# LAB 2

## Q1. Upload the Backprop.ipynb to Jupyter notebook (or google colab) and see if you can understand the code. Increase the number of iterations (epochs) and see whether it improves the prediction accuracy.

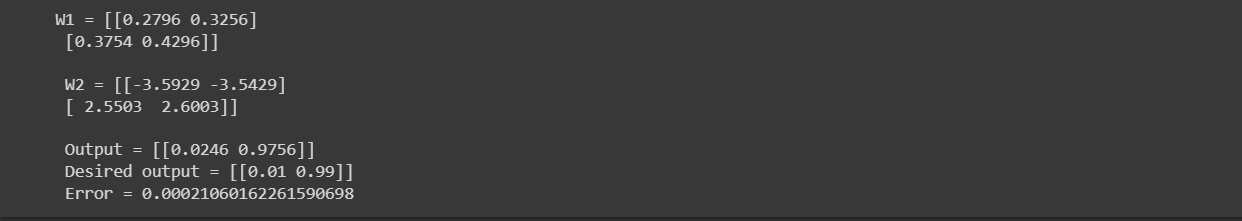
No\_of\_iter = 100

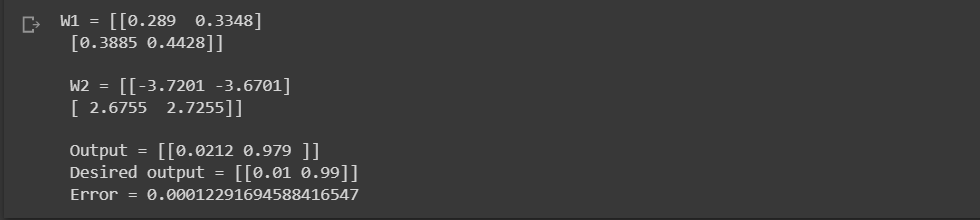
No\_of\_iter = 150



No\_of\_iter = 500



No\_of\_iter = 3500

No\_of\_iter = 5000

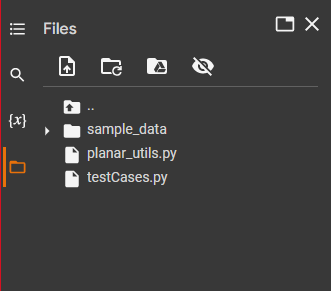
* **When we are increasing the number of iterations the output gets closed to desired output and the error getting decreased.**

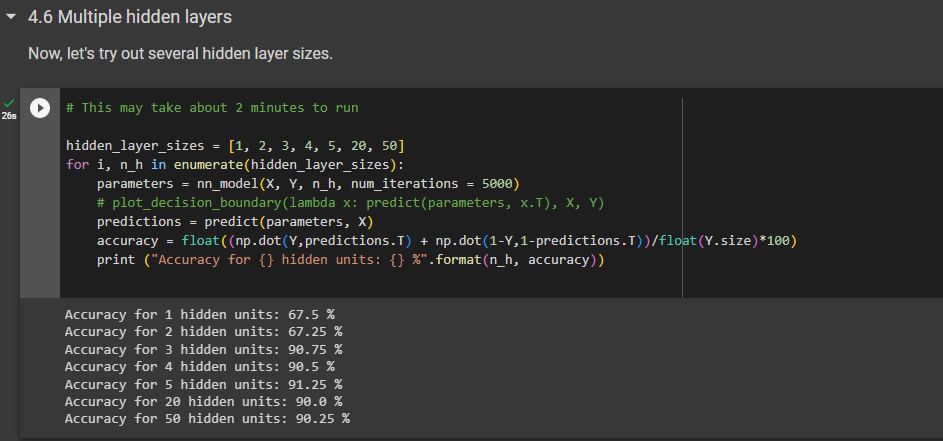
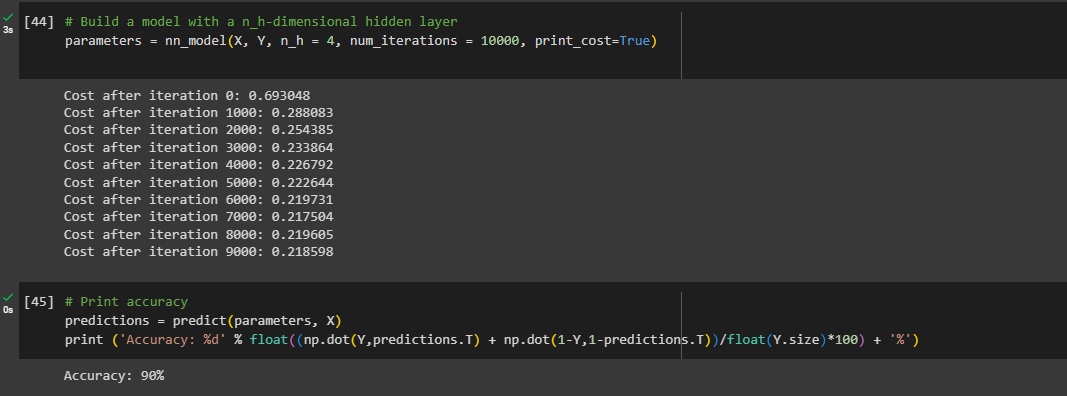
## Q2. Upload the NN\_sample.ipynb to Jupyter notebook (or google colab) and see if you can understand the code. Add the following text cell and the code cell to the notebook and run it again.

