**SAPTARSHI DAS**

**JU IT 4TH YR 1ST SEM**

**ROLL-001811001090**

**MACHINE LEARNING LAB**

**ASSIGNMENT 3**

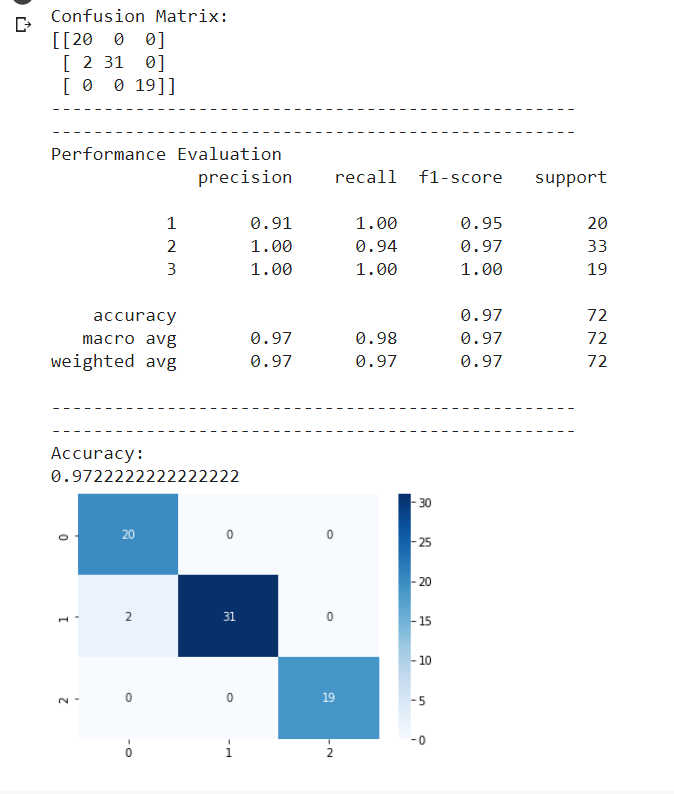
**GITHUB LINK:**

<https://github.com/SaptarshiDas7/Machine-Learning-Assignments/tree/main/Assignment%203>

**PART 1**

**1) Wine Dataset**

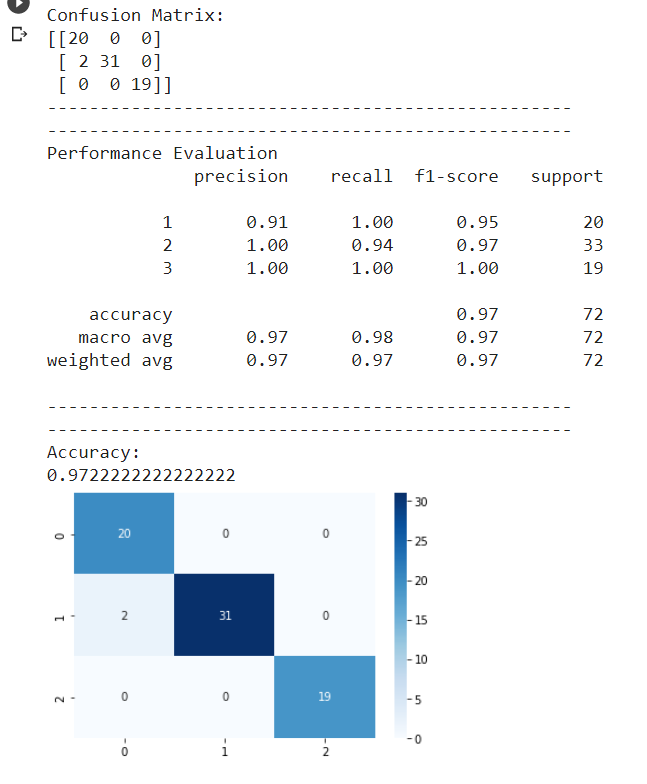
**1.1) GaussianHMM Without Tuning**

****

**1.2) GaussianHMM With Tuning**

****

**1.3) GMMHMM Without Tuning**

****

**1.4) GMMHMM With Tuning**

****

**1.5) MultinomialHMM Without Tuning**

****

**1.6) MultinomialHMM Without Tuning**

****

**The maximum accuracy was achieved when the Train-Test split ratio was 70:30, which was achieved by using the Gaussian Model. The maximum range of accuracies was achieved by the Gaussian Model, followed by the GMMHMM model, which is followed by the MultinomialHMM model.**

**2) Ionosphere Dataset**

**2.1) GaussianHMM Without Tuning**

****

**2.2) GaussianHMM With Tuning**

****

**2.3) GMMHMM Without Tuning**

****

**2.4) GMMHMM With Tuning**

****

**2.5) MultinomialHMM Without Tuning**

****

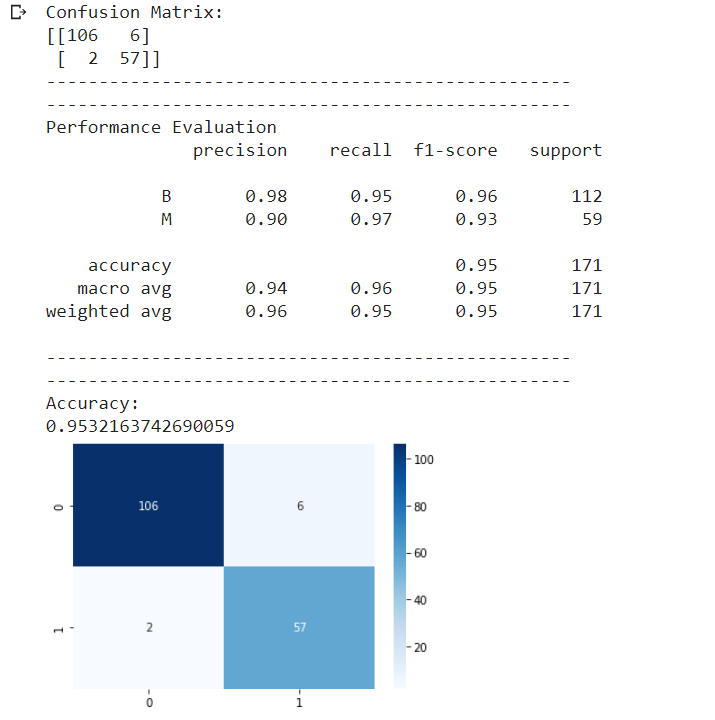
**2.6) MultinomialHMM Without Tuning**

****

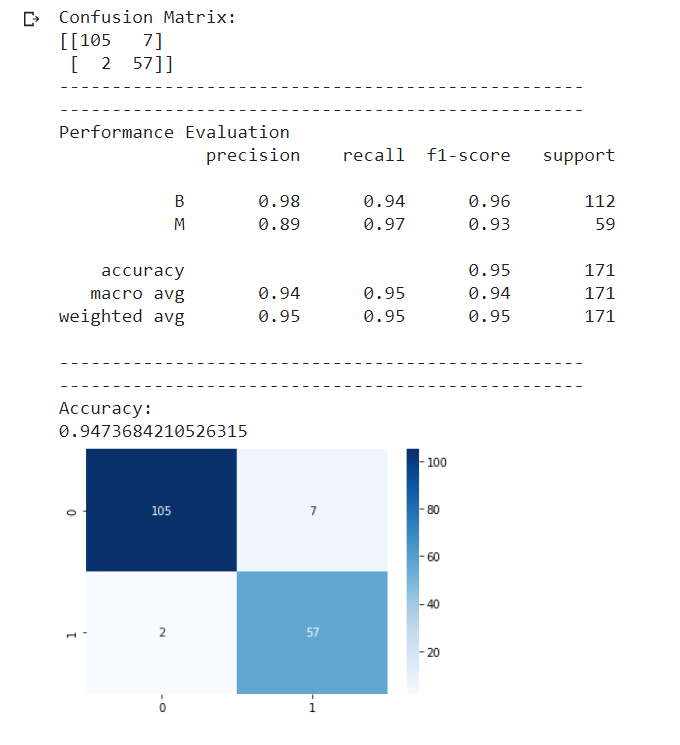
**The maximum accuracy was achieved when the Train-Test split ratio was 70:30, which was achieved by using the Gaussian Model. The maximum range of accuracies was achieved by the Gaussian Model, followed by the GMMHMM model, which is followed by the MultinomialHMM model.**

