaNNaN

0scompleted at 7:14 PM

Assignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

1s

data = pd.read_csv('/content/drive/MyDrive/spam.csv',delimiter=',',encoding='latin-1')
data

<>

v1v2 Unnamed: 2 Unnamed: 3 Unnamed: 4

0hamGo until jurong point, crazy.. Available only ...NaNNaNNaN

1hamOk lar... Joking wif u oni...NaNNaNNaN

2spamFree entry in 2 a wkly comp to win FA Cup fina...NaNNaNNaN

3hamU dun say so early hor... U c already then say...NaNNaNNaN

4hamNah I don't think he goes to usf, he lives aro...NaNNaNNaN

... ..NaNNaNNaN

5567spamThis is the 2nd time we have tried 2 contact u...NaNNaNNaN

5568hamWill I _ b going to esplanade fr home?NaNNaNNaN

5569hamPity, * was in mood for that. So...any other s...NaNNaNNaN

5570hamThe guy did some bitching but I acted like I'd...NaNNaNNaN

5571hamRofl. Its true to its nameNaNNaNNaN

5572 rows x 5 columns

0s

[6] data.describe().T

Assignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

0s

[6] data.describe().T

{x}

count unique top freq

v155722ham4825

v255725169Sorry, I'll call later30

Unnamed: 25043bt not his girfrnd... G o o d n i g h t . . @ "3

Unnamed: 31210MK17 92H. 450Ppw 16"2

Unnamed: 465GNT:-)"2

0s

data.shape

(5572, 5)

0s

[8] data.isnull().sum()

<>

v10

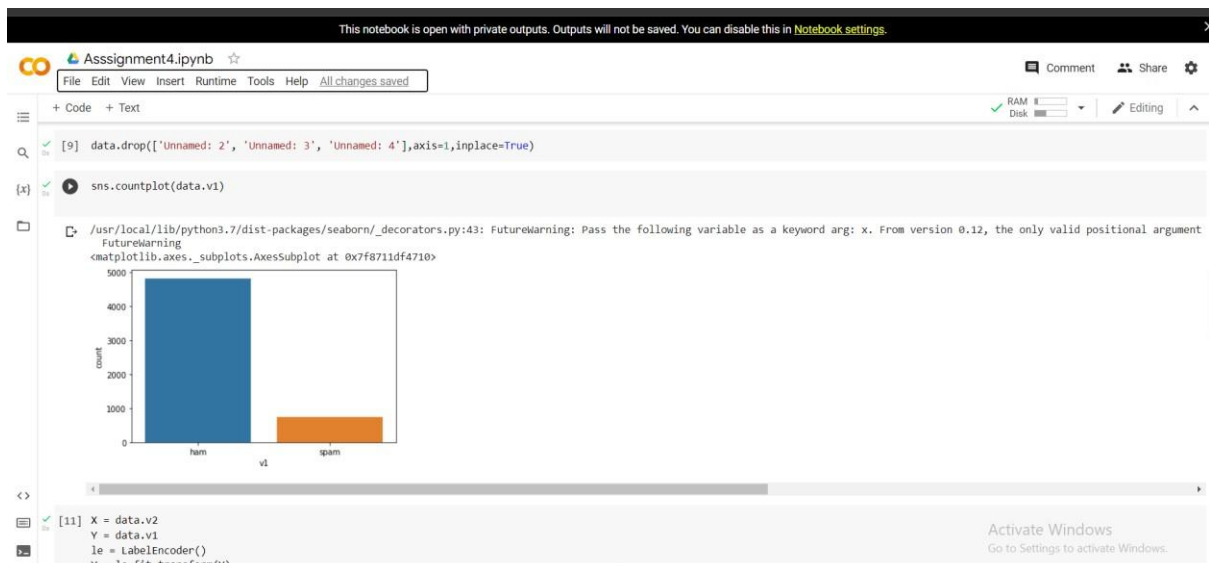
v20

Unnamed: 25522

Unnamed: 35560

Unnamed: 45566

dtype: int64



Assignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[11] X = data.v2
Y = data.v1
le = LabelEncoder()
Y = le.fit_transform(Y)

