**DNN Training**

Not-Yets-MacBook-Pro:ML\_LoRaWAN home$ python3 prediction\_train.py --model\_type='deepnn'

[Epoch 199] [Batch 4689] [MSELoss loss: 0.003415]

Finished Training

1408

predicted [ 5.9217564e-06 -8.1873685e-04 8.6675066e-01 1.2705224e-02

6.0971260e-02 -7.6981261e-05 -1.8236693e-04 -1.2560934e-04

6.0763020e-02 5.8107078e-05]

real\_value [2.48987050e-05 1.30693886e-04 8.42429009e-01 5.74927852e-03

7.58186105e-02 1.90630710e-08 6.53591005e-08 1.43645555e-05

7.58330595e-02 1.26758517e-09]

**SVR Training**

Not-Yets-MacBook-Pro:ML\_LoRaWAN home$ python3 prediction\_train.py --model\_type='svr'

[Epoch 199] [Batch 4689] [MSELoss loss: 0.018003]

Finished Training

1408

predicted [ 2.09800899e-04 5.29471040e-03 8.73757720e-01 1.36155784e-02

5.74361309e-02 1.21332705e-05 5.96679747e-05 -4.11660923e-03

5.35032302e-02 2.33861501e-05]

real\_value [2.97650240e-05 5.58094201e-03 8.43339867e-01 1.84001438e-02

6.57805142e-02 4.82286405e-07 1.27664048e-06 5.42243818e-04

6.63245169e-02 2.48010921e-07]

Not-Yets-MacBook-Pro:ML\_LoRaWAN home$

**INPUT AND OUTPUT**

* Before reduction we gathered 33 parameters and after reduction, we ended up with 25 parameters.
* The first 15 columns are the input parameters
* The remaining 10 columns are taken as an output parameter
* The total number of datasets we used are 35192

**ABSOLUTE DIFFERENCE BETWEEN REAL AND PREDICTED**

* DNN – lays between 0 to 0.025 according to our previous result this result is very improved its almost close to zero which makes it similar to real data
* SVR – lays between 0 to 0.040 which is better than our previous result as well

**Battery Consumption in years**

* We compared the predicted with the real data in both ML algorithms by taking 10 rows of the predicted and real data randomly and compared to see which one shows better battery consumption.
* Both ML algorithms show better result than the previous generated but DNN shows better result than SVR