*PYTHON Deep LearningLAB-2 ASSIGNMENT*

*Team Members*

*Aarthi Nagireddy 16292448 Class ID : 24*

*Manaswini Vedula 16292800 Class ID : 35*

*Yamini Bommineni 16292447 Class ID : 10*

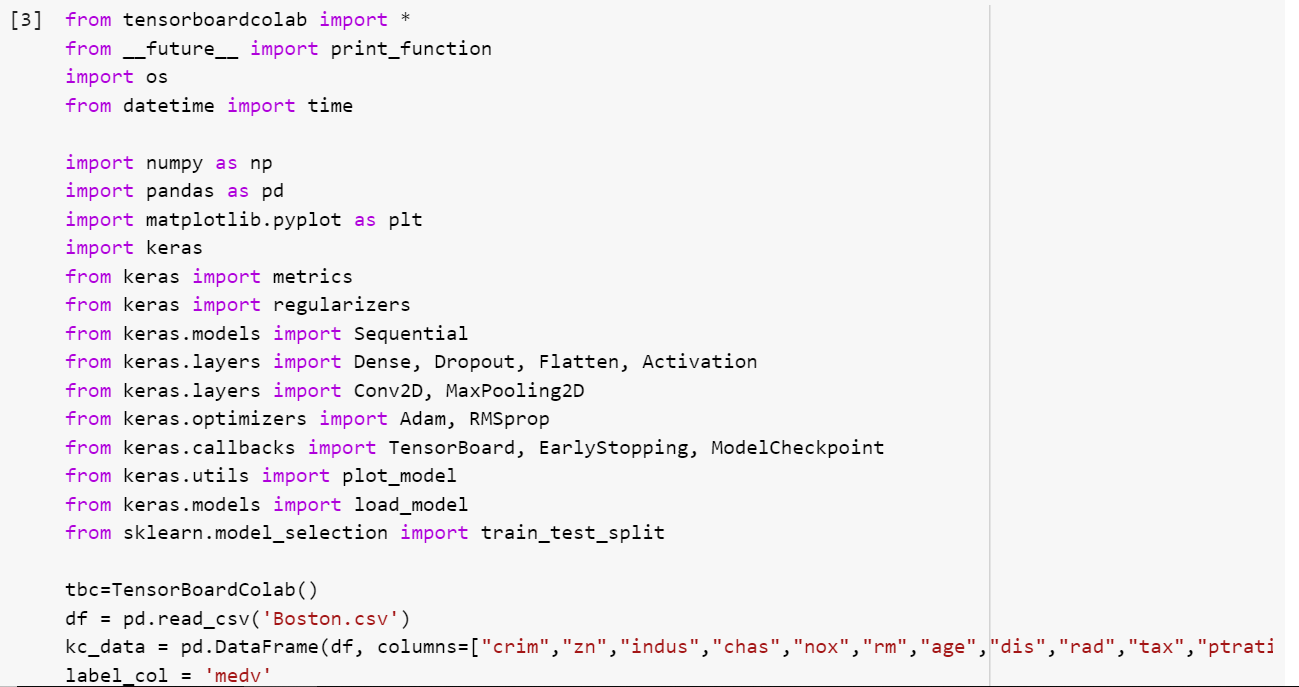
*WorkFlow:*

*Task-1:* *Implement Linear Regression with any data set of your choice except the datasets being discussed in the class or source code.*

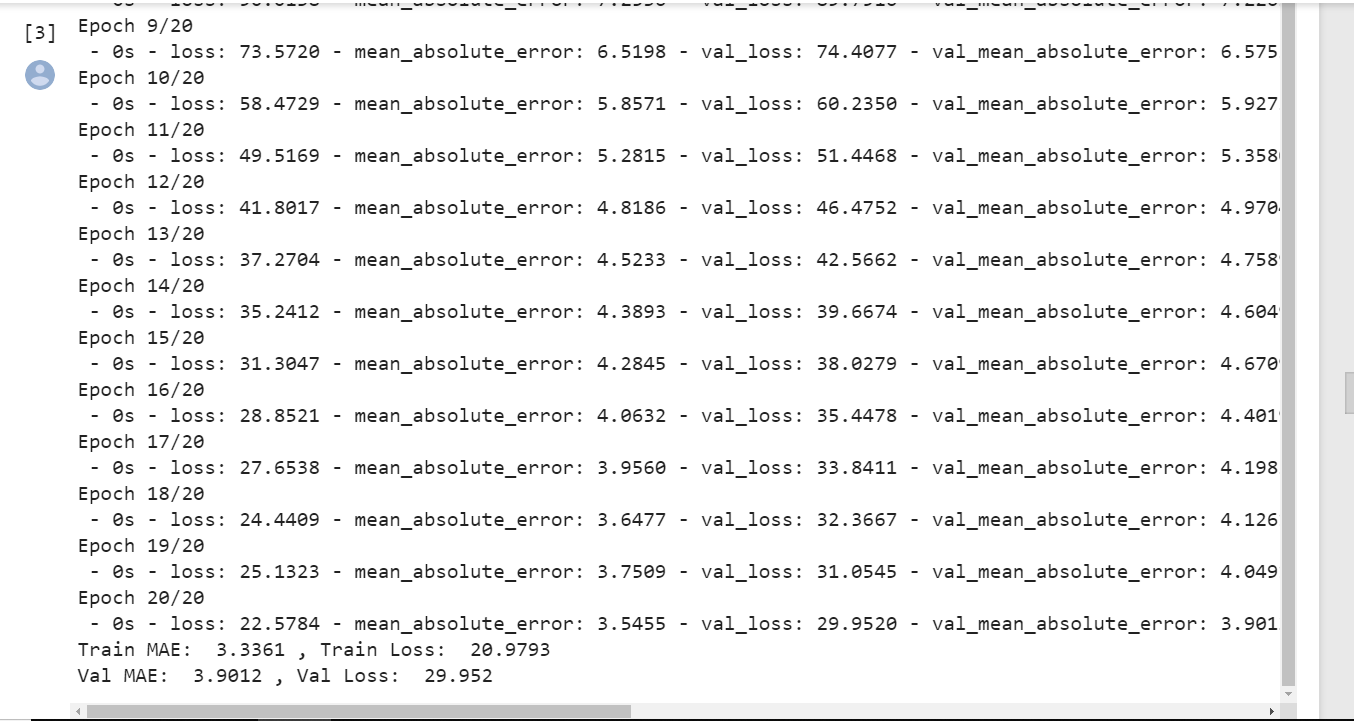
1. *Show the graph in Tensor Board.*
2. *Plot the loss and then change the below parameter and report your view how the result changes in each case*
   * *learning rate*
   * *batch size*
   * *optimizer*
   * *activation function*

