**Machine Learning Project**

PREDICTIVE MAINTENANCE OF AIRCRAFT ENGINE USING LSTM NETWORK

**Subject:** Deep Learning, Machine Learning.

**Dataset:** NASA Turbofan dataset.

**Algorithm Applied:** Long Short Term Memory.

**Application Field:** Industry Predictive Maintenance

# PROBLEM STATEMENT

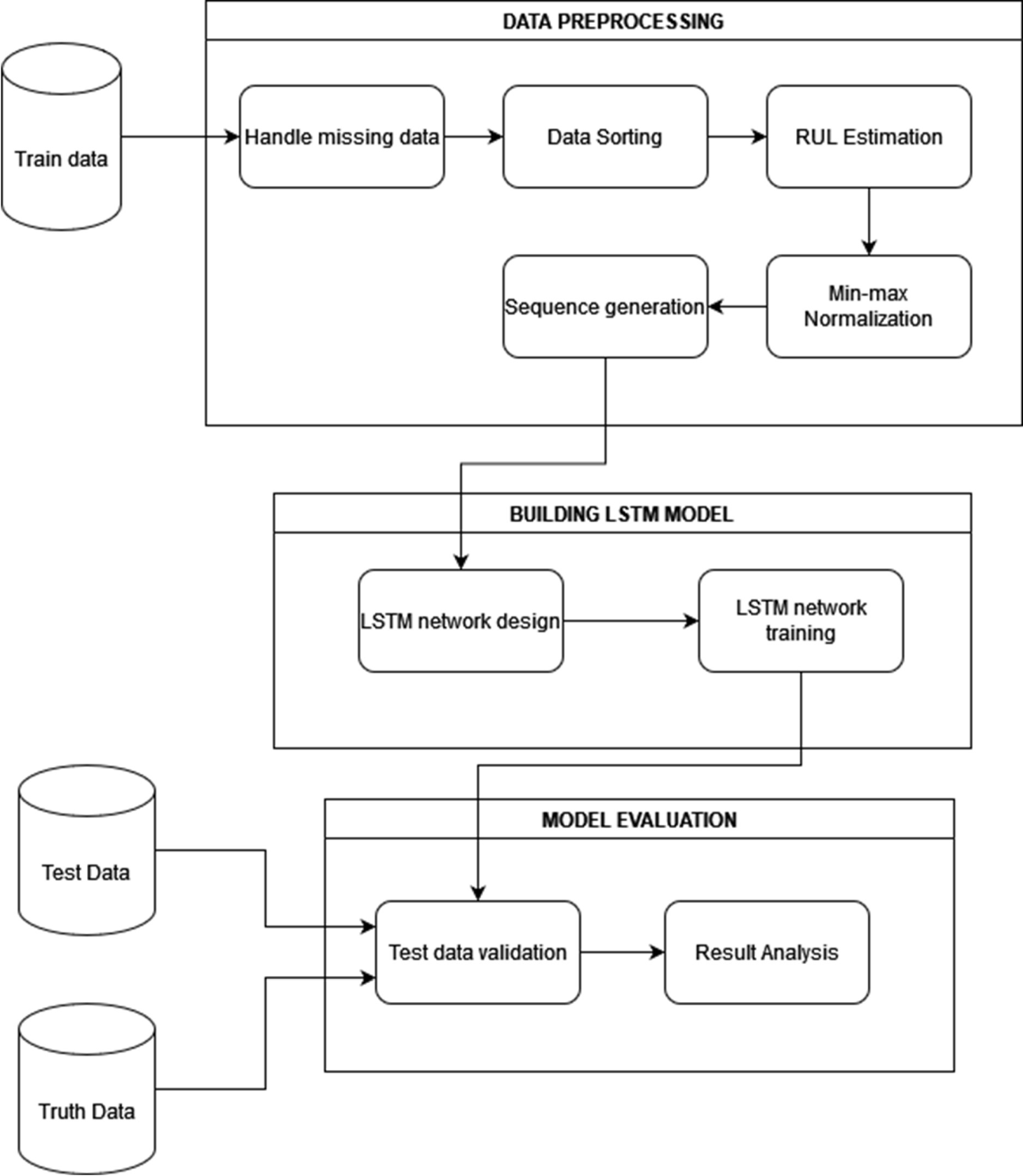
What if a part of aircraft could let know when the aircraft component needed to be replaced or repaired? It can be done with continuous data collection, monitoring and advanced analytics. In the aviation industry, predictive maintenance promises increased reliability as well as improved supply chain and operational performance. The main goal is to ensure that the engines work correctly under all conditions and there is no risk of failure. The main source of data regarding the health of the engine is measured during the real time. Several variables are calculated, including fan speed, quantity and oil pressure and environmental variables such as temperature, air speed.

