**Model Report**

In the given task we have 166 features and we have two classes to classify between. We could have gone for standard machine learning algorithms for the classification since they are faster but since the task provided asks to implement a deep learning model and we have enough data to do so I went ahead for a deep learning model.

Now we know CNN works better in case of image data and RNN is good for remembering states, neither do we have an image data nor do we want to remember states so multilayer perceptron would be a good choice in this case.

**Data Analysis**

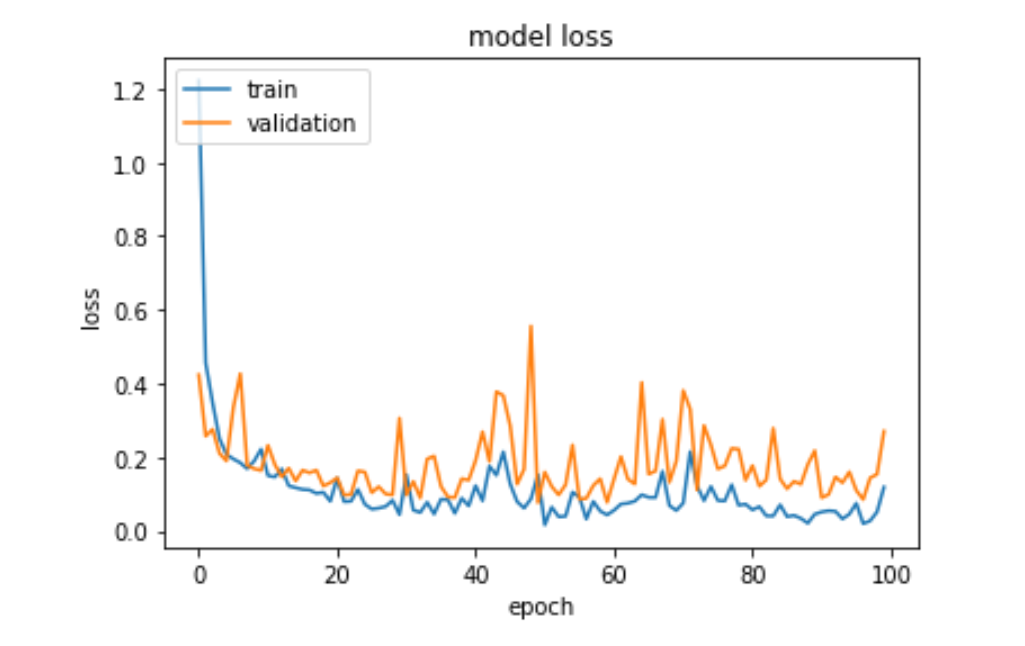
Now for the data analysis part there are no missing values or any other kind of raw data although many features are on different scale from each other and we could use feature scaling to bring them on a single scale but that won’t affect the accuracy that much in this case since the difference in scale is not that large. So I didn’t implement any feature scaling or any other kind of pre-processing of the data.

**Model Implementation**

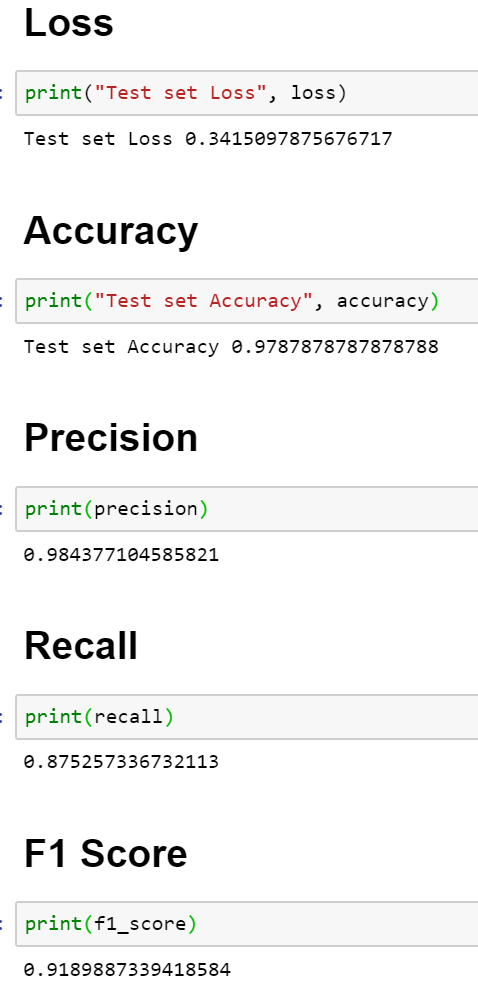
Coming to the implementation part I have used three hidden layers each of 128, 64 and 8 neurons respectively with activation function “relu”, I have also used a dropout layer between the hidden layer 2 and 3 to avoid any kind of over fitting of the data. The final/output layer uses the sigmoid activation function. Optimizer used in the model is ‘rmsprop’ along with “binary\_crossentropy” as loss since we only have to classify between Musk (class 1) and Non-Musk (class 0).

**Loss and Accuracy Graph**





**Final Performance Measures**



**Final Assessment**

The model is working satisfactorily for both training and testing data as the variance is very low. We might be able to improve the model by doing feature scaling or by tuning the hyperparameters.