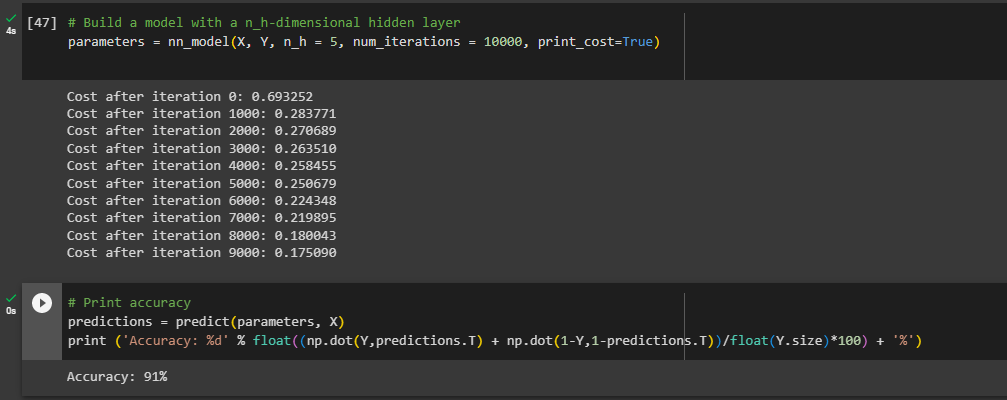
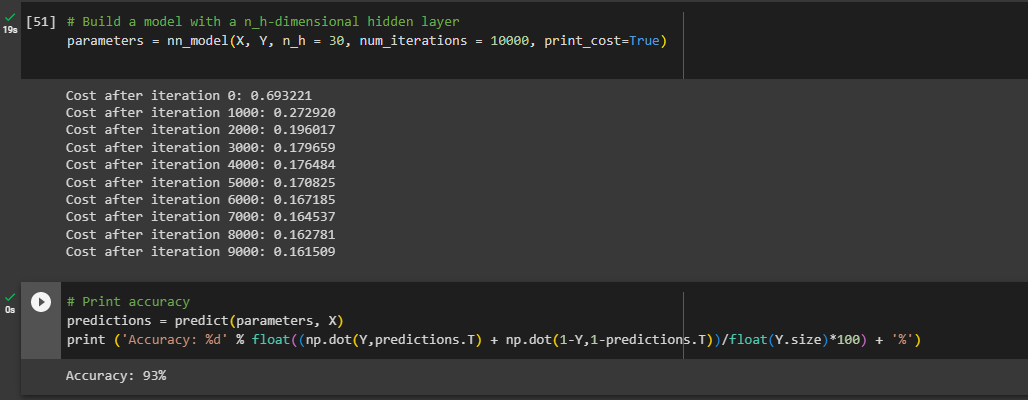
## 

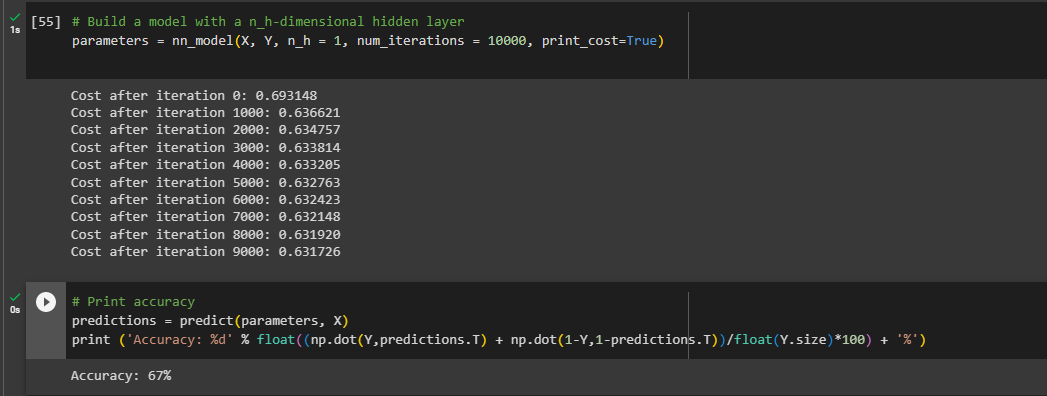
### What happens when the number of hidden nodes increase?