Aspects related to the maintenance have become especially failure of a have catastrophic consequences. Current systems have the ability to warn inform only when the failure occurred. An early warning system that predicts the occurrence of component failure is required. A machine learning approach, using LSTM networks, is proposed to predict the RUL (Remaining Useful Life) by analysing failure patterns in the past. Training of LSTM networks are carried out on a high performance large-scale processing engine.

# OBJECTIVE

A machine learning approach through the use of Long Short Term Memory (LSTM) networks is used to analyze sensor time series sequences to estimate the RUL of turbofan engines, and we provide an analysis where we show how the LSTM performance changes when varying its internal hyper parameters.

# OVERALL ARCHITECTURE DIAGRAM



**DETAILS OF MODULE DESIGN**

1. Data pre-processing
2. Building LSTM network
3. Training the model
4. Testing the model and Result analysis.

# DATA PRE-PROCESSING:

The dataset has inputs.

* + Training data: It is the aircraft engine run-to-failure data.
  + Testing data: It is the aircraft engine operating data without failure events recorded.
  + Ground truth data: It contains the information of true remaining cycles for each engine in the testing data.

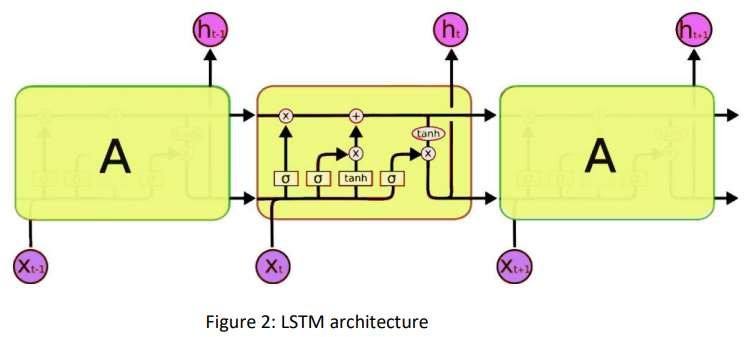
The training data ("train\_FD001.txt") consists of multiple multivariate time series with "cycle" as the time unit, together with 21 sensor readings for each cycle. Each time series can be assumed as being generated from a different engine of the same type. Each engine is assumed to start with different degrees of initial wear and manufacturing variation, and this information is unknown to the user. In this simulated data, the engine is assumed to be operating normally at the start of each time series. It starts to degrade at some point during the series of the operating cycles. The degradation progresses and grows in magnitude. When a predefined threshold is reached, then the engine is considered unsafe for further operation. In other words, the last cycle in each time series can be considered as the failure point of the corresponding engine**.** Taking the sample training data shown in the following table as an example, the engine with id=1 fails at cycle 192, and engine with id=2 fails at cycle 287.with this information we can calculate RUL for training dataset

The testing data ("test\_FD001.txt") has the same data schema as the training data. The only difference is that the data does not indicate when the failure occurs (in other words, the last time period does NOT represent the failure point). Taking the sample testing data shown in the following table as an example, the engine with id=1 runs from cycle 1 through cycle 31. It is not shown how many more cycles this engine can last before it fails.