**3) Breast Cancer Dataset**

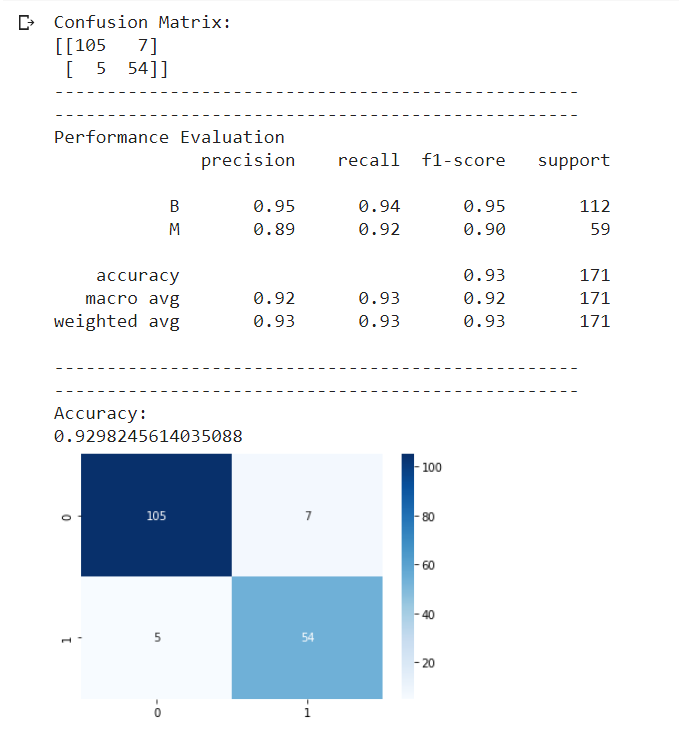
**3.1) GaussianHMM Without Tuning**

****

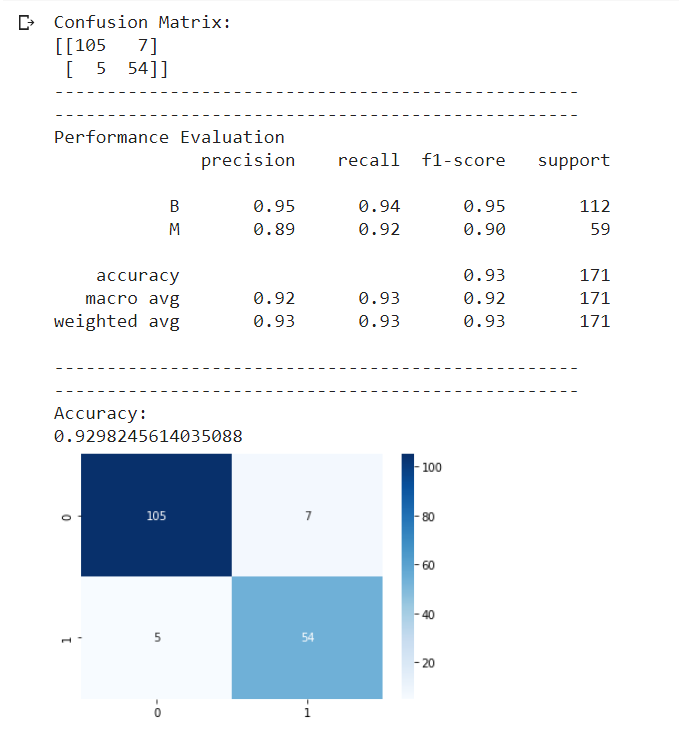
**3.2) GaussianHMM With Tuning**

****

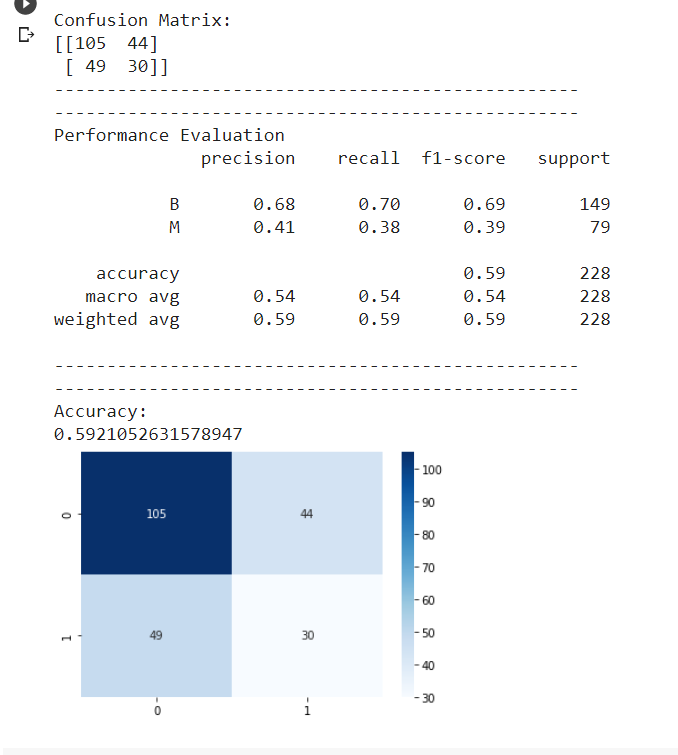
**3.3) GMMHMM Without Tuning**

****

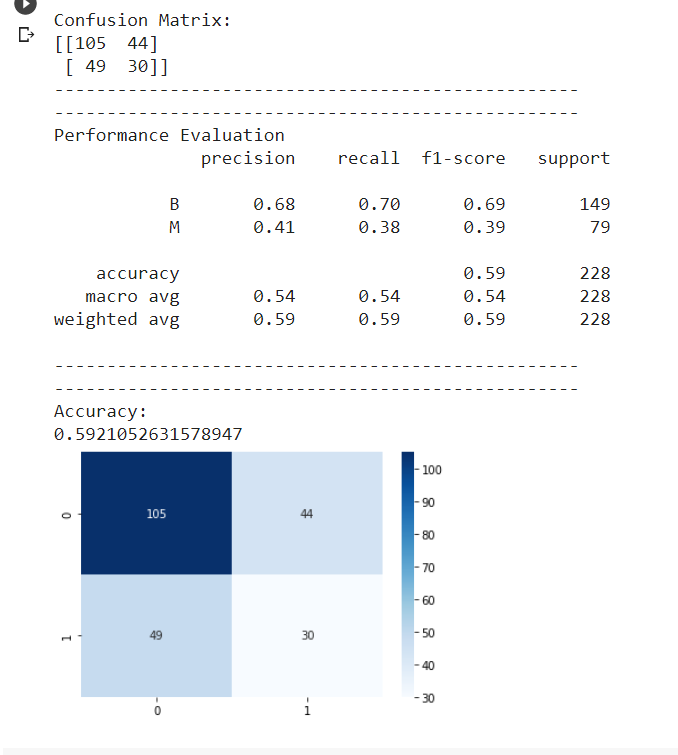
**3.4) GMMHMM With Tuning**

****

**3.5) MultinomialHMM Without Tuning**

****

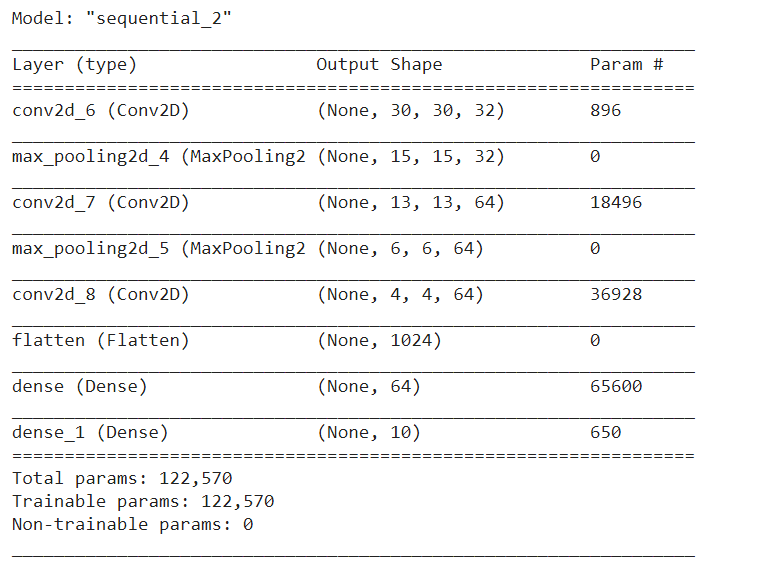
**3.6) MultinomialHMM Without Tuning**

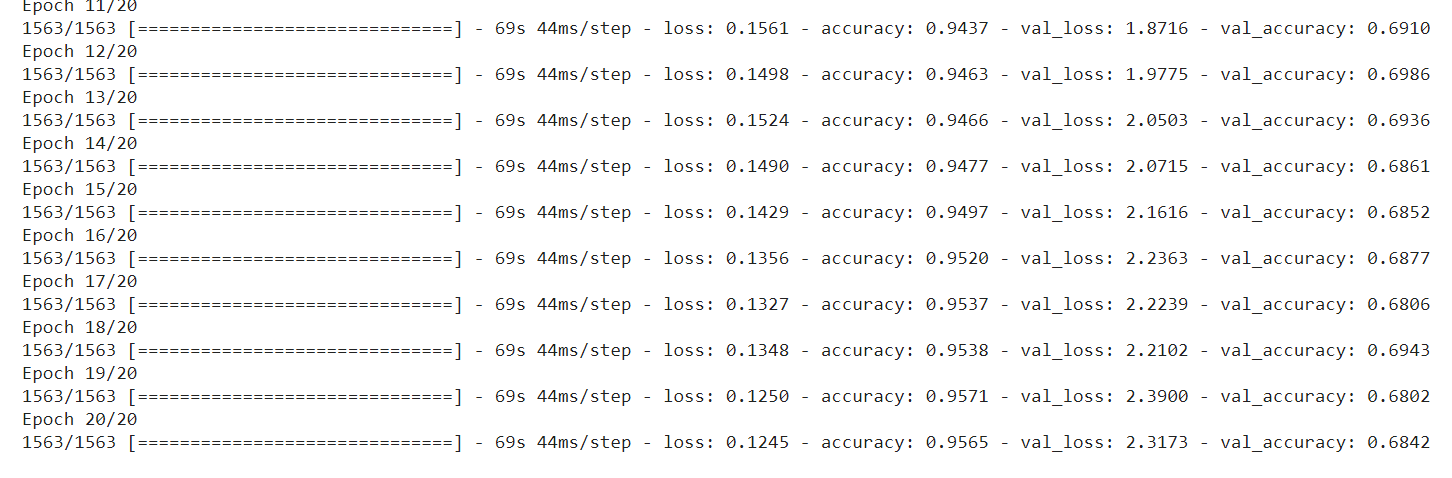
****

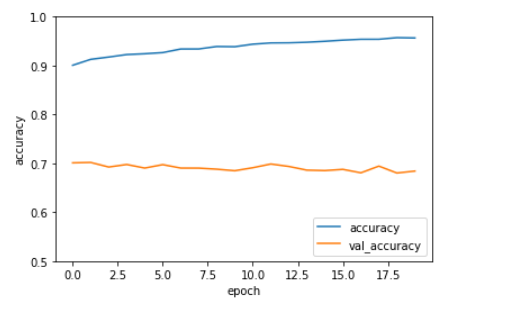
**The maximum accuracy was achieved when the Train-Test split ratio was 70:30, which was achieved by using the Gaussian Model. The maximum range of accuracies was achieved by the Gaussian Model, followed by the GMMHMM model, which is followed by the MultinomialHMM model.**