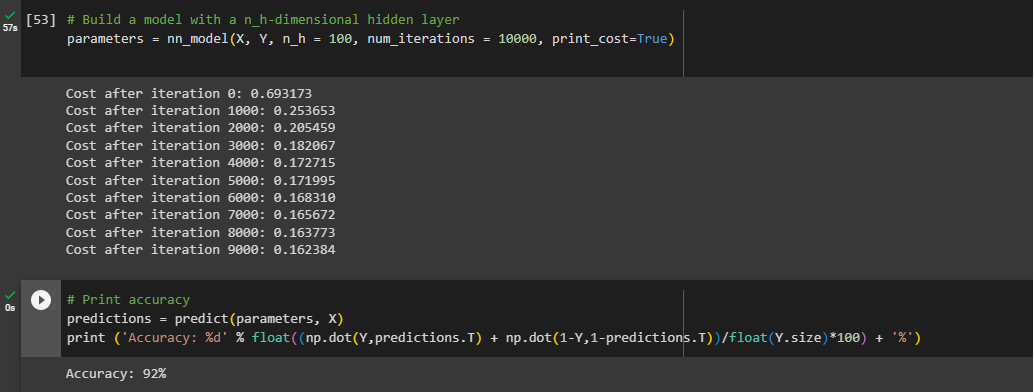
As the number of hidden nodes increases, the accuracy increases.











### Can you explain the pattern of accuracy when the hidden nodes increase?

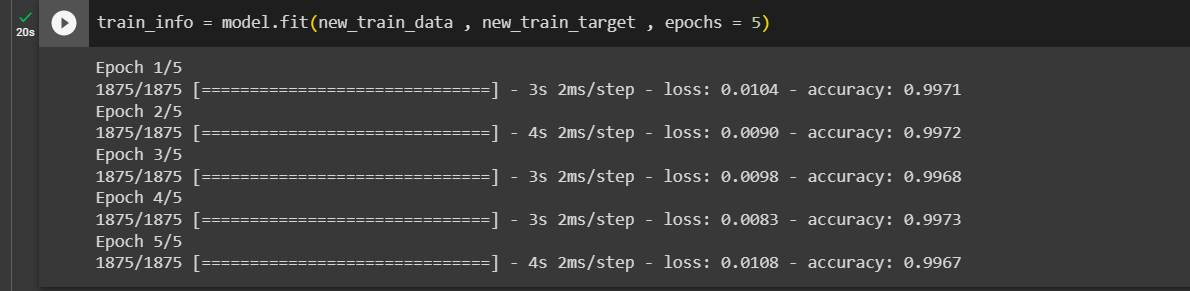
When the number of hidden nodes is reduced to two, the accuracy is 67.25%. Also, when the number of hidden nodes is 5, 10, 20,50, the accuracy increases to 92.0%. Even if the number of hidden nodes is increased to about 100, the accuracy remains at 93%. Finally, it became like constant.

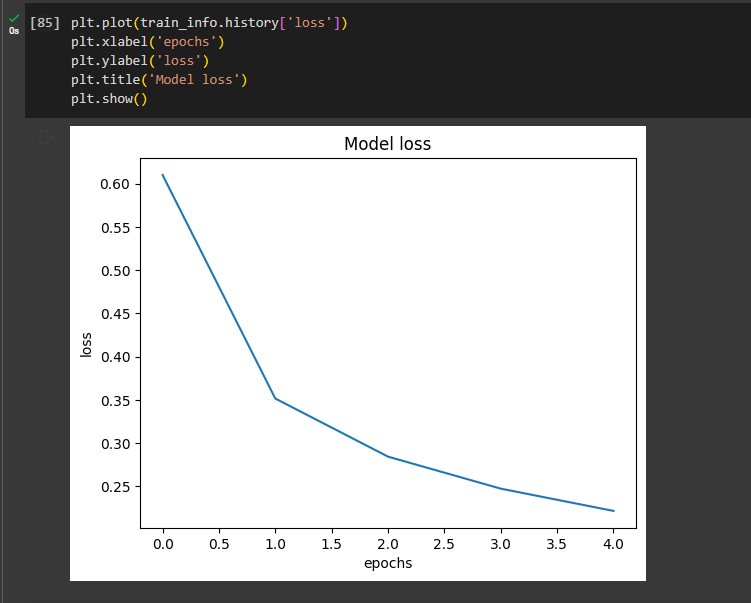
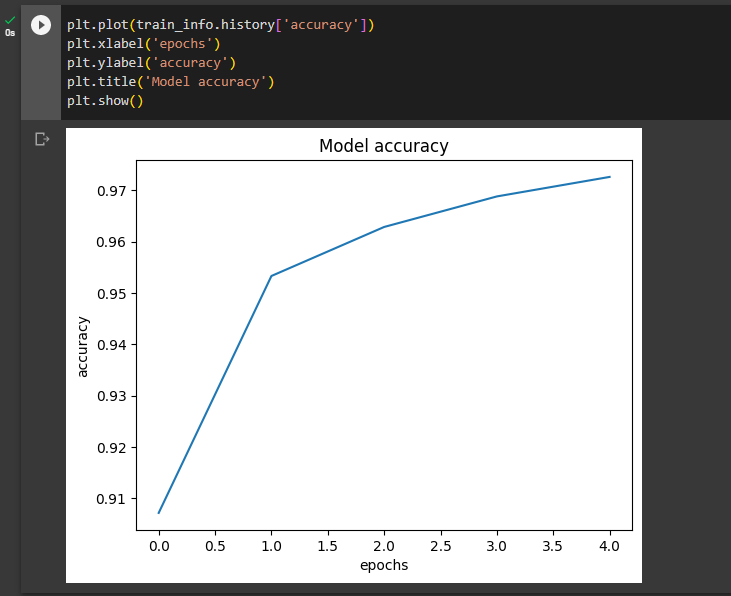
## Q3. Run the MLP\_with\_MNIST\_dataset.ipynb using Jupyter notebook (or google colab) and see if you can understand the code.