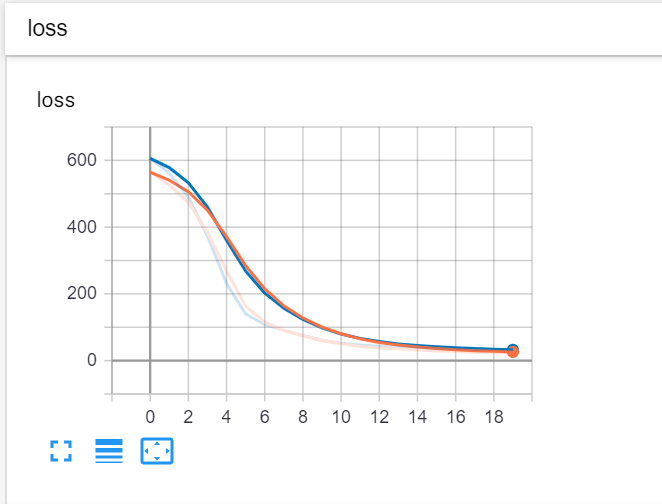
*To implement Linear Regression, we used boston dataset. For various values of learning rate, batch size and optimizers and activation functions loss and accuracy graphs are shown in tensorboard. Optimizers used are SGD, adam and Activation functions used are Sigmoid, Softmax.*

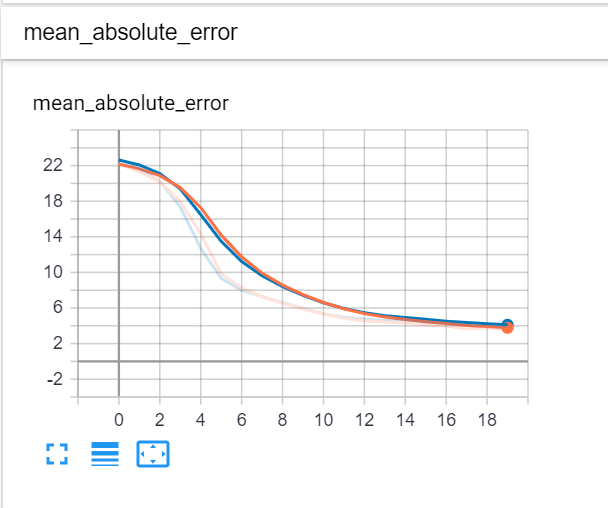
*Code:*



*Output:*





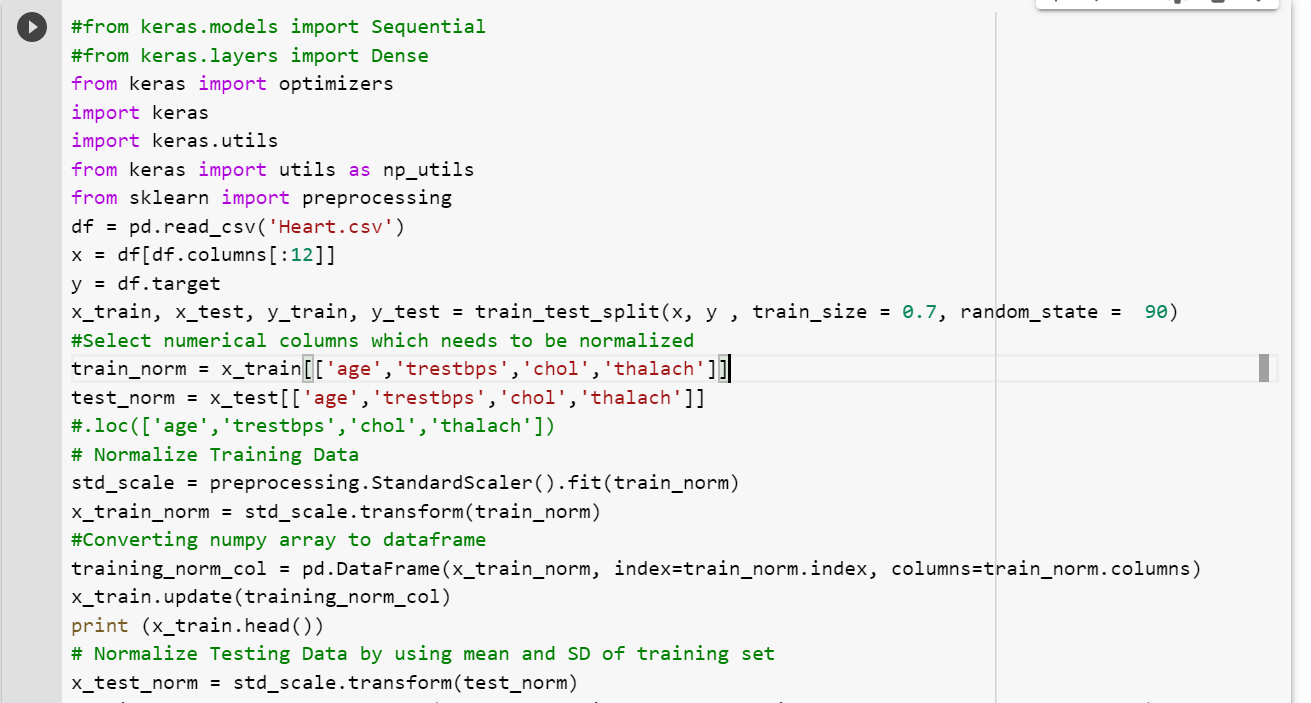


*Task-2:* *Implement the Logistic Regression with any data set of your choice except the datasets being discussed in the class or source code.*