**PART 2**

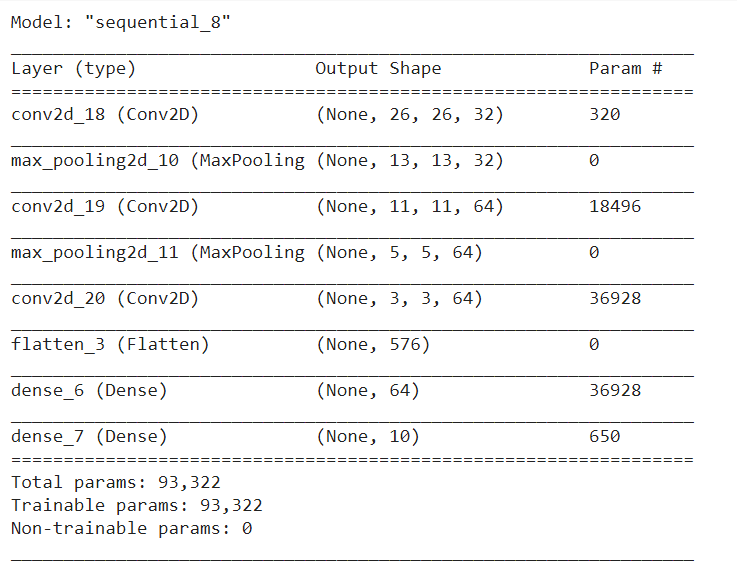
**1) CIFAR-10**

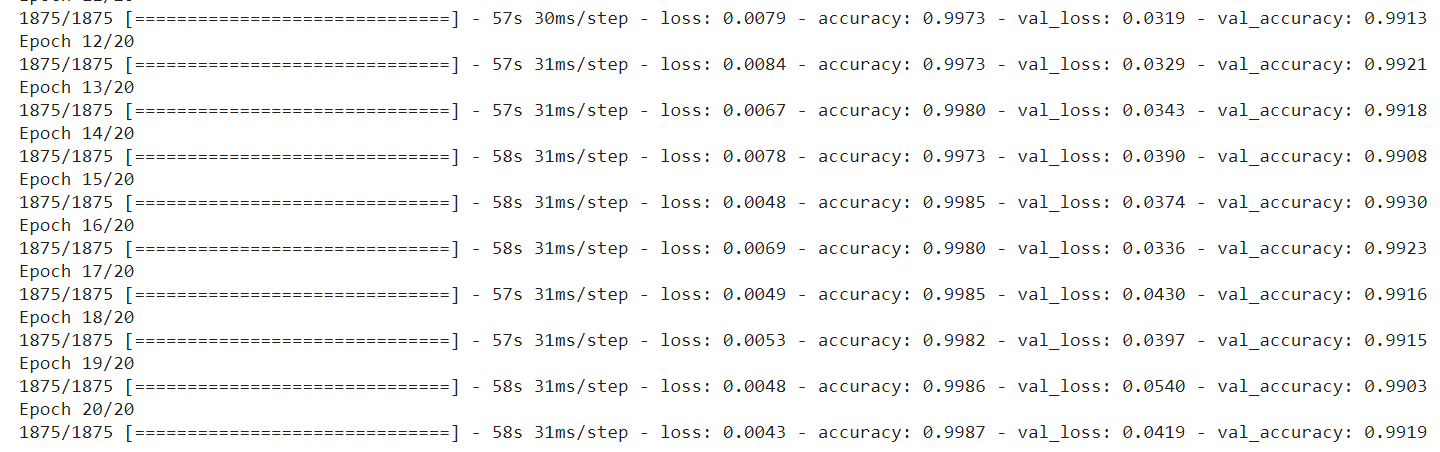
****

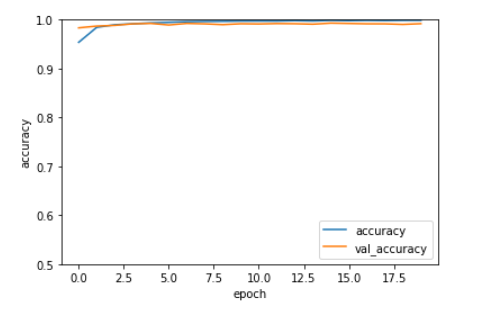
****

****

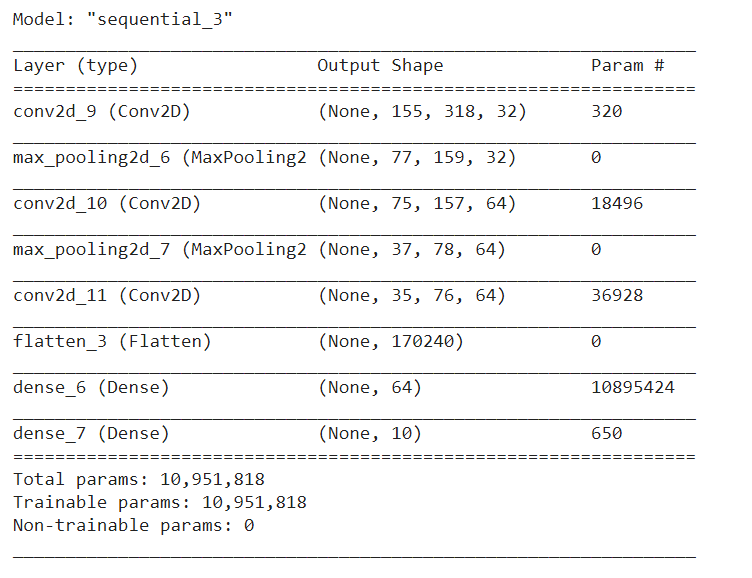
**2) MNIST**

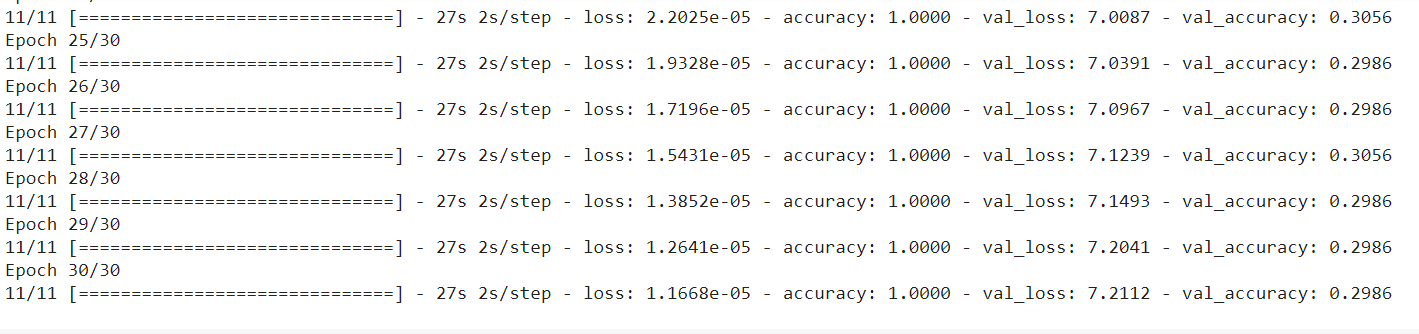
****

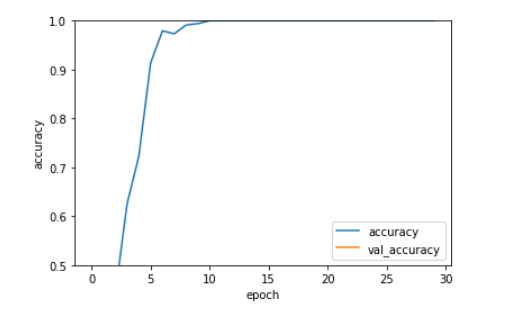
****

****

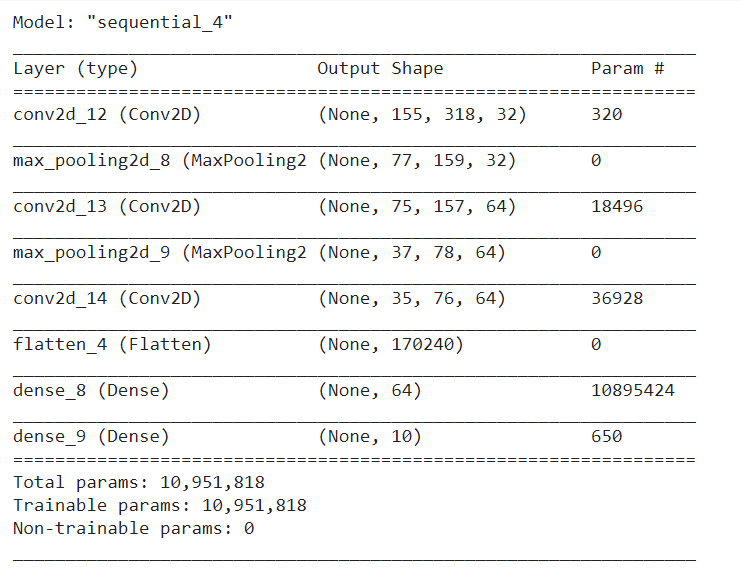
**3) SAVEE**

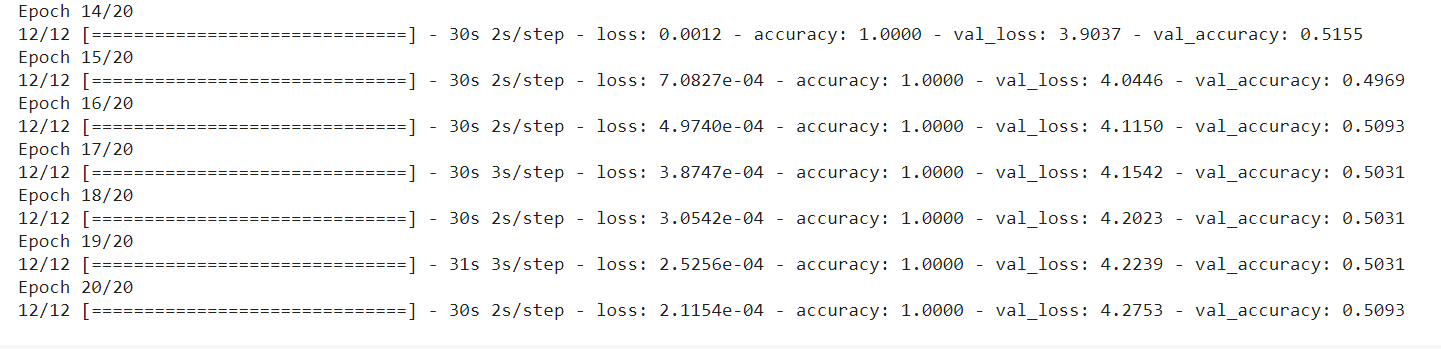