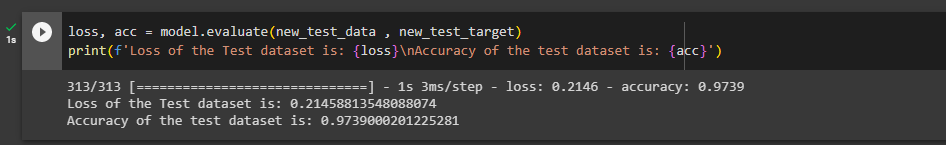
### Improve the test accuracy of the model by changing the hyperparameters.

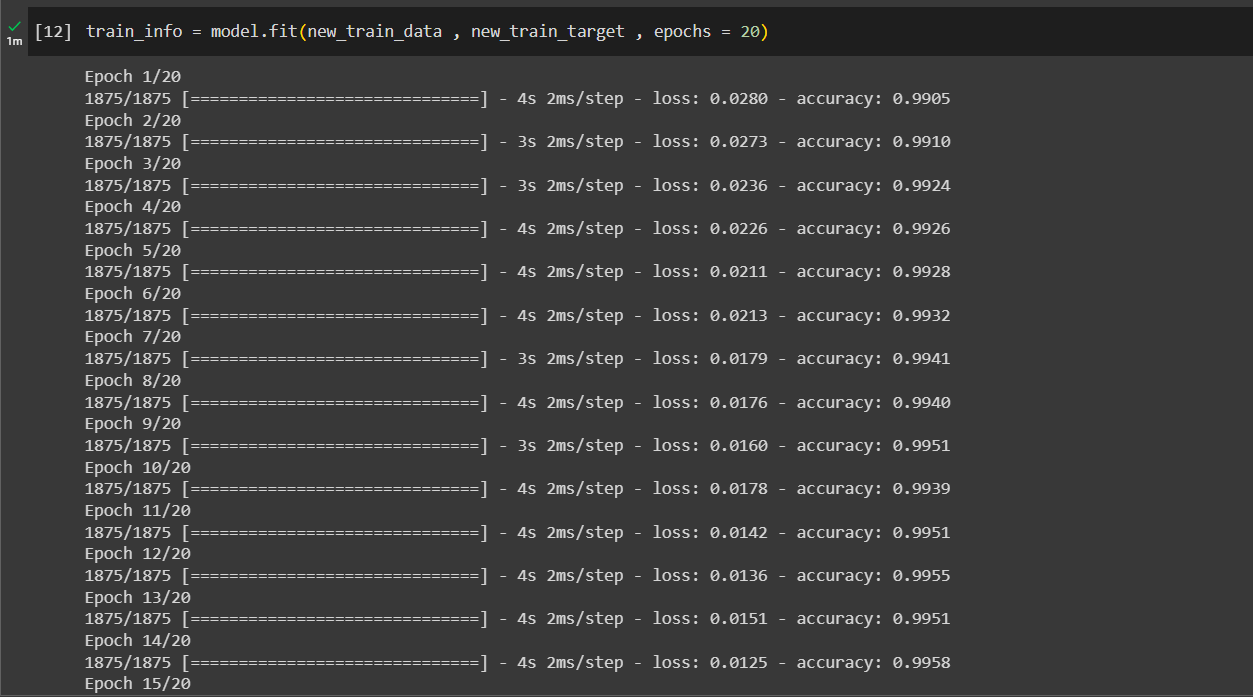