The ground truth data ("RUL\_FD001.txt") provides the number of remaining working cycles for the engines in the testing data. Taking the sample ground truth data shown in the following table as an example, the engine with id=1 in the testing data can run another 112 cycles before it fails.

# BUILDING THE MODEL

Long short term Memory Network Known as LSTM. It is one of the best kind of RNN with capability of avoiding gradient dispersion. It is designed to avoid long term dependencies. LSTM Cells are where data is transfers and updated, cell states of the LSTM is changed as compared to RNN. Network is based on short term states, long term states and its three gates: input gate, output gat and forget gate. Where ft is forget gate use to forget the information that is no longer required.



ft = σ(Wf · [ht−1, xt] + bf)

here σ is activation function x is input to the gate and b is bias vector.

The input gate contains two path one for the new input and second for vector I generated by forget gate, which is use to modify the cell state.

it = σ(Wi · [ht−1, xt] + bi)

C˜ t = tanh(Wc · [ht−1, xt] + bc)

Where W are the weight matrices and b are the bias vectors for input gate with activation function of tanh. And updated state of gate is:

Ct = ft ⊗ Ct−1 + it ⊗ C output gates have two parts as input gate.

ot = σ(Wo · [ht−1, xt] + bo) ht = ot ⊗ tanh(Ct)

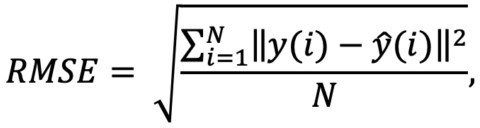
# TRAINING THE MODEL.

We input our training dataset for which we know the Remaining Useful lifetime[RUL]

,our LSTM model takes this as input and calculates the RUL. From our experience, we noticed that the obtained results can be very different when varying the model

architecture so the base idea of such an analysis is to show how the model performance, in terms of Root Mean Squared Error (RMSE), changes over a different set of hyperparameters. RMSE is commonly used in supervised learning applications, as RMSE uses and needs true measurements at each predicted data point.

Root mean square error can be expressed as



where N is the number of data points, y(i) is the i-th measurement, and y ̂(i) is its corresponding prediction.

We were able to analyze the RMSE variations when changing the hyperparameters values and understand which are the elements that affects it mostly. The result of such an analysis is very important because it allows to have a better understanding on how this kind of models work and how we can improve them by changing their configuration. Finally we fix the hyper parameters which give us high accuracy.

# TESTING THE MODEL AND RESULT ANALYSIS.

We finally apply the Training dataset whose ground truth Remaining useful lifetime is is given in a separate dataset into out trained and optimized LSTM model , we obtain the output for each case of aircraft engine based on the aircraft id. The Rul predicted and actual ground truth is compared to analyze the result.

Model performance on the test data showing the differences between the predictions made and the ground truth values where the generic i-th difference has been computed as:

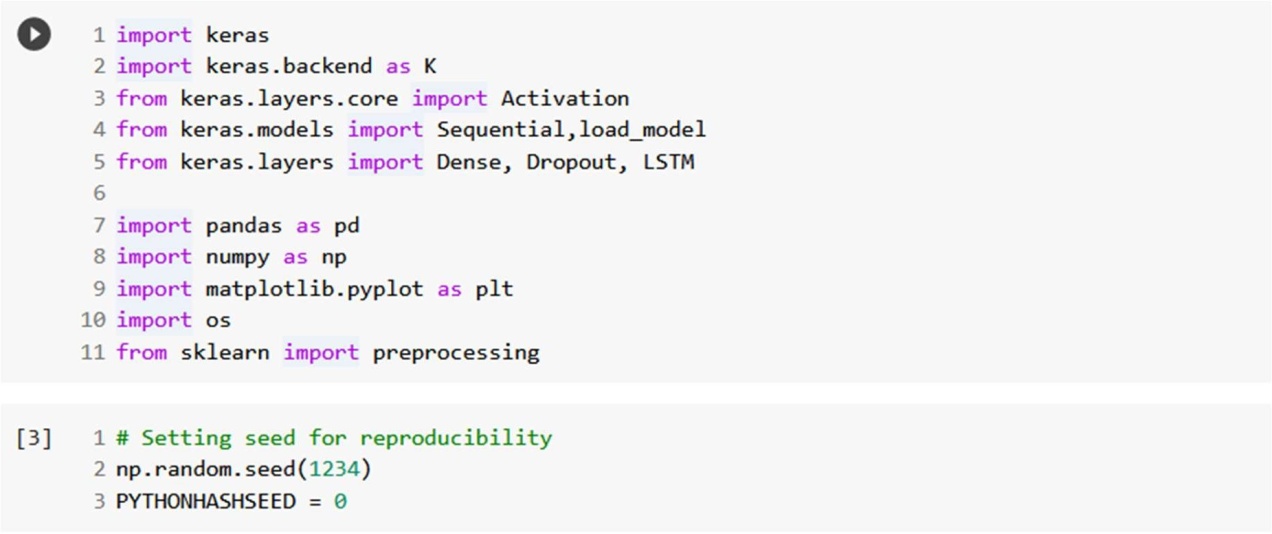
di = y^i − yi

In this sense, a value higher than zero means that our model made an optimistic prediction while a value lower than zero means a pessimistic prediction. Nevertheless, the majority of them presents a difference near to zero meaning that the model is able to correctly predict the RUL with a very good level of accuracy. We analyze the accuracy, error rate using various graph plotting packages such as pyplot, seaborn to get a conclusion of how our LSTM model perform for predicting Remaining useful lifetime for Training dataset.

