

Assignment Date	17 October 2022
Student Name	S.Abisha
Student Roll Number	962719106002
Maximum Marks	2Marks

1.Download the Dataset

Link: https://drive.google.com/file/d/1Sjqx5H5R86tRp2YZKzdd4_iEfjChZ3ob/view?usp=share_link

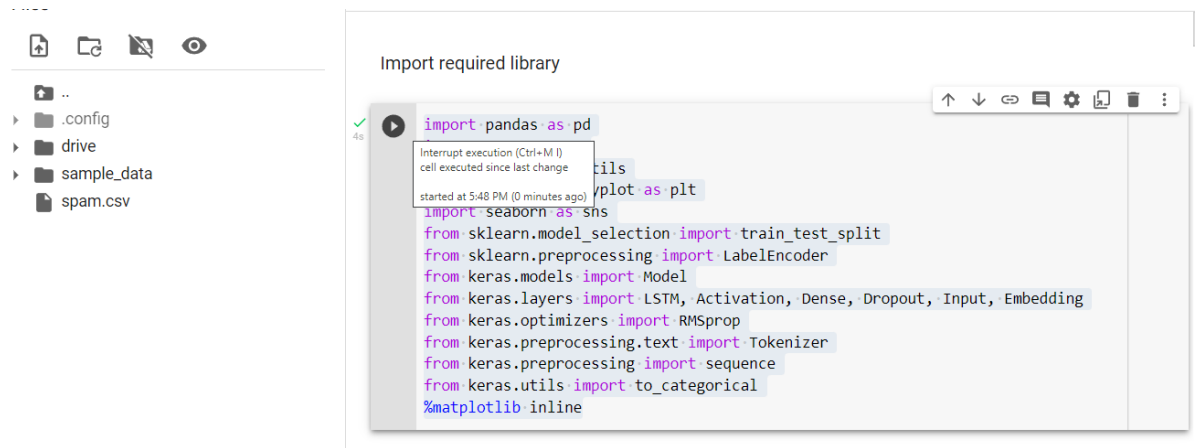
2.Import required library

Solution:

```
import pandas as pd

import numpy as np

from keras import utils
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
%matplotlib inline
```



3. Read dataset and do pre-processing

Read Data

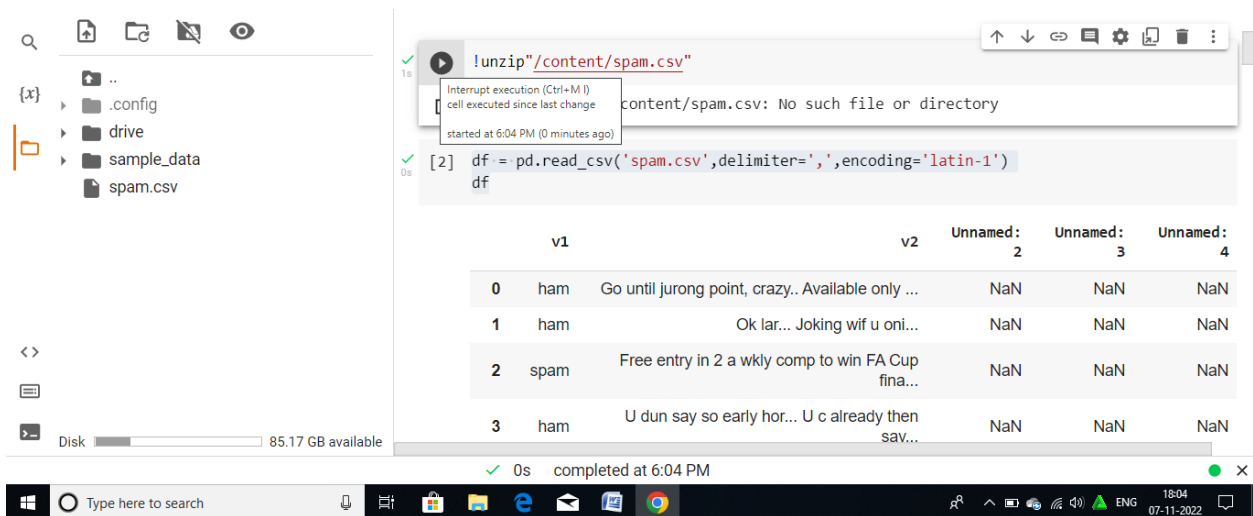
Solution:

```

!unzip"/content/spam.csv"
df = pd.read_csv('spam.csv',delimiter=',',encoding='latin-1')

```

df



Pre processing

Solution:

```

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

```

df

```
sns.countplot(df.v1,palette='Set3')

plt.xlabel('Label')
plt.title('Number of ham and spam messages')
X=df.v2
Y=df.v1
le=LabelEncoder()
Y=le.fit_transform(Y)
Y=Y.reshape(-1,1)
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.15)
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 = utils.pad_sequences(sequences,maxlen=max_len)
sequences_matrix.shape
sequences_matrix.ndim
sequences_matrix=np.reshape(sequences_matrix,(4736,150,1))
sequences_matrix.ndim
```