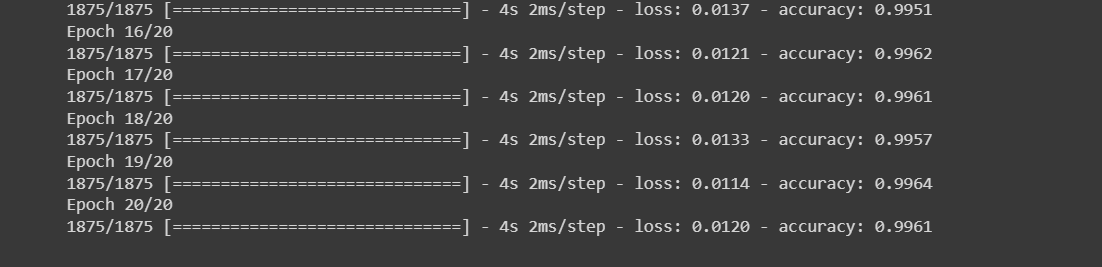
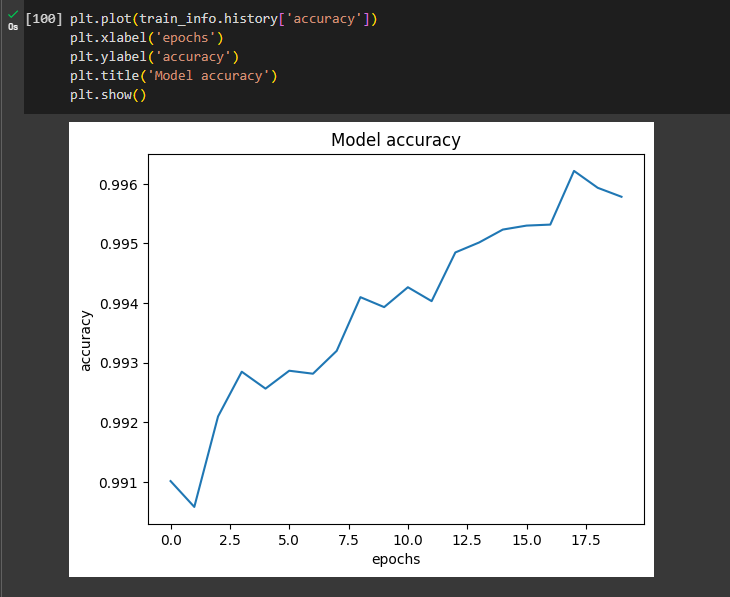
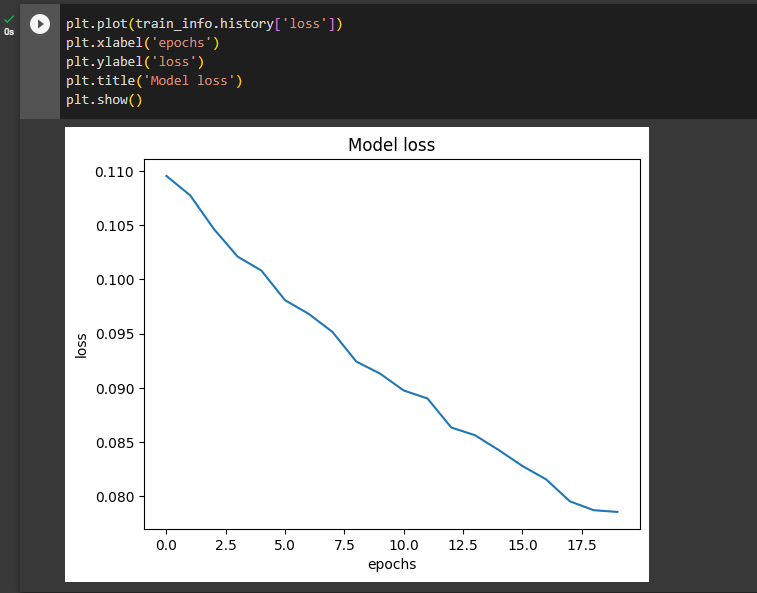
* Changing epoch size