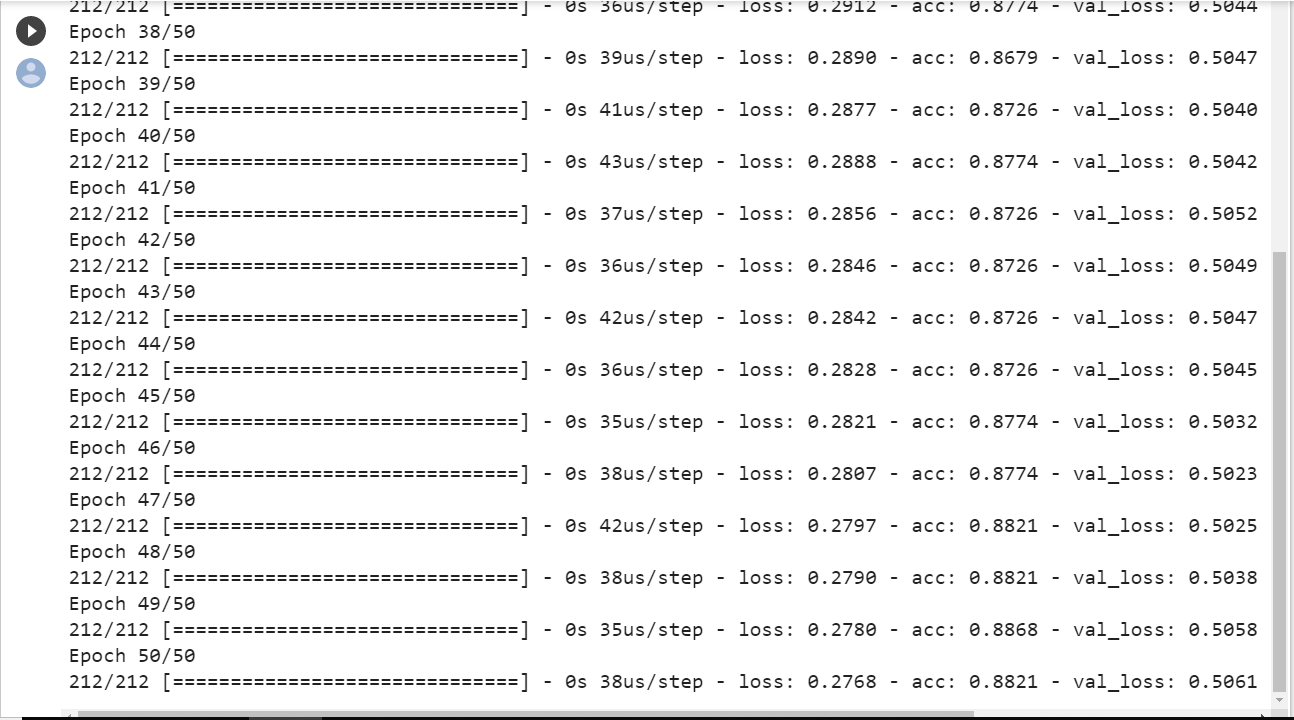
1. *Show the graph in TensorBoard*
2. *Show the Loss in TensorBoard*
3. *Use score=model.evaluate(x\_text,y\_test)and then print(‘test accuracy’, score[1])to print the accuracy*
4. *Change three hyper parameter and report how the accuracy changes*

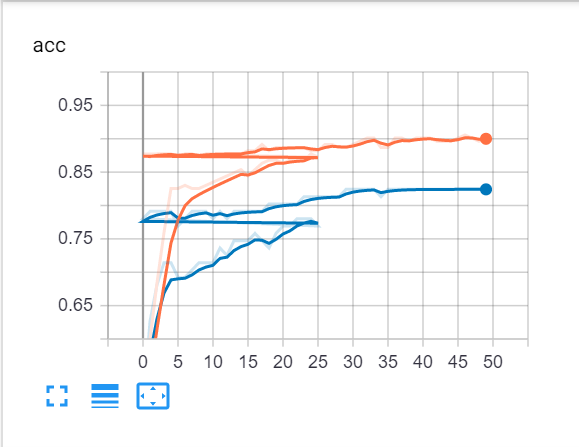
*To Implement Logistic Regression, we used heart dataset. For various values of learning rate, batch size, optimizers and activation functions the accuracy, loss graphs are shown in tensorboard. Optimizers used are SGD, adam and the activation functions used are Sigmoid, Softmax.*

