

Assignment -IV STM for Text Classification

Assignment Date	11 November 2022
Student Name	S.Padma Priya
Student Roll Number	9517201903100
Maximum Marks	2 Marks

#Import necessary libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

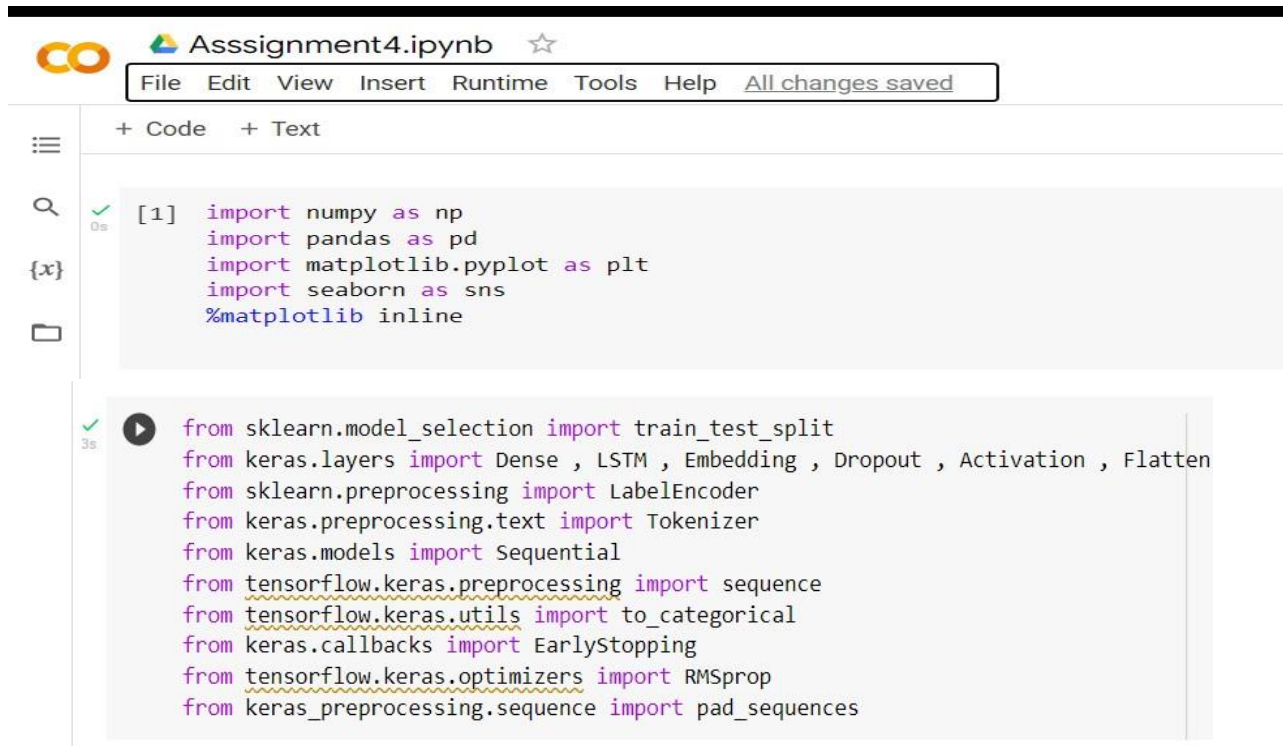
from sklearn.model_selection import train_test_split

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
```



```
Assignment4.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[1] 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
```

#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)
```

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

	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

✓ 0s completed at 7:14 PM

This notebook is open with private outputs. Outputs will not be saved. You c

Assignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

	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
...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will I_b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like I'd...	NaN	NaN	NaN
5571	ham	Roff. Its true to its name	NaN	NaN	NaN

5572 rows x 5 columns

```
[6] data.describe().T
```

Asssignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[6] data.describe().T
```

	count	unique	top	freq
v1	5572	2	ham	4825
v2	5572	5169	Sorry, I'll call later	30
Unnamed: 2	50	43	bt not his girlfrnd... G o o d n i g h t . . @"	3
Unnamed: 3	12	10	MK17 92H. 450Ppw 16"	2
Unnamed: 4	6	5	GNT:-)"	2

```
data.shape
```

```
(5572, 5)
```

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

```
v1      0
v2      0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
dtype: int64
```

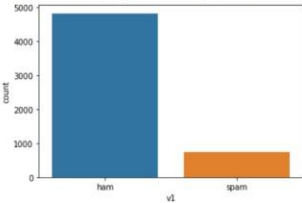
Asssignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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



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

Asssignment4.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)
```

```
✓ 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)
```