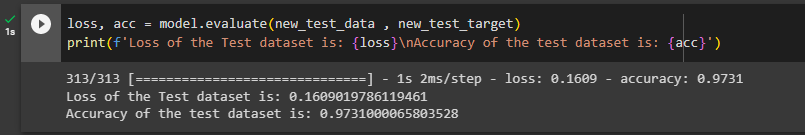
epochs = 5



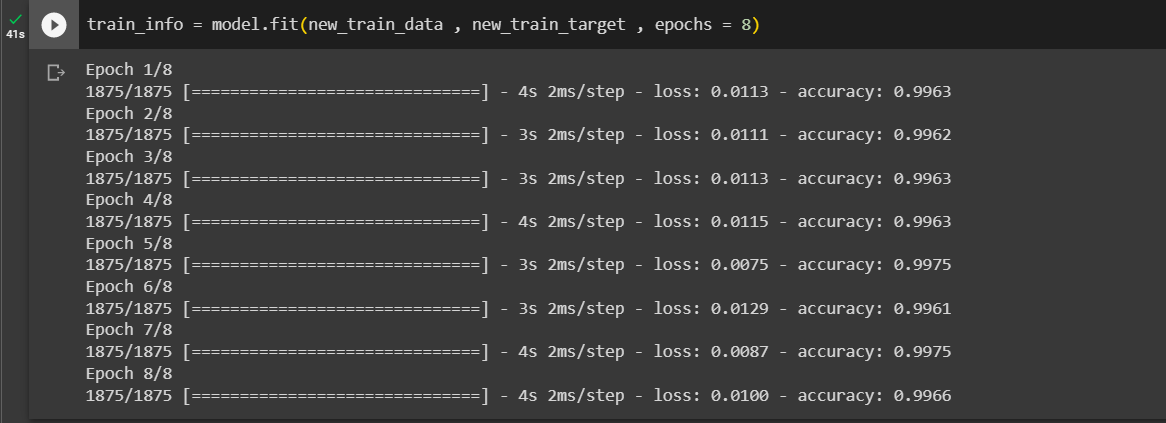


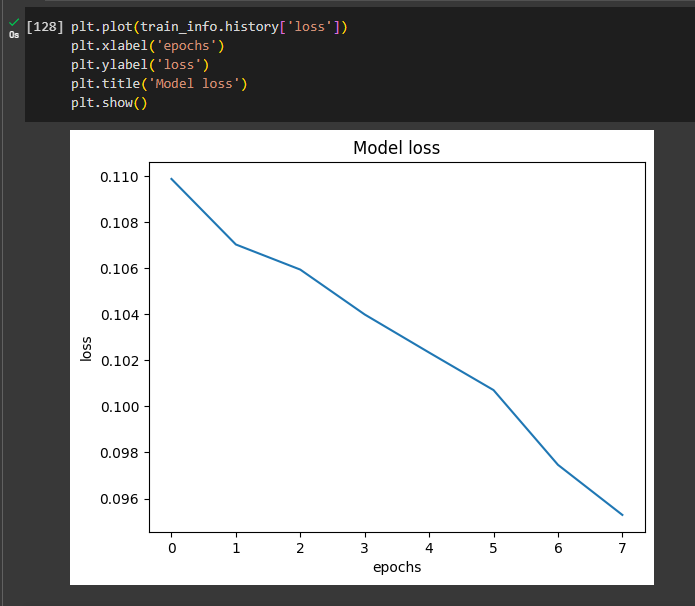
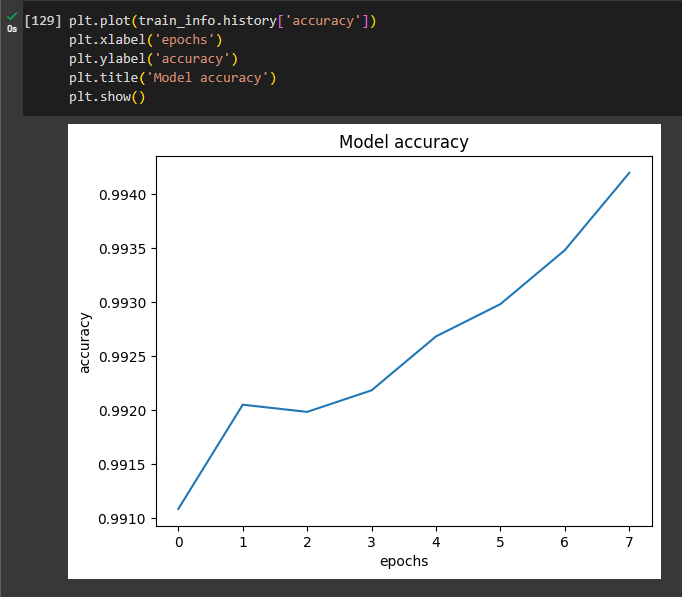