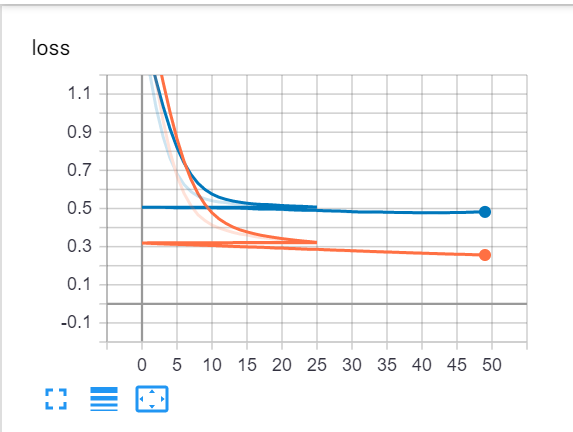
*Code:*



*Output:*



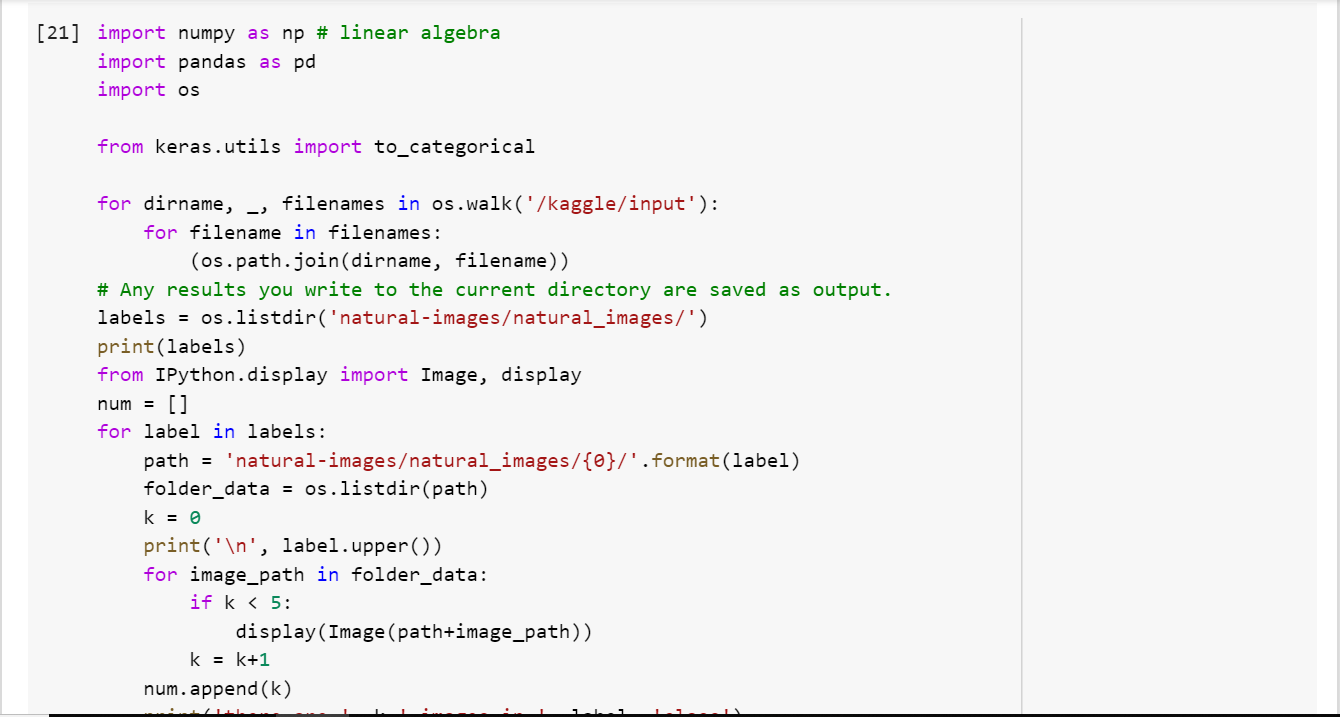


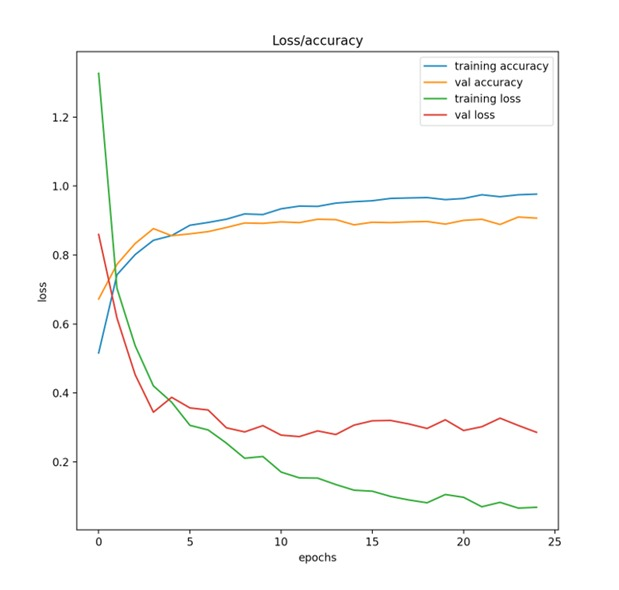


*Task-3:* *Implement the image classification with CNN model on Natural Images dataset*

* *Here we have downloaded the image data set from the Kaggle.*
* *Then normalized the data to the 255 pixels*
* *Then added embedding layer and convolution 2 Dimensional and compiled the model.*
* *Finally fitted the model and plotted the loss and accuracy graphs.*

*Code:*





*Task 4. Implementing text classification by using CNN model on the movie reviews dataset*

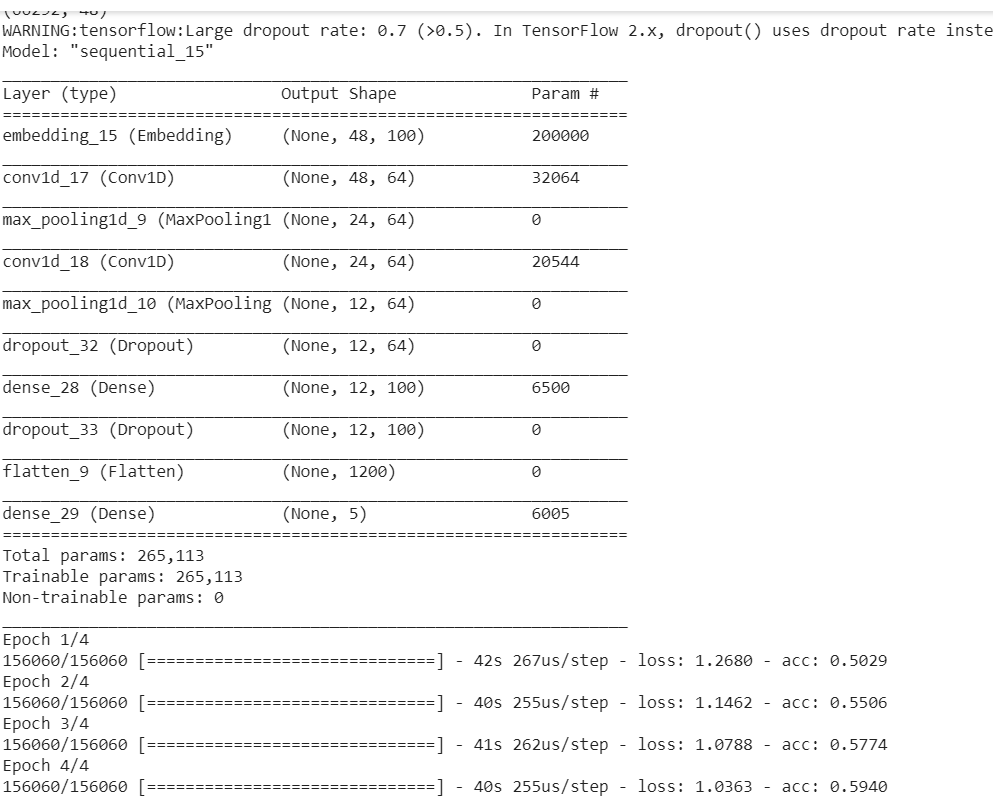
* *To implement CNN model, we have taken the train and test data set from the Kaggle on the movie reviews from the given link to perform sentiment analysis.*
* *Then removed the unnecessary symbols and digits from the phrase column and converted them into the lower case.*
* *Then tokenized the sentences of the phrase column and padded sequences in order to perform convolution.*
* *Then we took a sequential model and have added embedding layers and convolution 1 Dimensional layer.*
* *And flattened the output and added dense layers with the output dimensions.*
* *Compiled fitted the phrase column and then plotted graph for loss and accuracy curve using matplot lib.*

