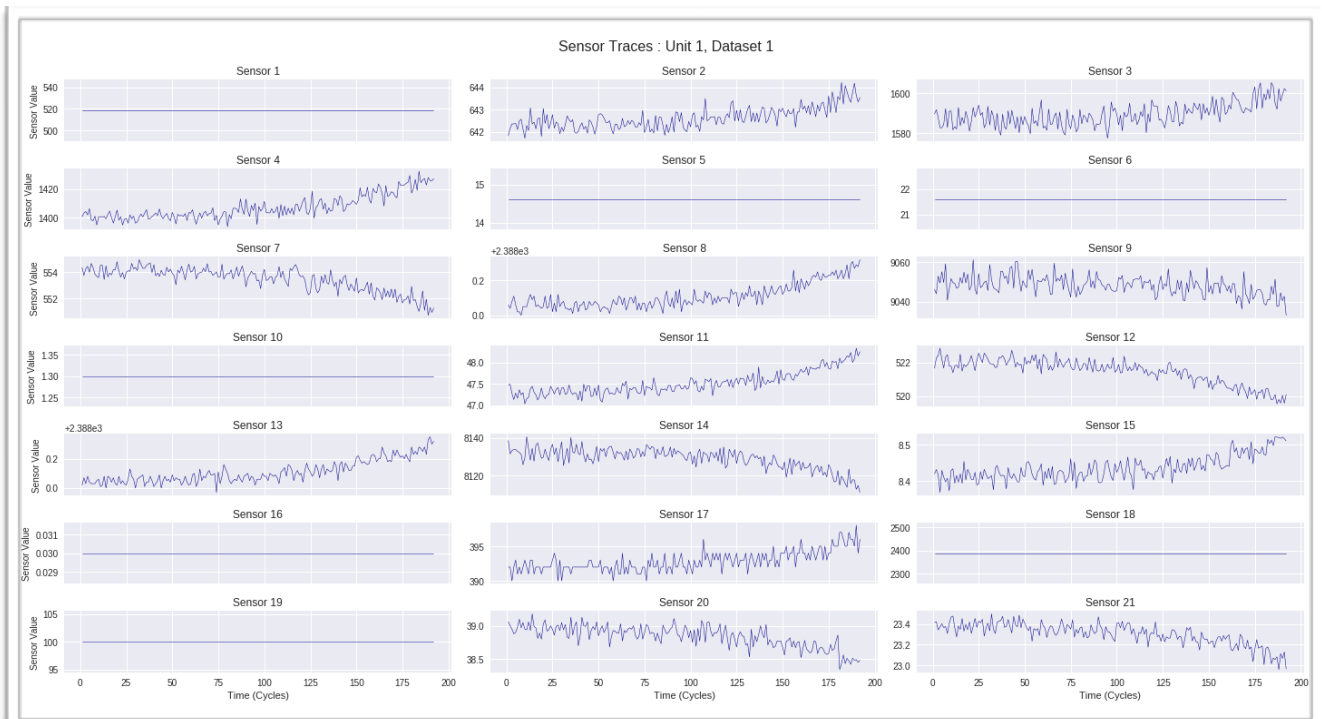


## Predictive Maintenance



### First approach:

Used linear regression with only selected sensors and settings, I basically dropped all the quite sensors shown above and tried to predict RUL using a simple way.

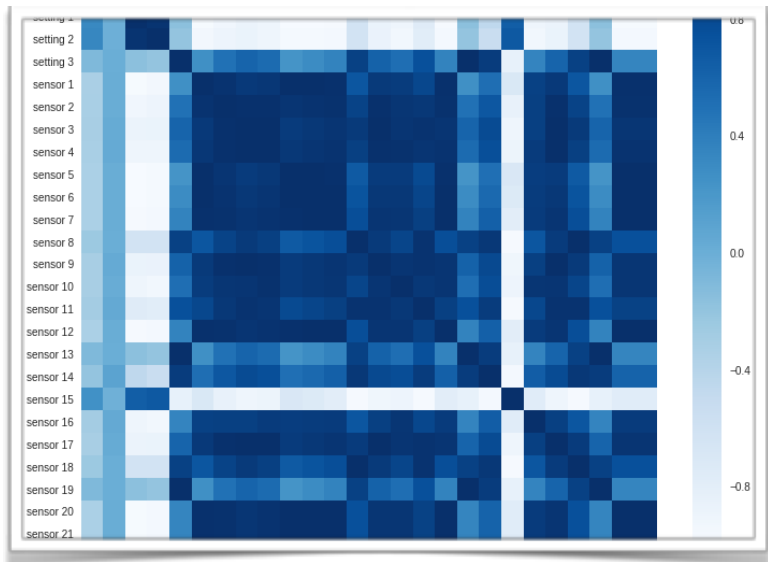
### Results:

Turned down with poor results.

### Conclusion:

Dropping sensors was a bad idea. Simple Linear Regression was not a good choice either.

Please check the notebook file for in depth code explanations and decision backgrounds.



The second step was understanding the input sensor data and determining whether it was sufficient enough information to predict the RUL of the engine.

From the visualisations I could see that each sensor does follow a specific degradation pattern when the engine experiences a failure event and by analysing the sensor value distribution when an engine is healthy compared to when it is about to fail we could see that the sensor values could distinguish between a healthy and unhealthy engine. Further, sensors behaved in the first dataset was somewhat not similar to second or any other dataset.

The next step was feature engineering and, then the data sets were split into appropriate train and test sets. Dataset identification numbers were replaced with some dummies and test dataset shrink and merge with given real answers comparing their appropriate values.

Data normalisation techniques and reshaping were performed on both data sets right after their train test splits.

Deciding and developing the Neural network.

Finally, tweaking and fine tuning was performed.

## **Challenges faced**

Deciding and implementing the Neural network was challenging and tweaking and fine tuning was the most challenged part of the task.

I have developed few NNs for the University homeworks but none of them were in Keras, thats the main reason I chose Keras for this, which will be ended up teaching me many things in Keras.

Spent many hours testing different layers, activation functions, optimisers as well as epoch and batch sizes during the implementation and overfitting was the toughest issue to be addressed.

But, in overall it was a great learning experience.

## **Conclusion**

Franking speaking, there are still many aspects of this problem is yet to explore.

From the figures I got from predictions, it's clear that the models struggle to produce accurate estimates when the RUL is especially large or small. Further I was not doing any real preprocessing or feature extraction. I have tried to use L2 regularisation to minimise this issue but then it ended up with model predicting same value every time, kind of overfitting.

Last but not least, compared to my first approach of linear regression and many many failed attempts of tweaking the NN, I have managed to get a decent results at the end.