The screenshot shows a Jupyter Notebook environment. On the left is a file explorer with a search bar and icons for file operations. It displays a directory structure with files like `.config`, `drive`, `sample_data`, and `spam.csv`. The main area on the right contains a code cell with the following text:

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

Below the code cell is a preview of the DataFrame `df`. It has two columns: `v1` and `v2`. The rows show a mix of 'ham' and 'spam' messages. The first few rows are:

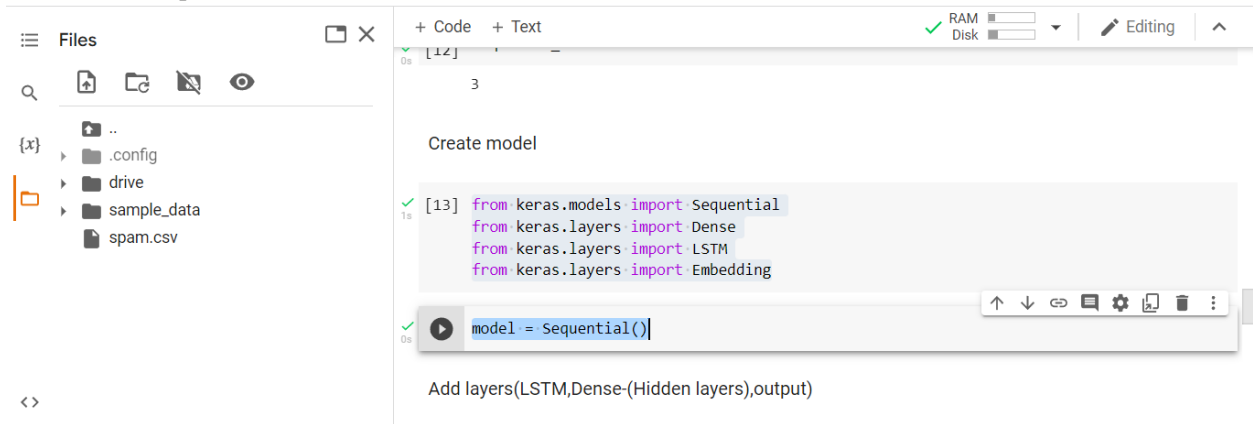
	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

The preview continues with an ellipsis and then shows rows 5567, 5568, and 5569. At the bottom of the interface, a status bar indicates 'Disk 85.17 GB available'.


```

from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Embedding
model = Sequential()

```

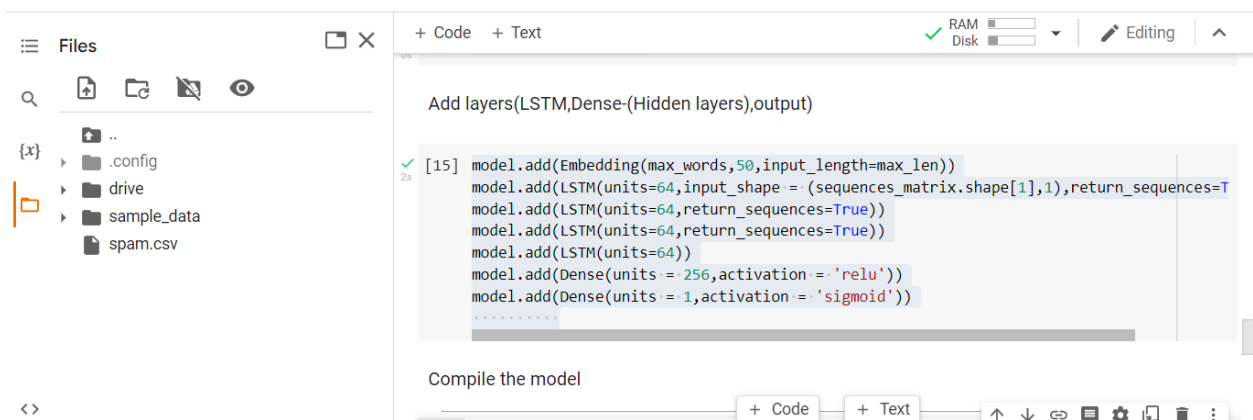


5.Add layers(LSTM,Dense-(Hidden layers),output)

```

model.add(Embedding(max_words,50,input_length=max_len))
model.add(LSTM(units=64,input_shape = (sequences_matrix.shape[1],1),return_sequences=True))
model.add(LSTM(units=64,return_sequences=True))
model.add(LSTM(units=64,return_sequences=True))
model.add(LSTM(units=64))
model.add(Dense(units = 256,activation = 'relu'))
model.add(Dense(units = 1,activation = 'sigmoid'))