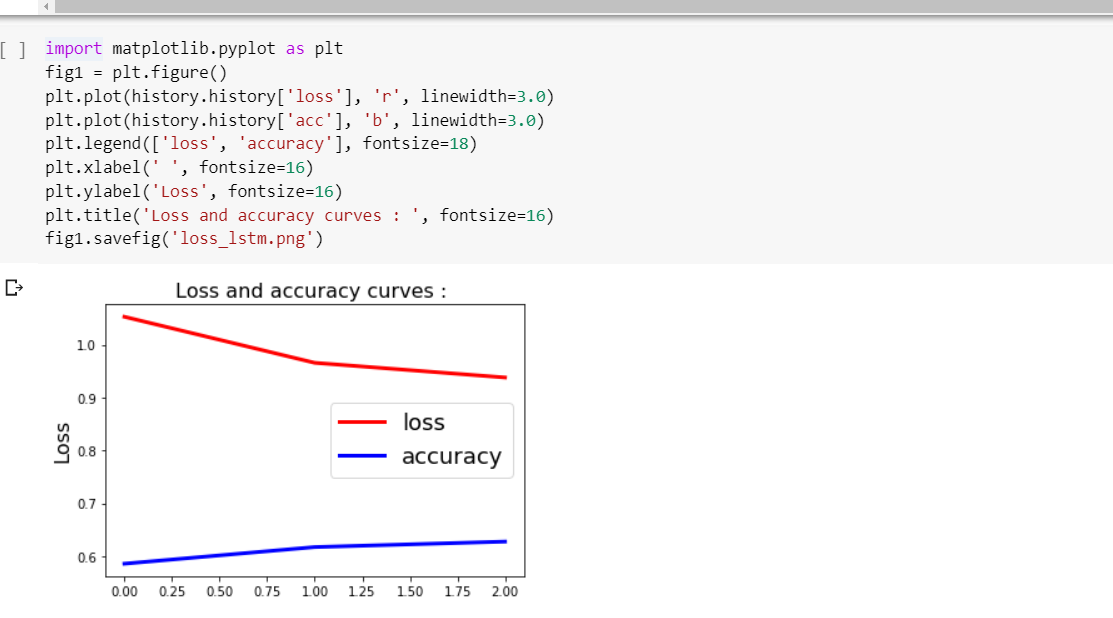
***Code:***





*Output:*





*Task 5. Implementing text classification by using LSTM model on the movie reviews dataset*

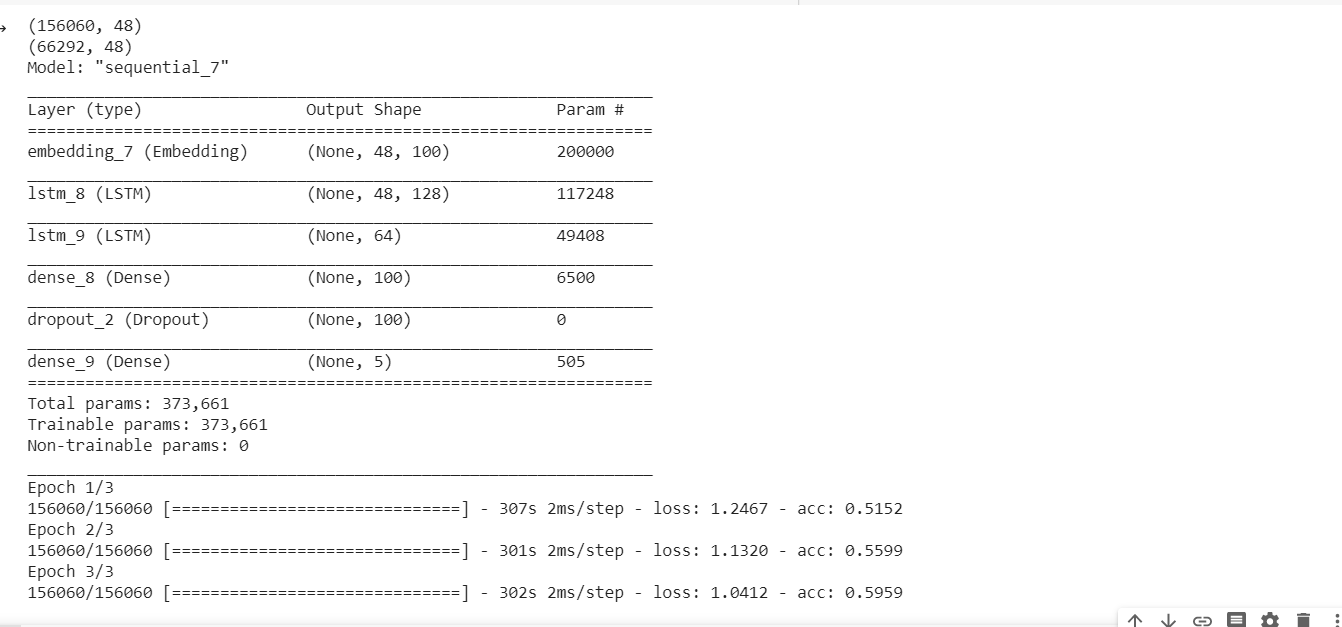
* *To implement LSTM model, we have taken the train and test data set from the Kaggle on the movie reviews from the given link to perform sentiment analysis.*
* *Then we removed the unnecessary symbols and digits from the phrase column and converted them into the lower case.*
* *Then tokenized the sentences of the phrase column and padded sequences in order to perform convolution.*
* *Then we took a sequential model and have added embedding layers and LSTM Dimensional layer.*
* *And flattened the output and added dense layers with the output dimensions.*
* *Compiled fitted the phrase column and then plotted graph for loss and accuracy curve using matplot lib.*