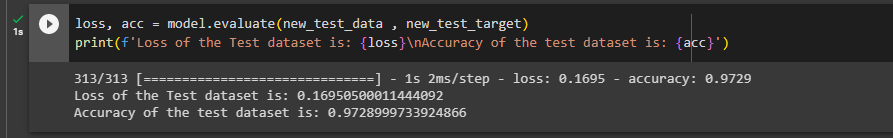
epochs = 20

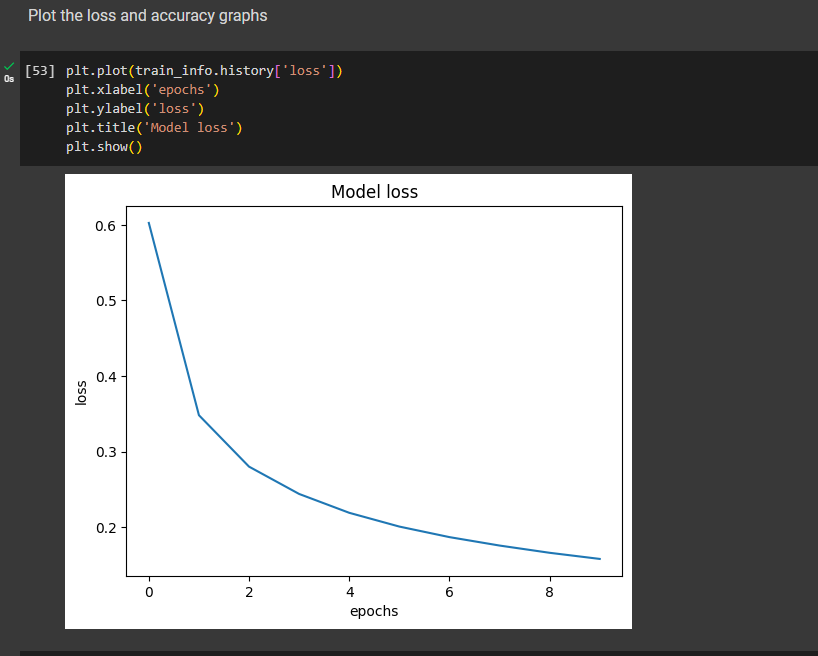
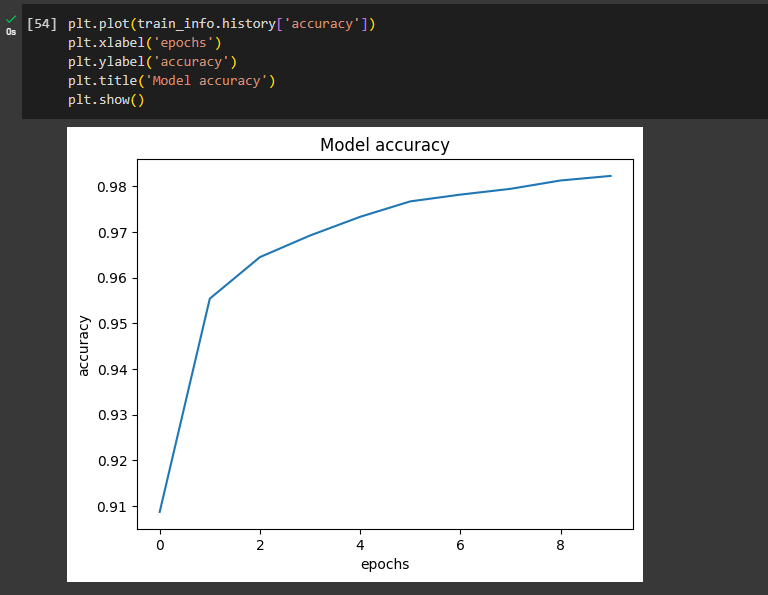