****

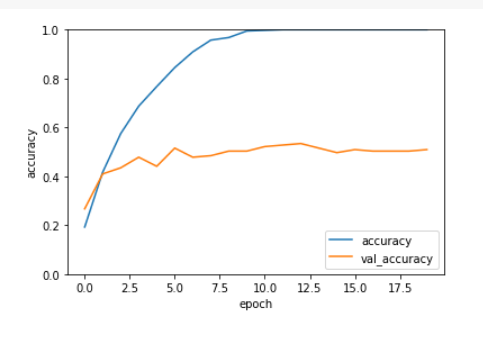
****

****

**4) EmoDB**

****

****

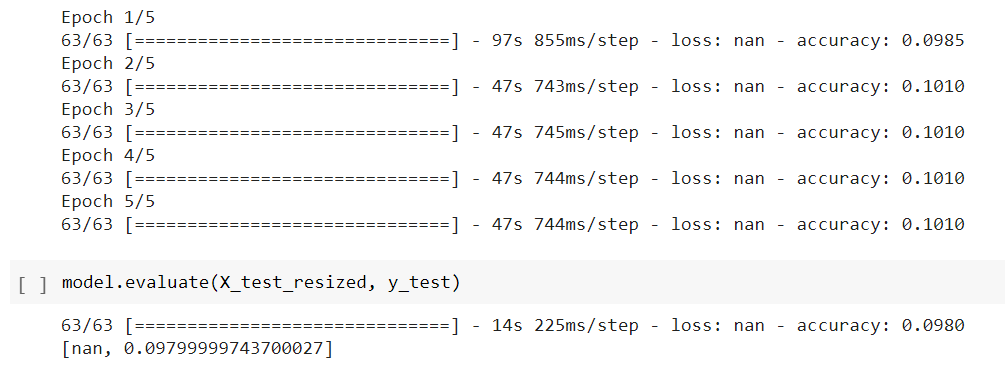
****

**It was observed that the more layers we add the higher accuracy we can achieve. At the same time, if we keep on adding more layers, the final accuracy will saturate. Also, the number of convolution and the pooling layers play an important role in training the model.**