#Create Model and Add Layers (LSTM, Dense-(Hidden Layers), Output)

#Splitting into training and testing data

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.2)
```

```
max_word = 1000
```

```
max_len = 250
```

```
token = Tokenizer(num_words = max_word)
```

```
token.fit_on_texts(X_train)
```

```
sequences = token.texts_to_sequences(X_train)
```

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

#Creating the model

```
model = Sequential()
```

```
model.add(Embedding(max_word , 32 , input_length = max_len))
```

```
model.add(LSTM(64))
```

```
model.add(Flatten())
```

```
model.add(Dense(250, activation='relu'))
```

```
model.add(Dropout(0.5))
```

```
model.add(Dense(120, activation='relu'))
```

```
model.add(Dense(1, activation='sigmoid'))
```

```
✓ [13] X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.2)
```

```
✓ 0s  
▶ max_word = 1000  
max_len = 250  
token = Tokenizer(num_words = max_word)  
token.fit_on_texts(X_train)  
sequences = token.texts_to_sequences(X_train)  
seq_matrix = sequence.pad_sequences(sequences , maxlen = max_len)
```

```
✓ [26] model = Sequential()  
1s model.add(Embedding(max_word , 32 , input_length = max_len))  
model.add(LSTM(64))  
model.add(Flatten())  
<> model.add(Dense(250, activation='relu'))  
model.add(Dropout(0.5))  
model.add(Dense(120, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))  
model.add(Dense(1, activation='sigmoid'))
```

✓ 0s completed at 7:14 PM

#compile the model

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

```
model.summary()
```

Assignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

Model: "sequential_1"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 250, 32)	32000
lstm_1 (LSTM)	(None, 64)	24832
flatten_1 (Flatten)	(None, 64)	0
dense (Dense)	(None, 250)	16250
dropout (Dropout)	(None, 250)	0
dense_1 (Dense)	(None, 120)	30120
dense_2 (Dense)	(None, 1)	121

=====
Total params: 103,323
Trainable params: 103,323
Non-trainable params: 0
=====
<>

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

#Fit the model

```
model.fit(seq_matrix,Y_train,batch_size=128,epochs=10,validation_split=0.2,c
allbacks=[EarlySt
```

```
opping(monitor='val_loss',min_delta=0.0001)])
```

```
test_seq = token.texts_to_sequences(X_test)
```

```
test_seq_matrix = sequence.pad_sequences(test_seq,maxlen=max_len)
```

Assignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[28] model.fit(seq_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 [=====] - 24s 753ms/step - loss: 0.3546 - accuracy: 0.8749 - val_loss: 0.2016 - val_accuracy: 0.9137
Epoch 2/10
28/28 [=====] - 10s 351ms/step - loss: 0.0952 - accuracy: 0.9753 - val_loss: 0.0734 - val_accuracy: 0.9787
<keras.callbacks.History at 0x7f870ce0ae50>

```
[29] test_seq = token.texts_to_sequences(X_test)
test_seq_matrix = sequence.pad_sequences(test_seq,maxlen=max_len)
```

```
[30] model.save(r'lstn_model.h5')
```

```
from tensorflow.keras.models import load_model
new_model=load_model(r'lstn_model.h5')
```