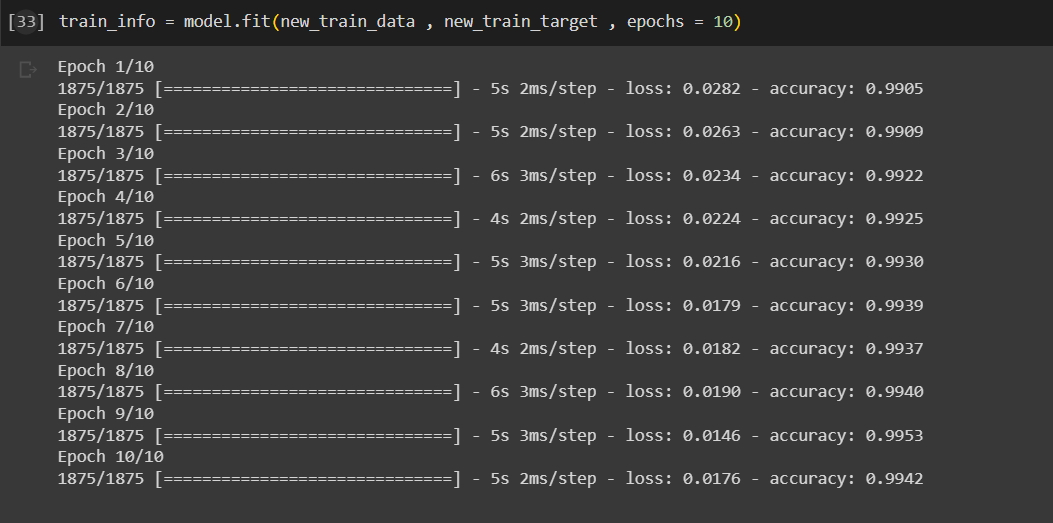
epochs = 8

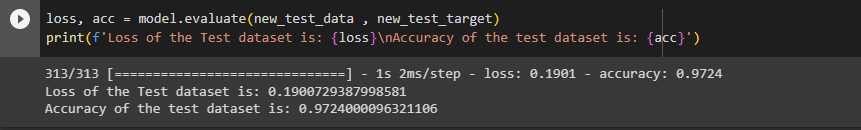




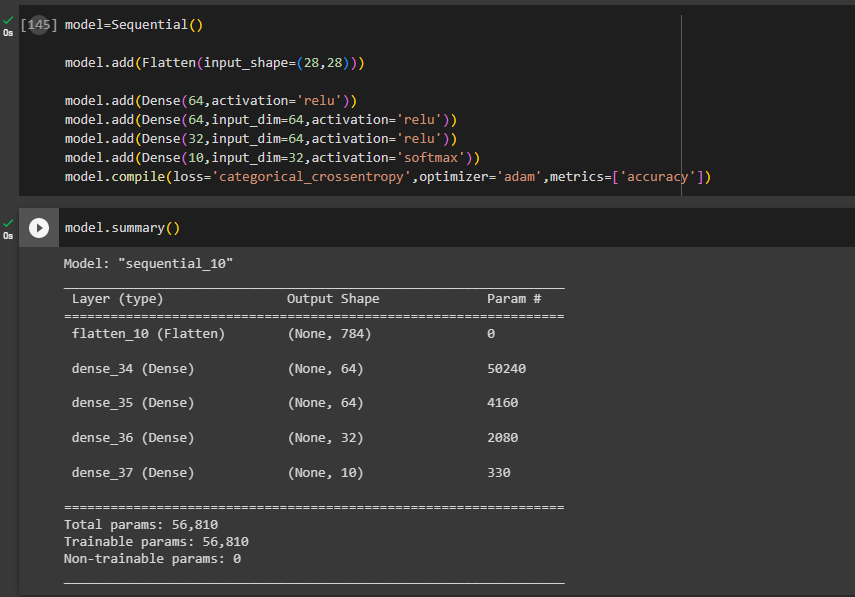


epochs = 10

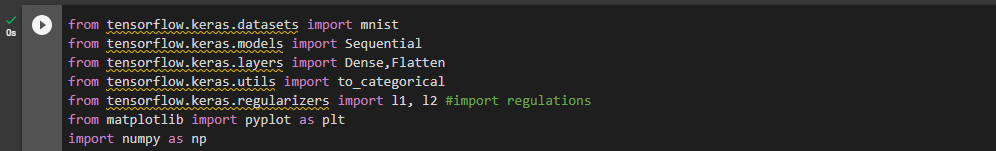


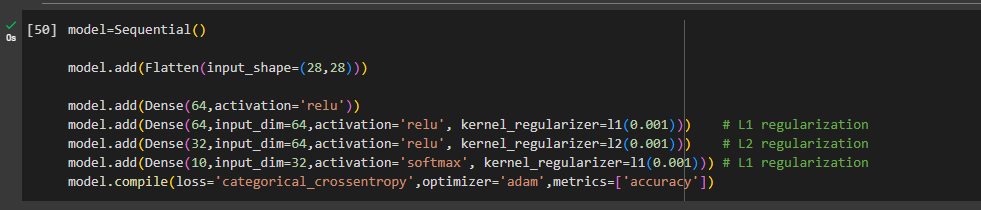


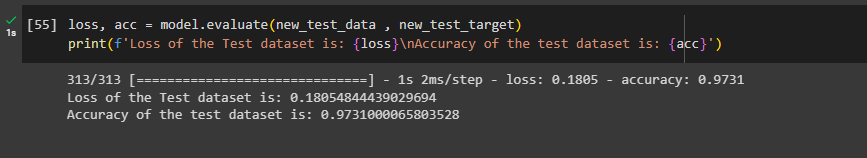
* Changing Number of perceptron in dense layer



### b. Add L1 and L2 regularization terms to the model and retrain the model.







### c. Visualize class-wise test dataset performance using a confusion matrix.