*Code:*





*Output:*

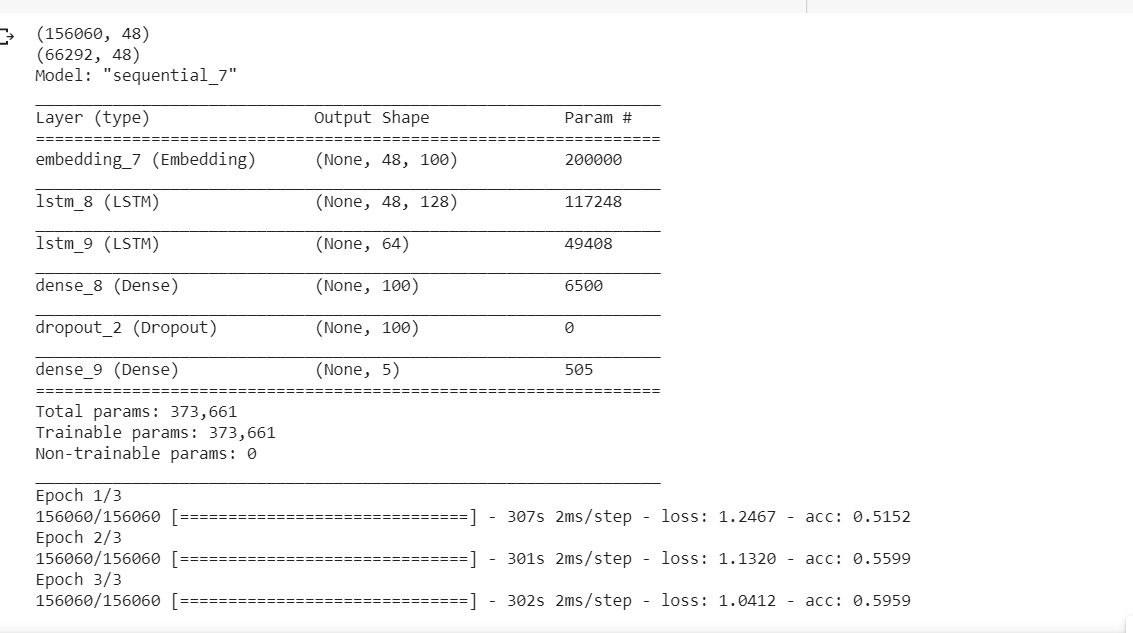




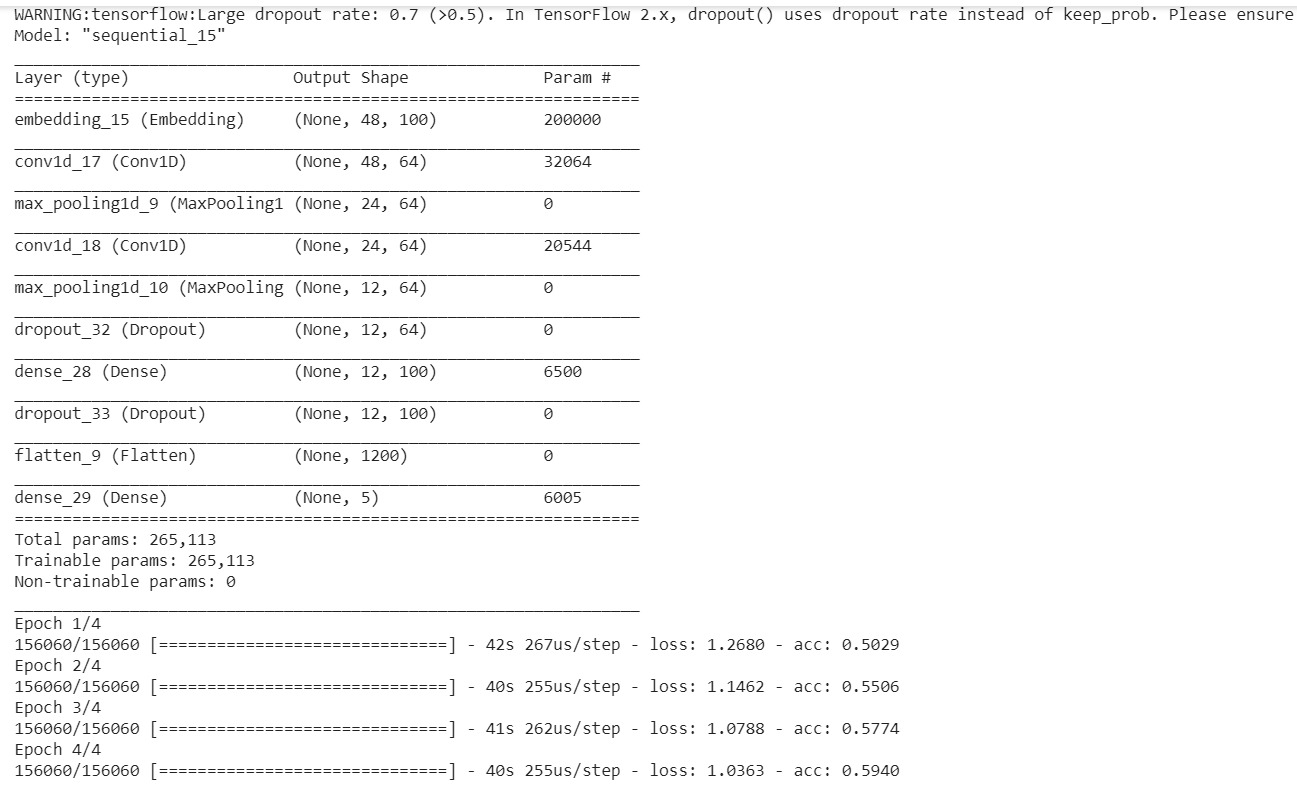
*Task-6. Compare the results of CNN and LSTM models, for the text classification and describe, which model is best for the text classification based on your results. Tune the hyperparameters to attain good accuracy and show the results.*