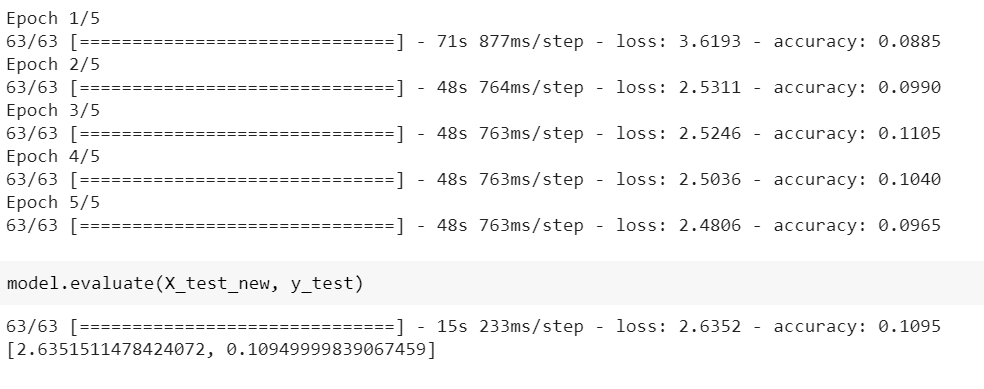
**PART 3**

**1) VGG-16**

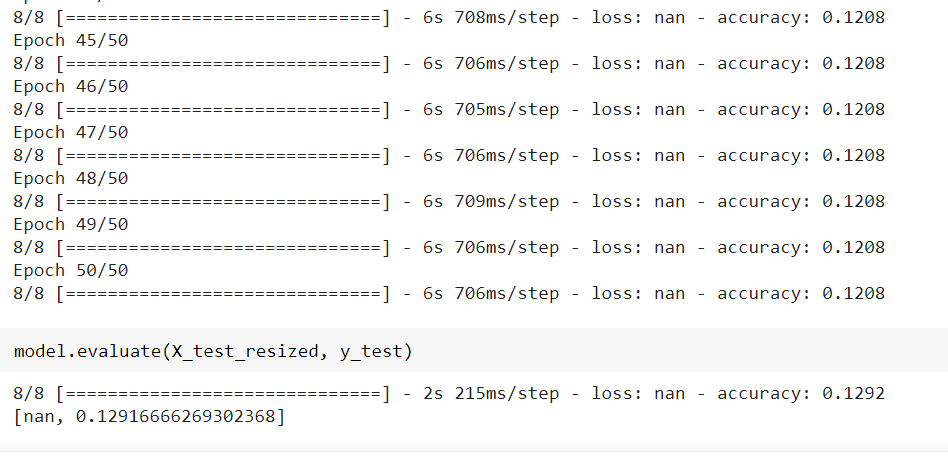
**1.1) CIFAR-10**

****

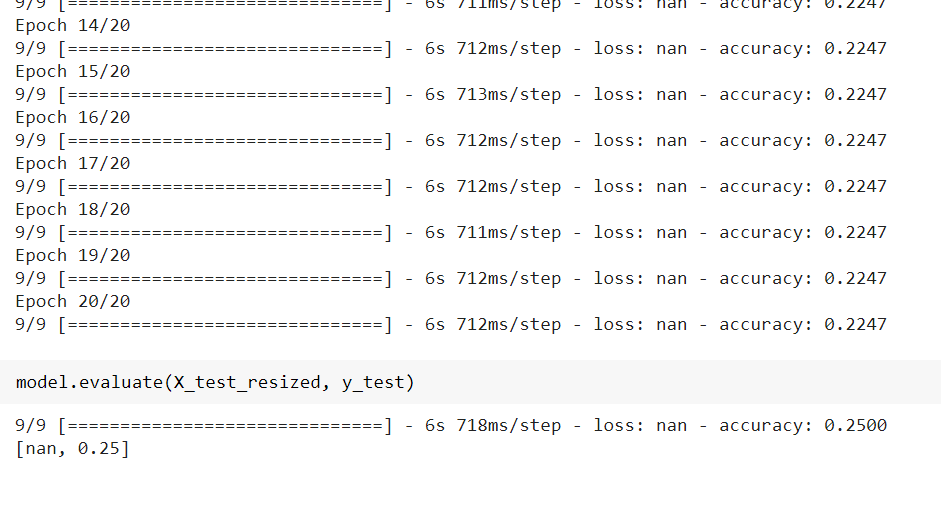
**1.2) MNIST**

****

**1.3) SAVEE**

****

**1.4) EmoDB**

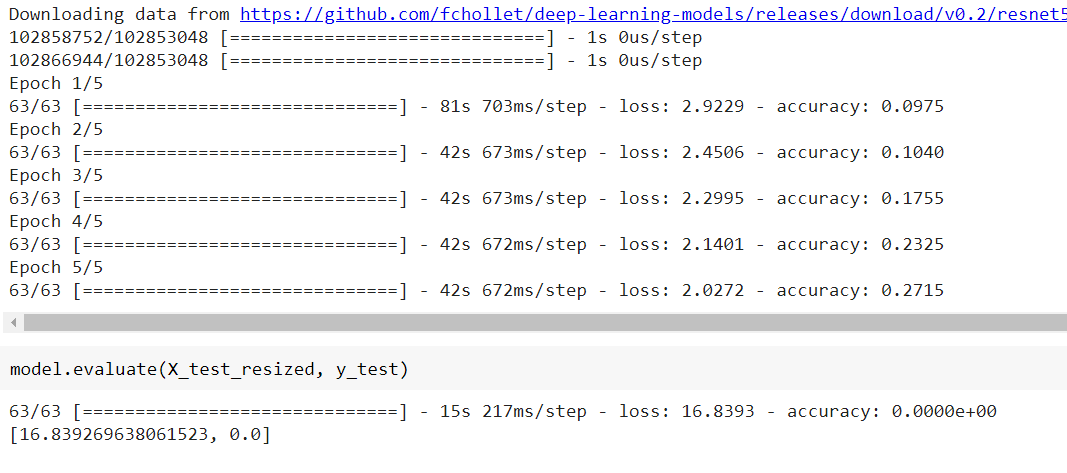
****

**The entire model can be broken down into 5 blocks, where each block contains 3 convolution and 1 max-pooling layers.**

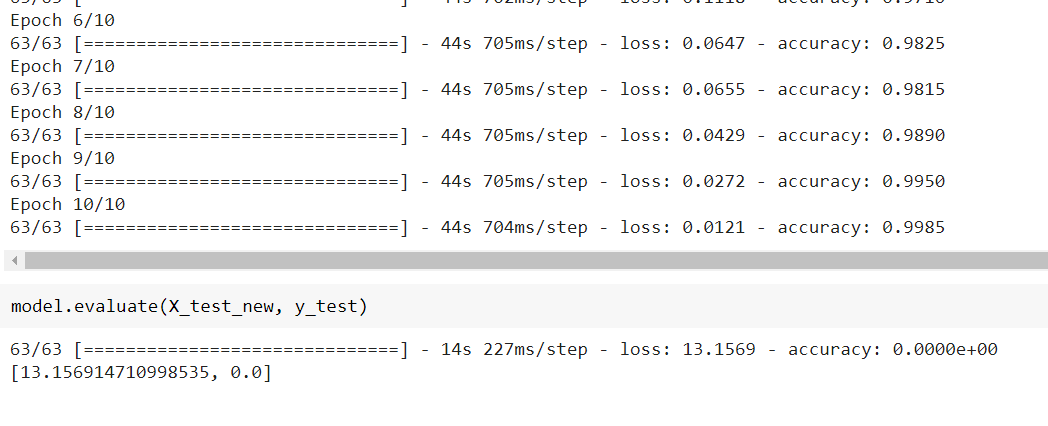
**Looking at the complexity of the model and the limitations of google colab, I have reduced the input size for the model,i.e., i have taken 2000 training data points and 2000 testing data points.