Lab Logbook Requirement:

1. *Create your own Multi-layer Perceptron (MLP) with two hidden layers, where the first hidden layer cells' number equals the last three digits of your SID. The number of cells in the next hidden layer is approximately two times smaller. For example, if your SID is 2287167, the number of cells on the first hidden layer is 167, and on the second – 84. Take epochs=10. Leave other parameters the same as in the practical session.*
2. *Compile the model.*
3. *Train your MLP with the same datasets and demonstrate the received MAE.*
4. *Compare your MAE with the MAE of the MLP in the practical session.*
5. *Please only add to your Lab Logbook a print-screen of your MLP architecture using model.summary() and the resulting MAE.*

¶

```python
model = keras.Sequential([

    keras.layers.Dense(529, input_dim = 500, activation = tf.nn.relu, kernel_initializer = "normal"),    #sid = 2368529

    keras.layers.Dense(265, activation = 'relu', kernel_initializer = "normal"),

    keras.layers.Dense(1)

])

print(model.summary())
```

Model: "sequential_4"

Layer (type)	Output Shape	Param #
dense_16 (Dense)	(None, 529)	265,029
dense_17 (Dense)	(None, 265)	140,450
dense_18 (Dense)	(None, 1)	266

Total params: 405,745 (1.55 MB)
Trainable params: 405,745 (1.55 MB)
Non-trainable params: 0 (0.00 B)
None

```python
model.compile(optimizer = "adam", loss = "mse", metrics = ["mae"])
```

```python
history = model.fit(X_train, y_train, batch_size =10, epochs = 10, validation_split = 0.2, verbose = 1)
```

```
Epoch 1/10
2640/2640 ──────────────── 20s 7ms/step - loss: 0.0356 - mae: 0.0442 - val_loss: 0.0107 - val_mae: 0.0936
Epoch 2/10
2640/2640 ──────────────── 18s 7ms/step - loss: 2.1776e-04 - mae: 0.0115 - val_loss: 0.0076 - val_mae: 0.0794
Epoch 3/10
2640/2640 ──────────────── 18s 7ms/step - loss: 1.9438e-04 - mae: 0.0108 - val_loss: 0.0048 - val_mae: 0.0614
Epoch 4/10
2640/2640 ──────────────── 18s 7ms/step - loss: 1.1215e-04 - mae: 0.0081 - val_loss: 0.0013 - val_mae: 0.0300
Epoch 5/10
2640/2640 ──────────────── 18s 7ms/step - loss: 9.0387e-05 - mae: 0.0071 - val_loss: 6.4911e-04 - val_mae: 0.0200
Epoch 6/10
2640/2640 ──────────────── 22s 7ms/step - loss: 7.6753e-05 - mae: 0.0066 - val_loss: 0.0013 - val_mae: 0.0323
Epoch 7/10
2640/2640 ──────────────── 18s 7ms/step - loss: 6.4826e-05 - mae: 0.0061 - val_loss: 0.0012 - val_mae: 0.0310
Epoch 8/10
2640/2640 ──────────────── 18s 7ms/step - loss: 5.9723e-05 - mae: 0.0058 - val_loss: 3.6299e-04 - val_mae: 0.0152
Epoch 9/10
2640/2640 ──────────────── 18s 7ms/step - loss: 5.8251e-05 - mae: 0.0058 - val_loss: 4.3948e-04 - val_mae: 0.0172
Epoch 10/10
2640/2640 ──────────────── 21s 7ms/step - loss: 5.0773e-05 - mae: 0.0055 - val_loss: 5.6606e-04 - val_mae: 0.0202
```

```python
[58]: print("Mean absolute error: %.5f" % mae)
```

```
Mean absolute error: 0.01238
```