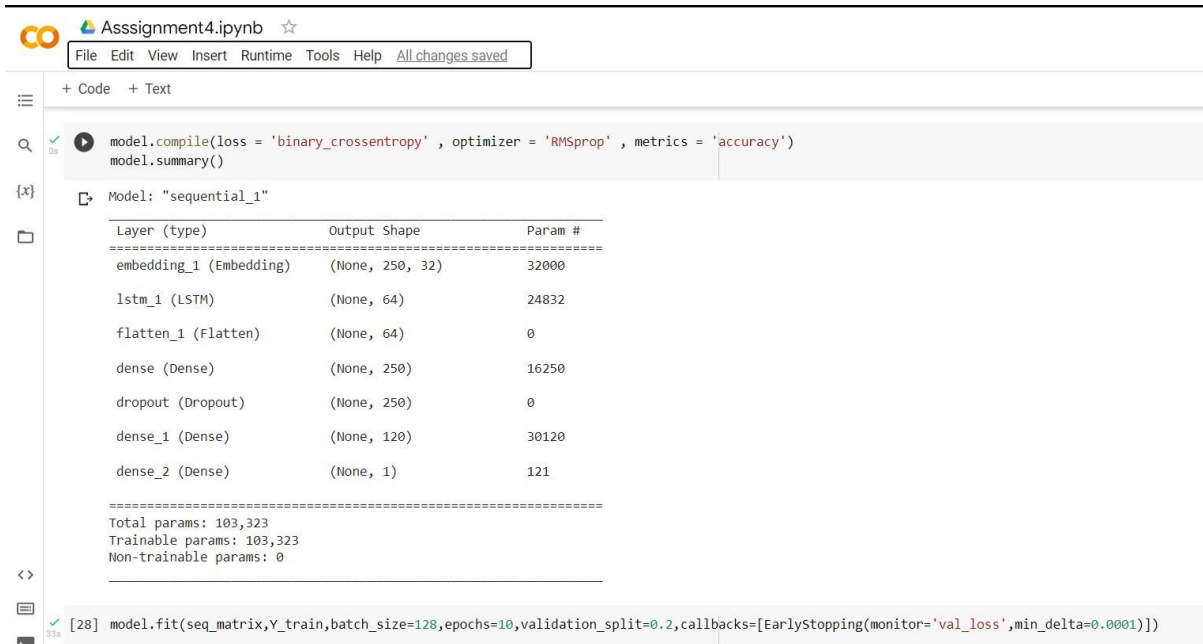
```
✓ [26] model = Sequential()  
model.add(Embedding(max_word , 32 , input_length = max_len))  
model.add(LSTM(64))  
model.add(Flatten())  
model.add(Dense(250, activation='relu'))  
model.add(Dropout(0.5))  
model.add(Dense(120, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```

✓ 0s completed at 7:14 PM

#compile the model

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

'accuracy') model.summary()



The screenshot shows a Jupyter Notebook interface with the following content:

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

Model: "sequential_1"

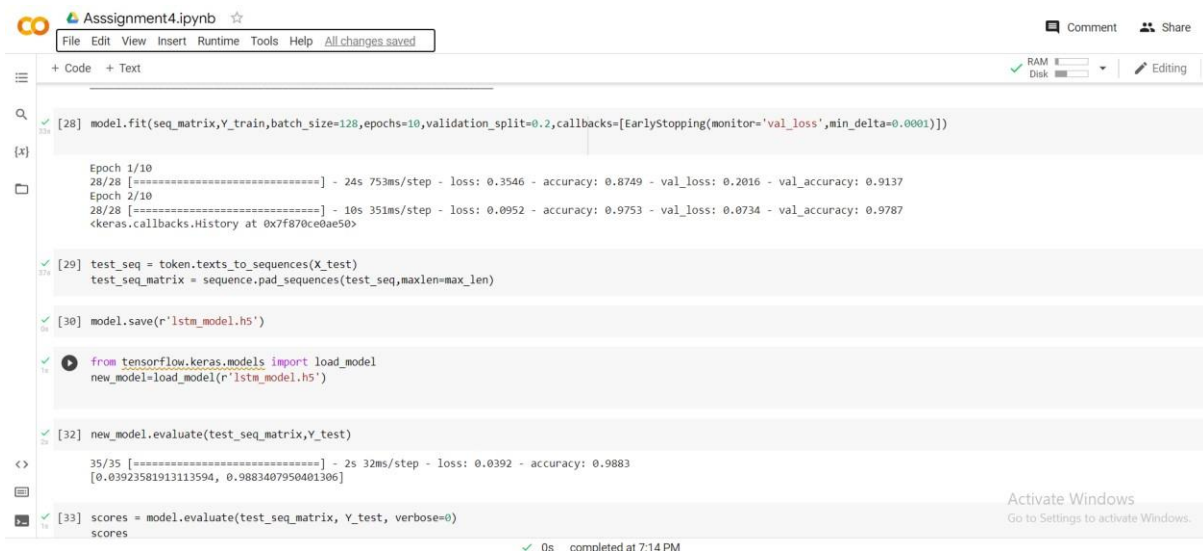
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 250, 32)	32000
lstm_1 (LSTM)	(None, 64)	24832
flatten_1 (Flatten)	(None, 64)	0
dense (Dense)	(None, 250)	16250
dropout (Dropout)	(None, 250)	0
dense_1 (Dense)	(None, 120)	30120
dense_2 (Dense)	(None, 1)	121

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, callbacks=[EarlyStopping(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)
```



The screenshot shows a Jupyter Notebook interface with the following content:

```
[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'lstm_model.h5')
```

```
from tensorflow.keras.models import load_model
new_model = load_model(r'lstm_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
```

completed at 7:14 PM

#Save the model

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

 Assignment4.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%
```