**

**2) ResNet-50**

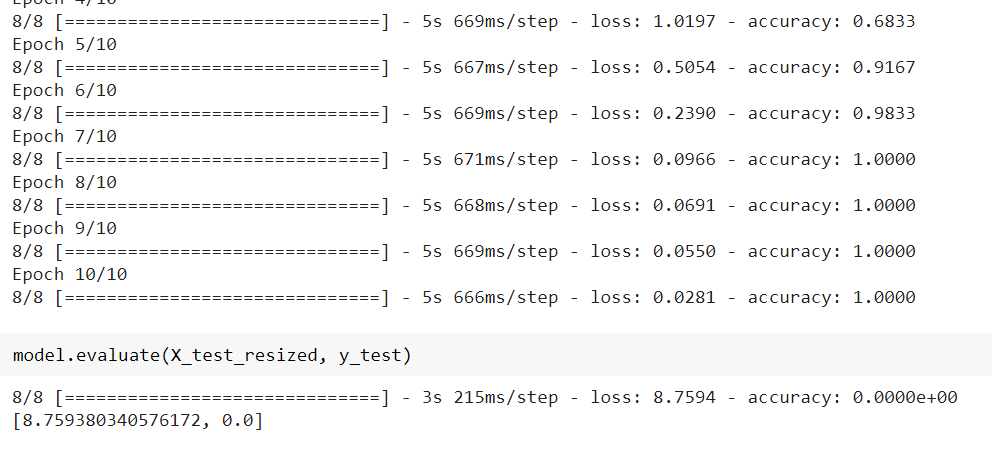
**2.1) CIFAR-10**

****

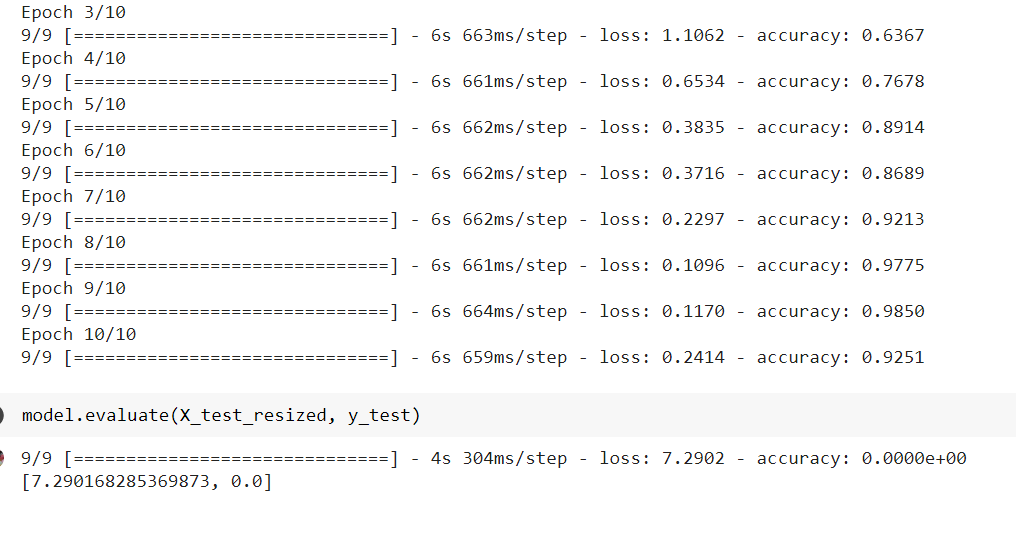
**2.2) MNIST**

****

**2.3) SAVEE**

****

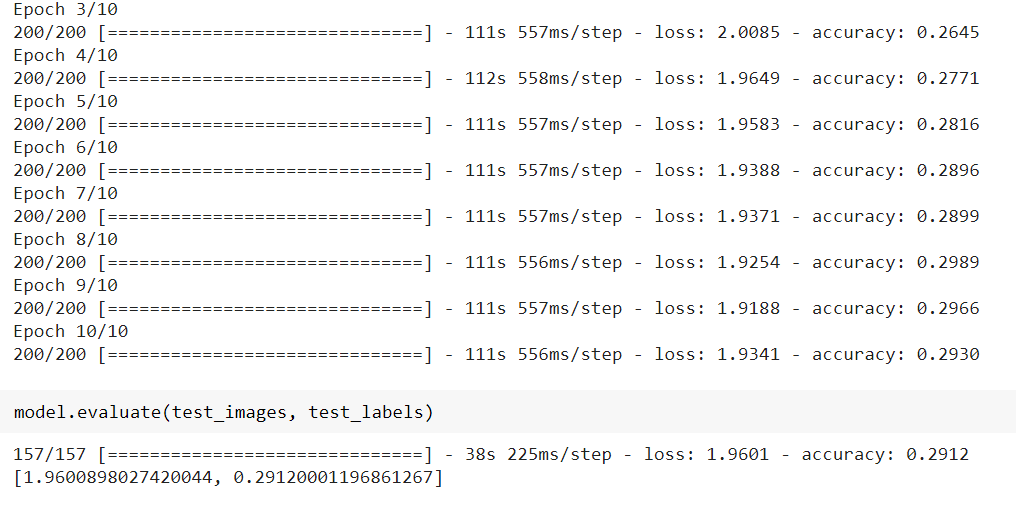
**2.4) EmoDB**

****

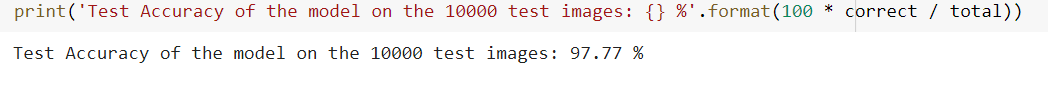
**Looking at the complexity of the model and the limitations of google colab, I have reduced the input size for the model,i.e., I have taken 2000 training data points and 2000 testing data points.**