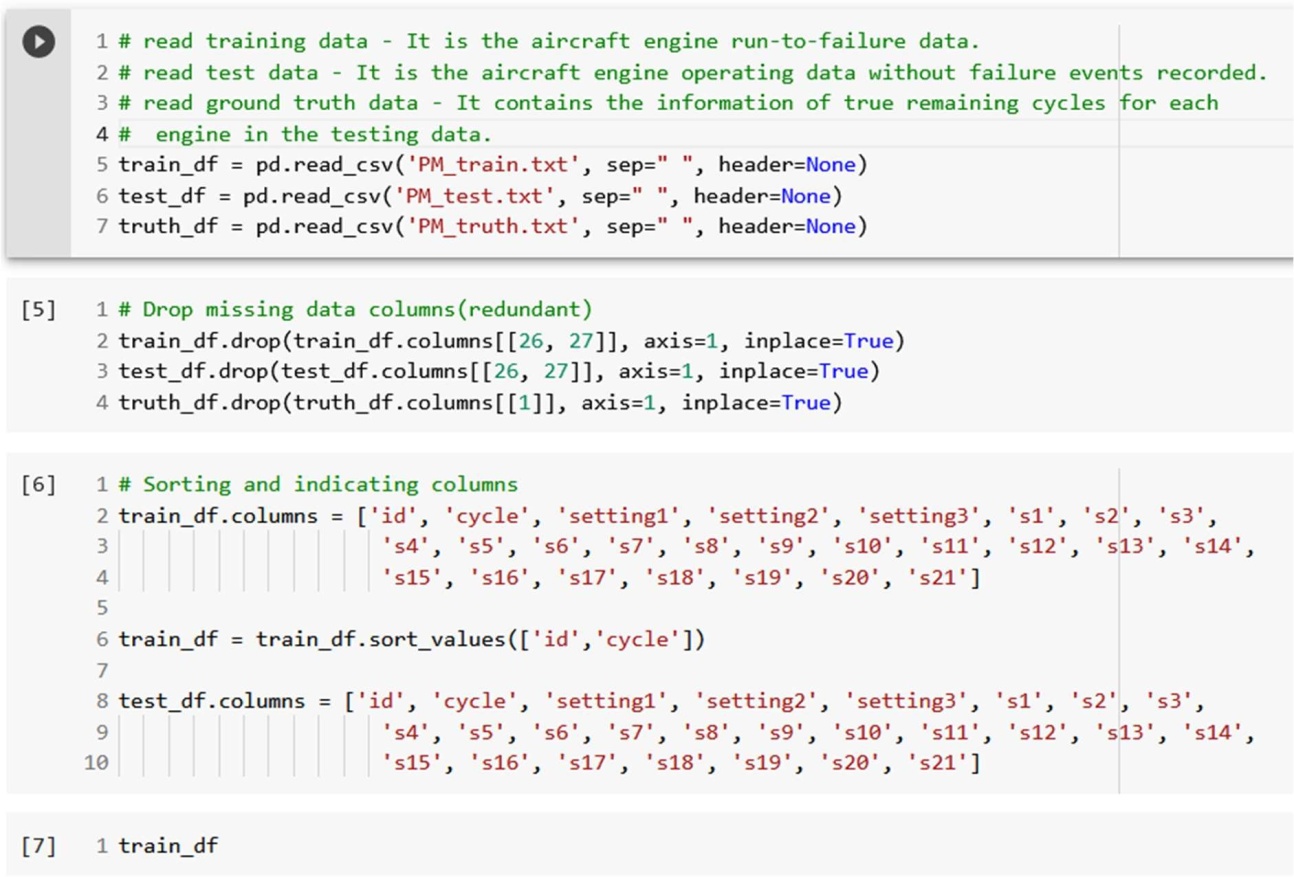
# IMPLEMENTATION

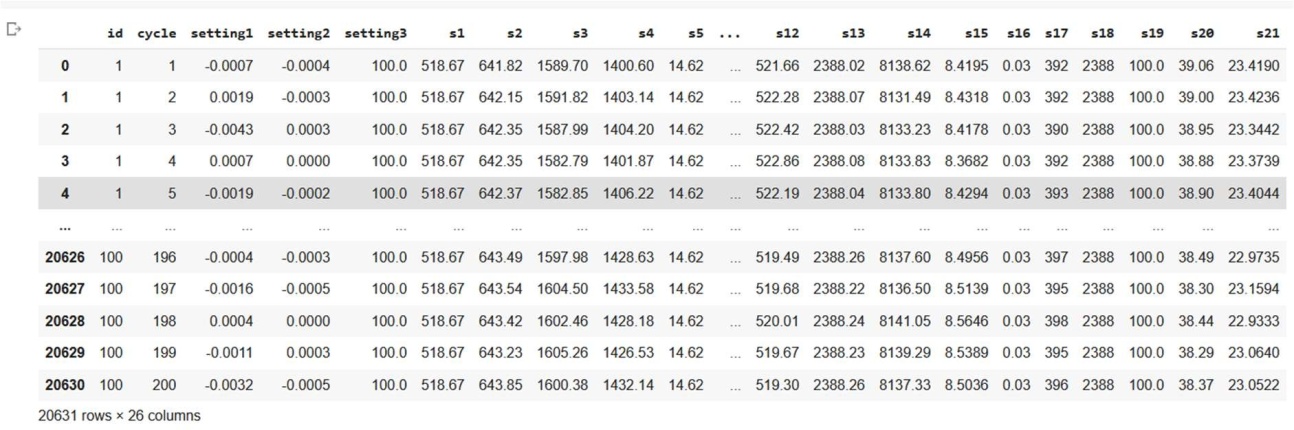
**MODULE-1:DATA PREPROCESSING**

Required packages and libraries needed for building the model are imported.