*The input data in LSTM is one-dimensional which is the reason why it is not suitable for data like images, videos. CNNs are suitable for spatial data, e.g. images.*

*OUTPUT OF CNN :*

**

*OUTPUT OF LSTM :*

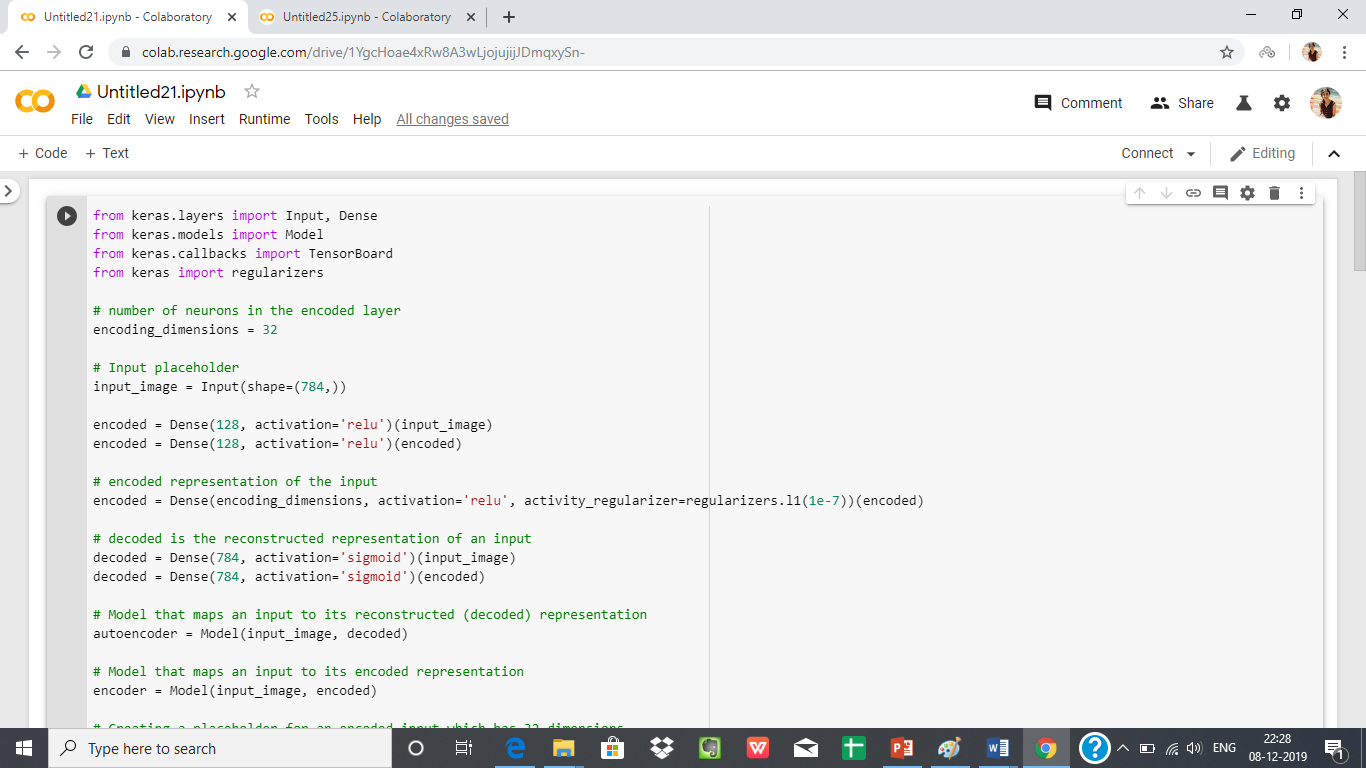
**

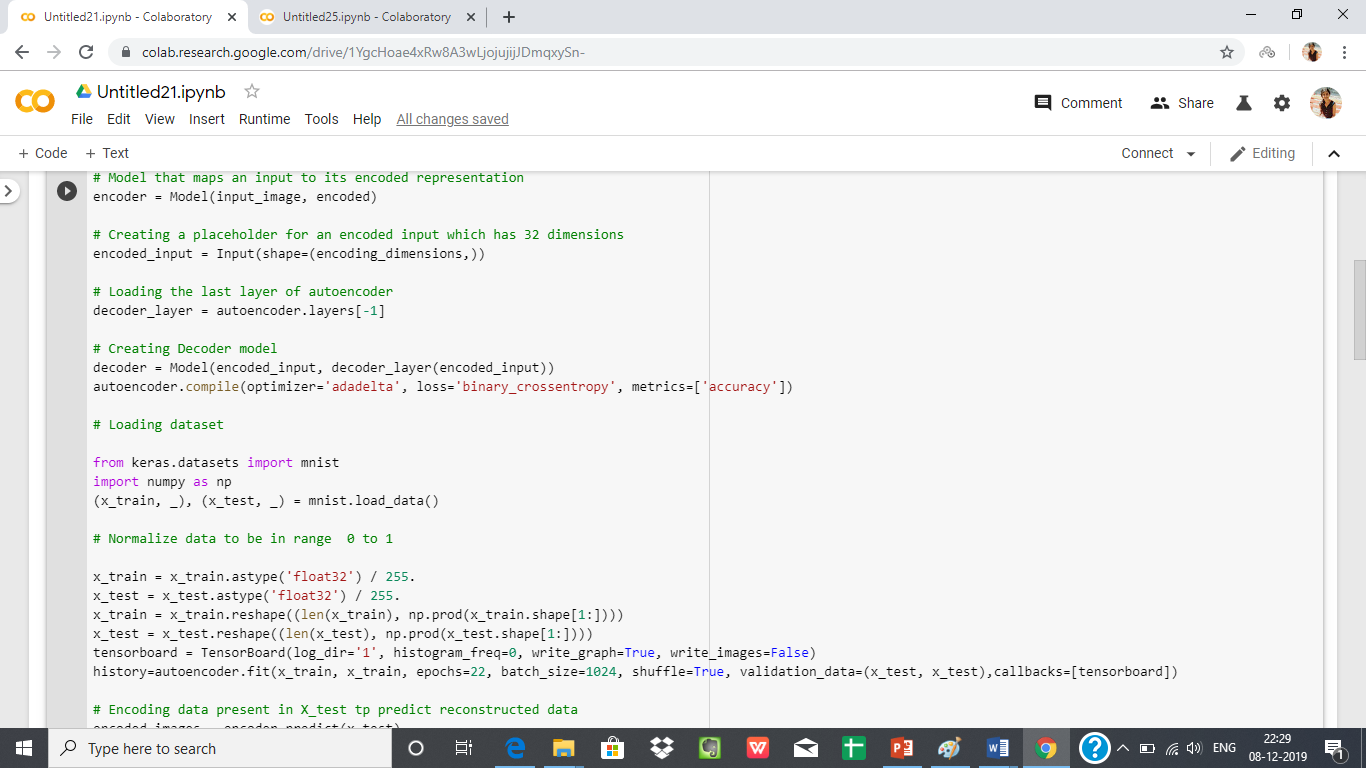
*Here we have executed a few epochs for both CNN and LSTM and got same accuracy for both the neural networks. But in general LSTM dominates CNN in terms of text analysis. Where CNN works well for the image classification and prediction. We have tuned all the parameters like adding more dense layers in the 4 & 5th question itself.*

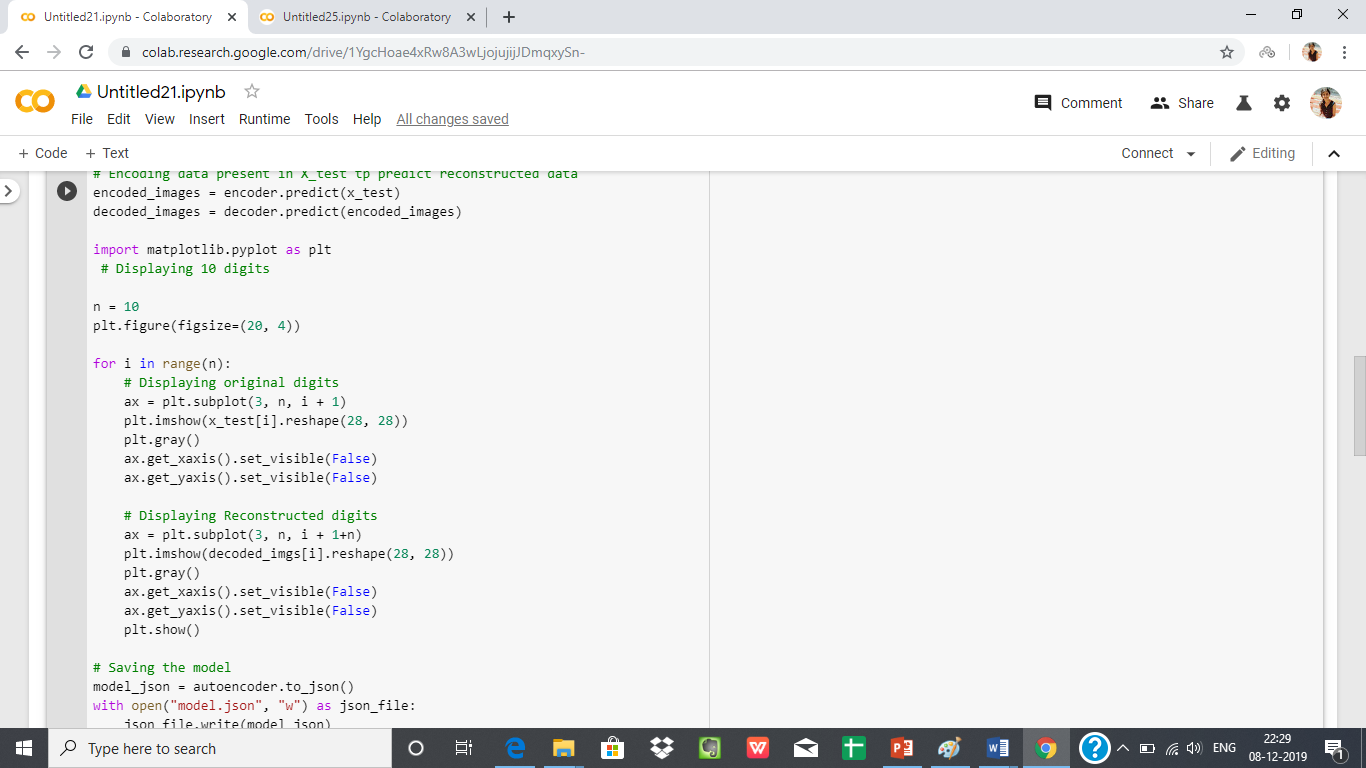
*Task-7. Apply Autoencoders on MNIST dataset and show the encoding and decoding on a particular image. Make sure you document each and every line of the code.*

*Autoencoders reconstruct the existing image, which will contain most important features of the image. The dataset used here is MNIST. Activation functions used in encoder and decoder are relu and sigmoid respectively. Autoencoder has 32 dimesnions whereas the input to encoder and the reconstructed images has 784 (28\*28) dimensions. The data has been normalized to avoid high distance variance. Encoded and Decoded images of 10 numbers are printed. Autoencoding is a lossy reconstructions since redundant features are eliminated.*

*CODE :*







*OUTPUT:*