**3) Recurrent Neural Networks (RNN)**

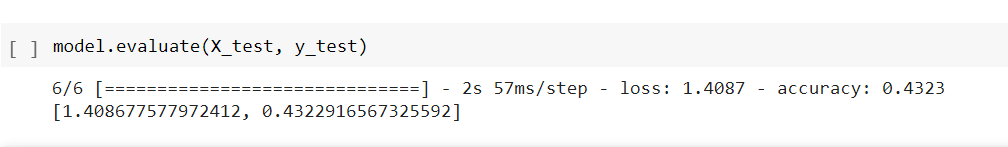
**3.1) CIFAR-10**

****

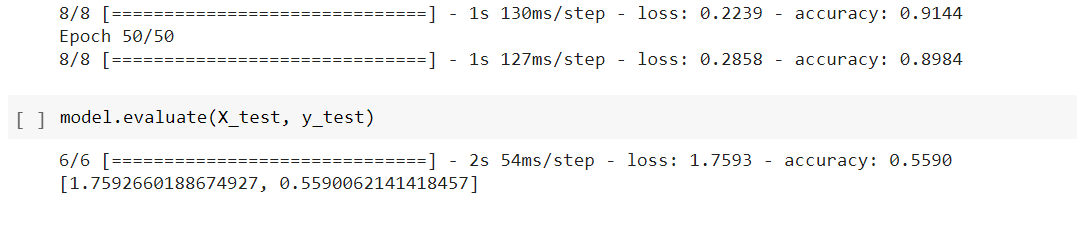
**3.2) MNIST**

****

**3.3) SAVEE**

****

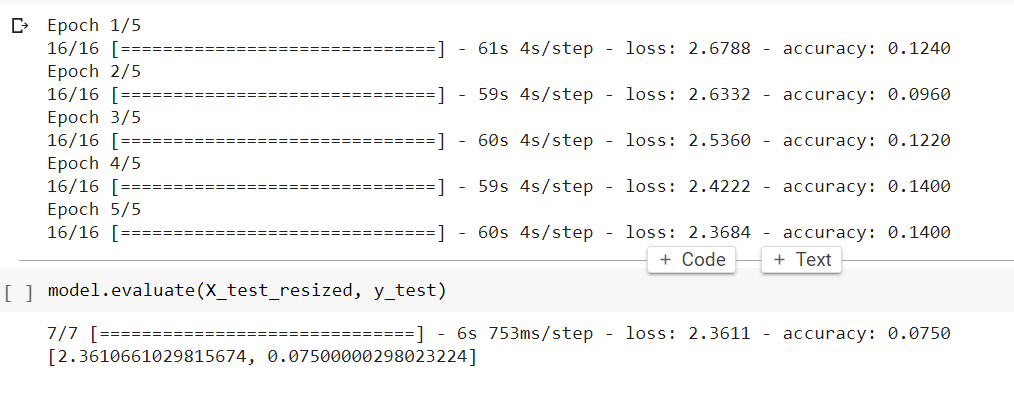
**3.4) EmoDB**

****

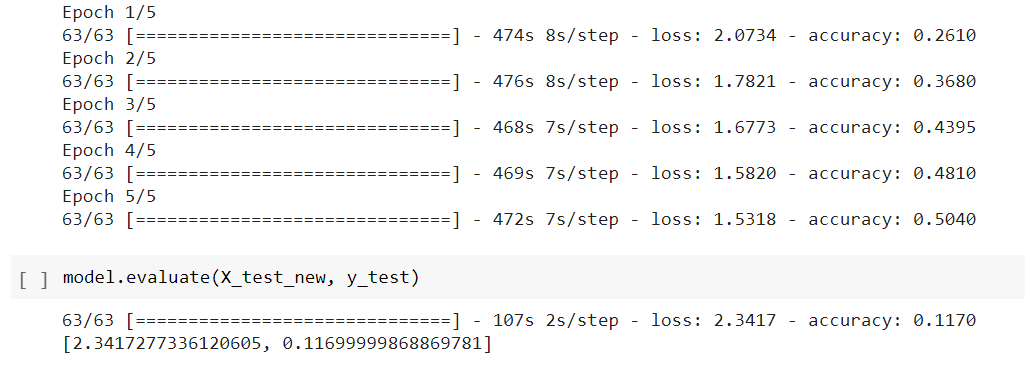
**Looking at the complexity of the model and the limitations of google colab, I have reduced the input size for the model,i.e., I have taken 2000 training data points and 2000 testing data points.**

