Report Practical Exercise 03

**Task 1** : Lowdata Regime and Autoencoders

1a:

2. *Implement the calculation of accuracy in the test function. Is accuracy appropriate here? Examine the data distribution to support your argument.*

In our data distribution, there is an imbalance between classes, with a greater number of samples that indicating the presence of disease than healthy ones. In the training set, we have 134 positive samples and 166 negative samples, while in the test set, there are 10,505 positive samples and 11,928 negative samples. By looking at the evaluation on the test set, we have a true positive rate of 0.5207 and a true negative rate of 0.6341. Our model seems to be overpredicting the negative class. This will lead to a higher accuracy because there are more negative samples present in the data. If we look at the balanced accuracy (balanced accuracy: 0.5774 / accuracy: 0.581) we can see that it’s lower than the conventional accuracy, which occurs when either the sensitivity or specificity is low due to a bias in the classifier towards the dominant class. While the training set has a perfect balanced accuracy/ accuracy of 1, we don’t rely on its evaluations as it is overfitting to the training data.

3. *What would be a better way to select a model than a fixed number of steps that would allow us to set a very high number of epochs and not overfit too much? How would we select the best model using such a ”smart train” function if we are still worried about overshooting our optimum?*

If we want to set a high number of epochs while preventing overfitting, we can make use of early stopping. Here we do monitor the validation metric (such as e.g. validation loss) and stop training once the performance on the validation set stops improving or starts degrading. To fine-tune this approach and avoid prematurely stopping training, a patience parameter can be set to control the number of epochs to wait before stopping. If we are worried about overshooting our optimum due to factors like setting a high patience parameter, we can also do model checkpointing. Here we save the models weights whenever there is an improvement in the validation metric. This way, even if training continues for an extended period, you have checkpoints of the model at different stages, allowing you to revert to the best-performing model if needed. It generally should be added that early stopping is not always a good idea because the model does not always converge in a straightforward manner. In most cases it is a good idea to make training harder using regularization techniques, data augmentation or simply making the model less capable to prevent overfitting. So, it’s essential to consider the characteristics of the specific problem before deciding about if you should make use of early stopping.