```
[32] new_model.evaluate(test_seq_matrix,Y_test)
```

35/35 [=====] - 2s 32ms/step - loss: 0.0392 - accuracy: 0.9883
[0.03923581913113594, 0.9883407950401306]

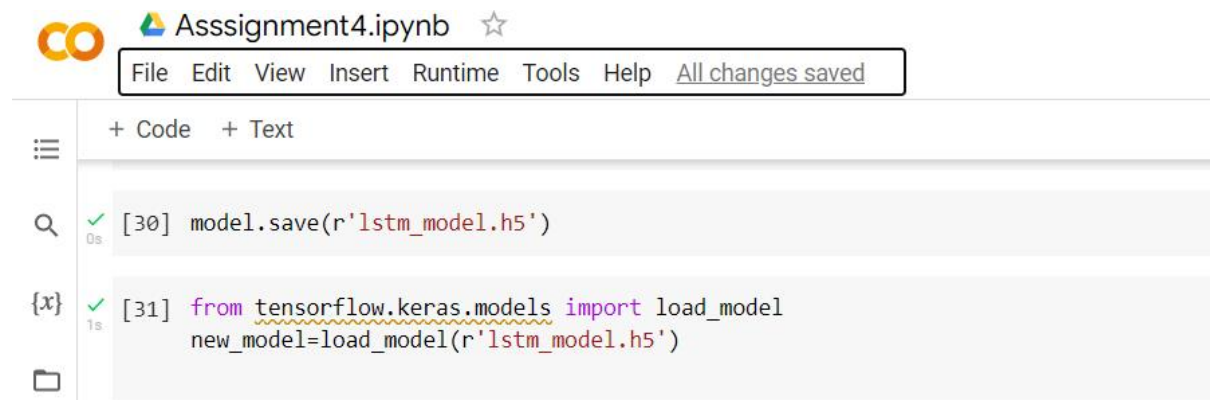
```
[33] scores = model.evaluate(test_seq_matrix, Y_test, verbose=0)
scores
```

0s completed at 7:14 PM

Activate Windows
Go to Settings to activate Windows.

#Save the model

```
model.save(r'lstm_model.h5')
```



Asssignment4.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

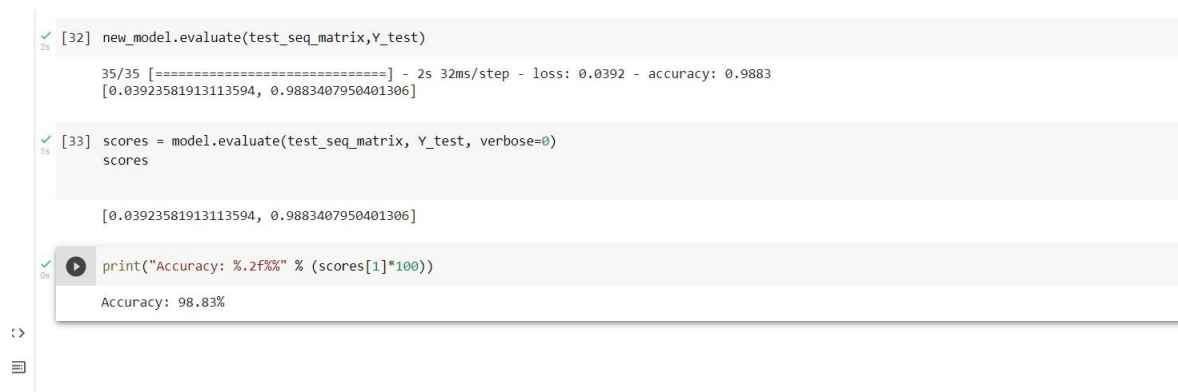
+ Code + Text

[30] `model.save(r'lstm_model.h5')`

[31] `from tensorflow.keras.models import load_model`
`new_model=load_model(r'lstm_model.h5')`

#Test the model:

```
from tensorflow.keras.models import load_model
new_model=load_model(r'lstm_model.h5')
new_model.evaluate(test_seq_matrix,Y_test)
scores = model.evaluate(test_seq_matrix, Y_test, verbose=0)
scores
print("Accuracy: %.2f%%" % (scores[1]*100))
```



[32] `new_model.evaluate(test_seq_matrix,Y_test)`

35/35 [=====] - 2s 32ms/step - loss: 0.0392 - accuracy: 0.9883
[0.03923581913113594, 0.9883407950401306]

[33] `scores = model.evaluate(test_seq_matrix, Y_test, verbose=0)`
`scores`

[0.03923581913113594, 0.9883407950401306]

`print("Accuracy: %.2f%%" % (scores[1]*100))`

Accuracy: 98.83%