**4) AlexNet**

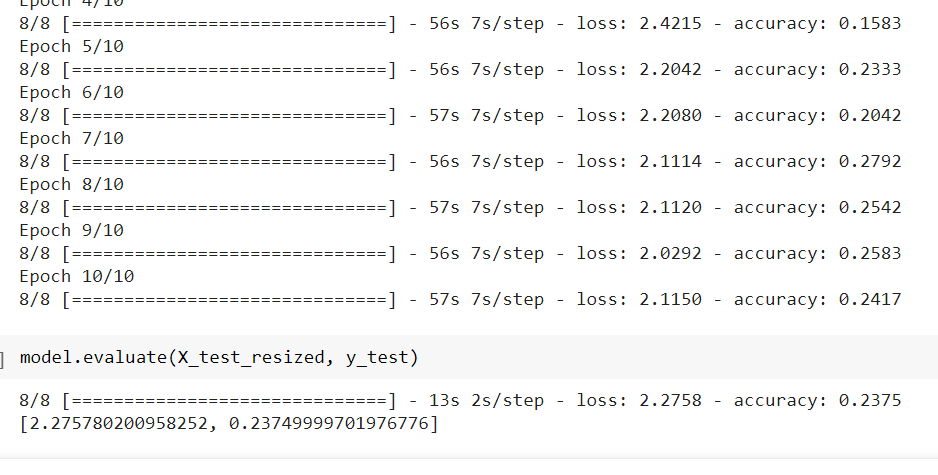
**4.1) CIFAR-10**

****

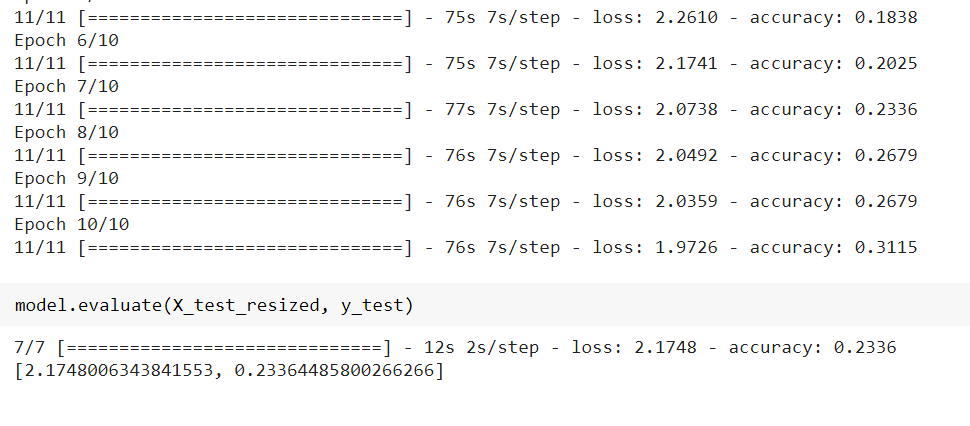
**4.2) MNIST**

****

**4.3) SAVEE**

****

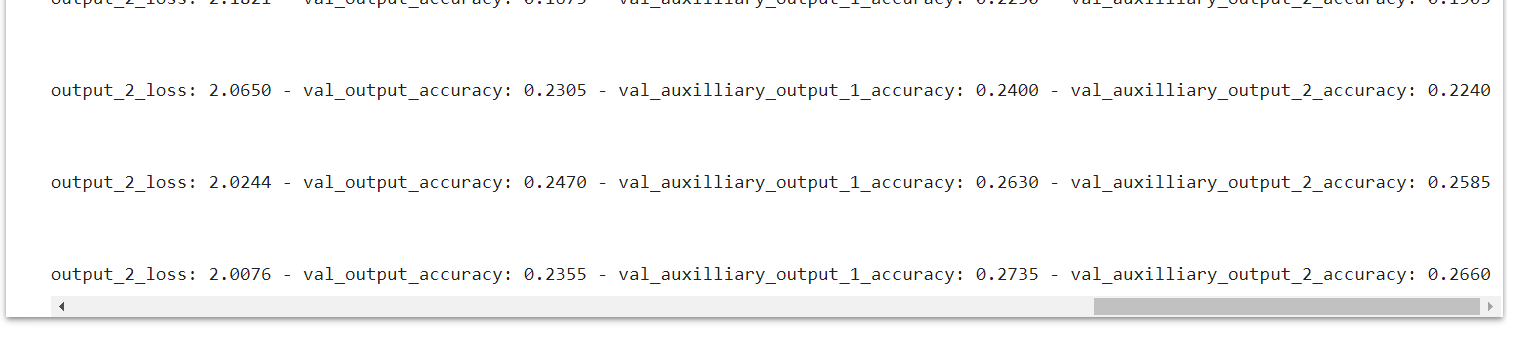
**4.4) EmoDB**

****

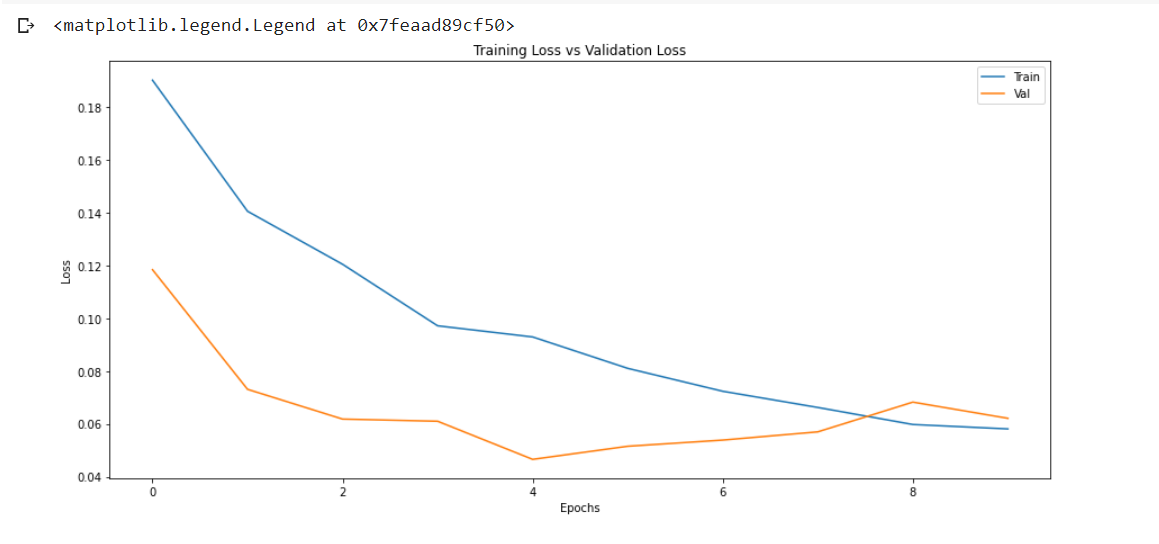
**Looking at the complexity of the model and the limitations of google colab, I have reduced the input size for the model,i.e., I have taken 2000 training data points and 2000 testing data points.**

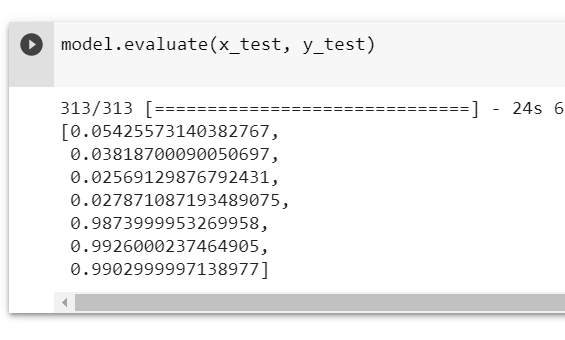
**5) GoogLeNet**

**5.1) CIFAR-10**

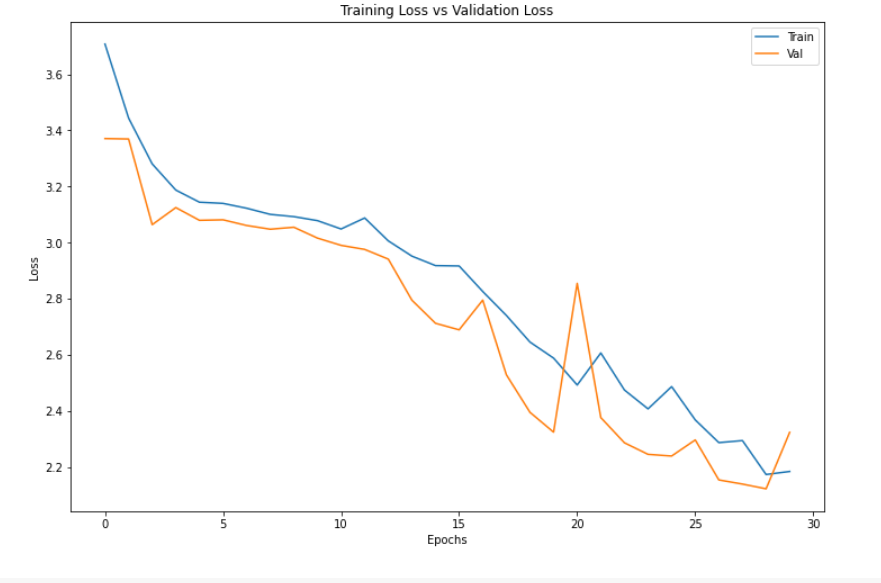
****

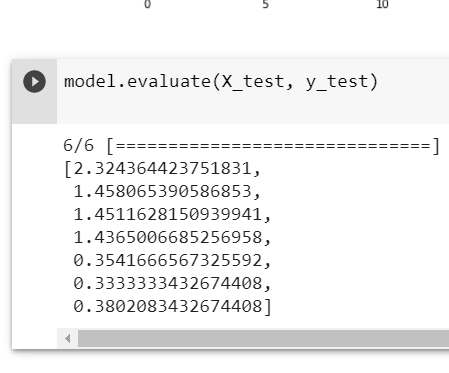
**5.2) MNIST**

****

****

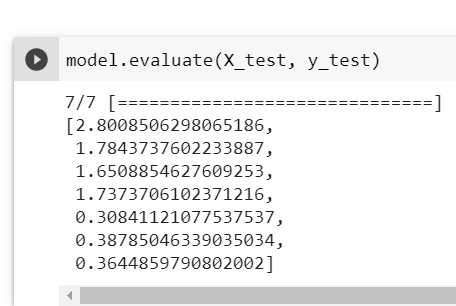
**5.3) SAVEE**

****

****

**5.4) EmoDB**

****

****

**Looking at the complexity of the model and the limitations of google colab, I have reduced the input size for the model,i.e., I have taken 2000 training data points and 2000 testing data points.**