```



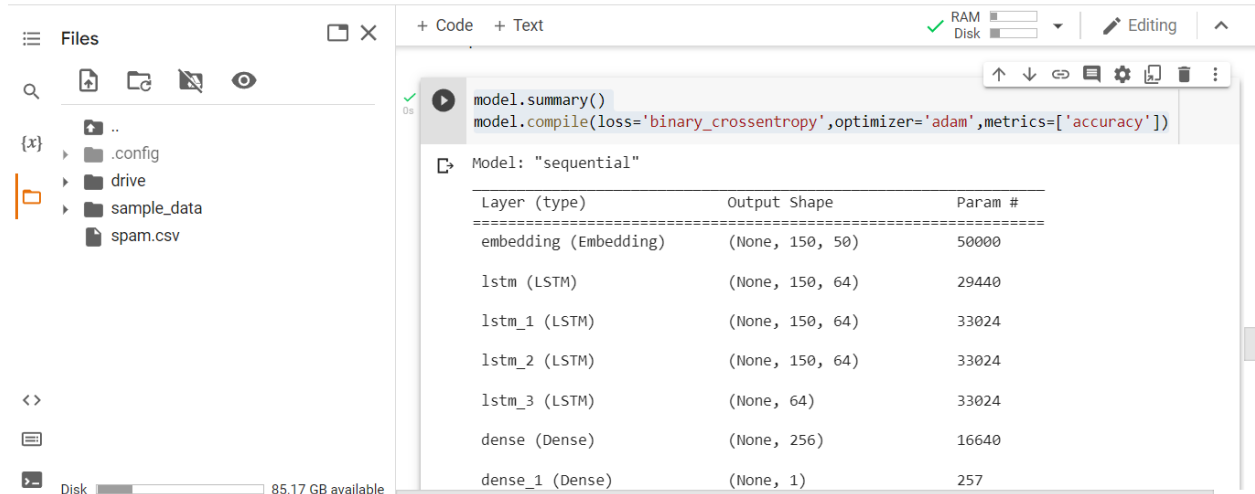
6.Compile the model

Solution:

```

model.summary()
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])

```

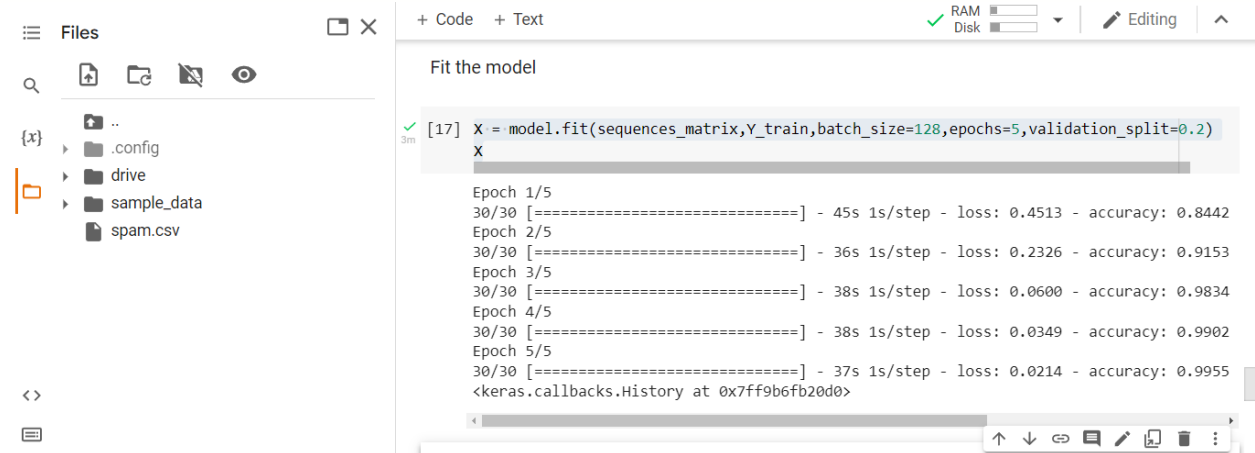


7. Fit the Model

Solution:

```
X = model.fit(sequences_matrix, Y_train, batch_size=128, epochs=5, validation_split=0.2)
```

X



8. Save the Model

Solution:

```
model.save
```

Files

RAM
Disk

Editing

Epoch 4/5
30/30 [=====] - 38s 1s/step - loss: 0.0349 - accuracy: 0.9902
Epoch 5/5
30/30 [=====] - 37s 1s/step - loss: 0.0214 - accuracy: 0.9955
<keras.callbacks.History at 0x7ff9bb9d0390>

Save the model

model.save

<bound method Model.save of <keras.engine.sequential.Sequential object at 0x7ff9bb9d0390>>

Test the model

9. Test the Model

Solution:

```
test_sequences = tok.texts_to_sequences(X_test)
test_sequences_matrix = utils.pad_sequences(test_sequences, maxlen=max_len)
accr = model.evaluate(test_sequences_matrix, Y_test)
l = accr[0]
a = accr[1]
print('Test set\n Loss: {:.3f}\n Accuracy: {:.3f}'.format(l, a))
```

Files

RAM
Disk

Editing

Test the model

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

accr = model.evaluate(test_sequences_matrix, Y_test)

27/27 [=====] - 10s 182ms/step - loss: 0.0897 - accuracy: 0.98

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

Test set
Loss: 0.090
Accuracy: 0.981

Disk 85.17 GB available