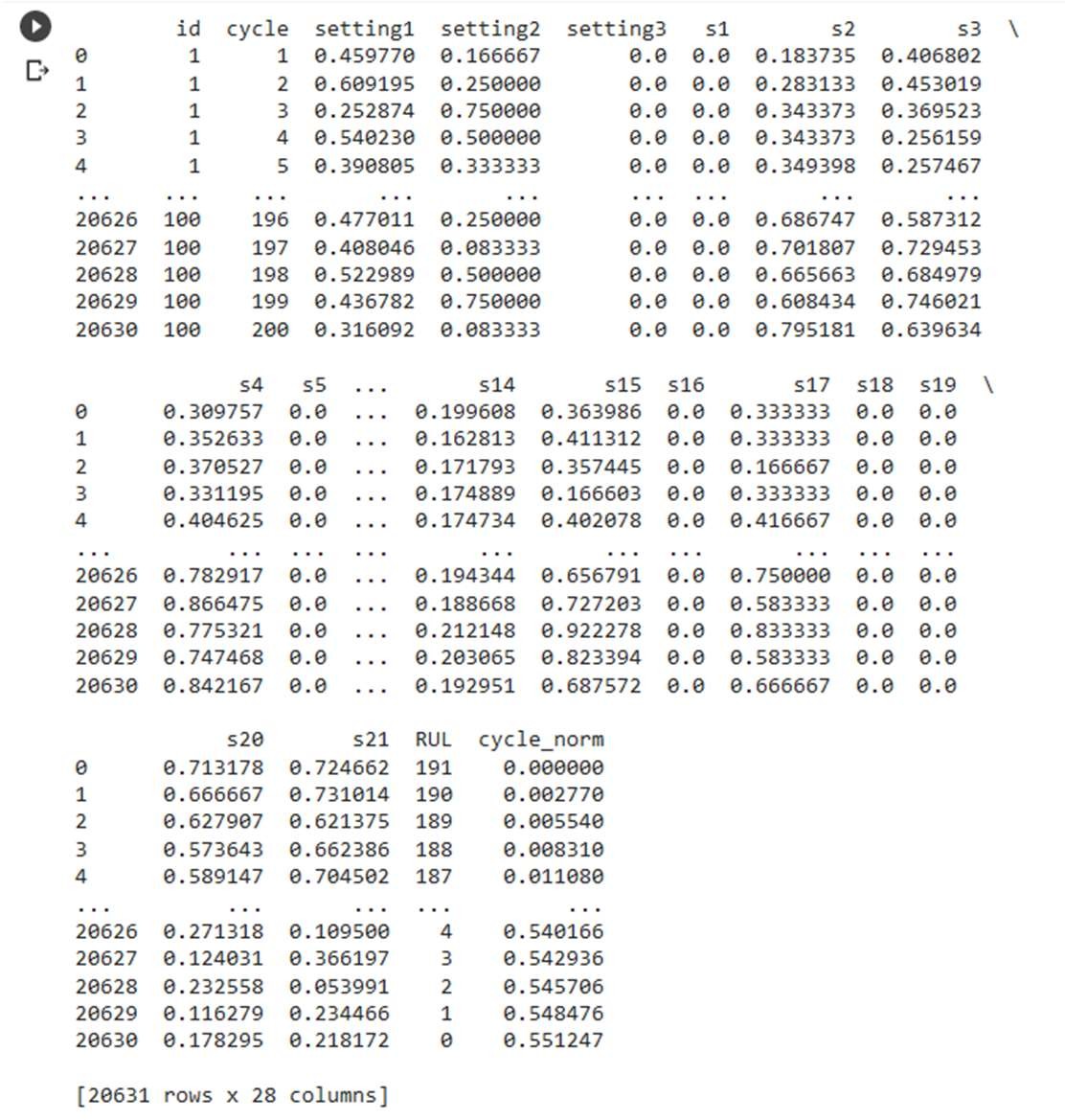
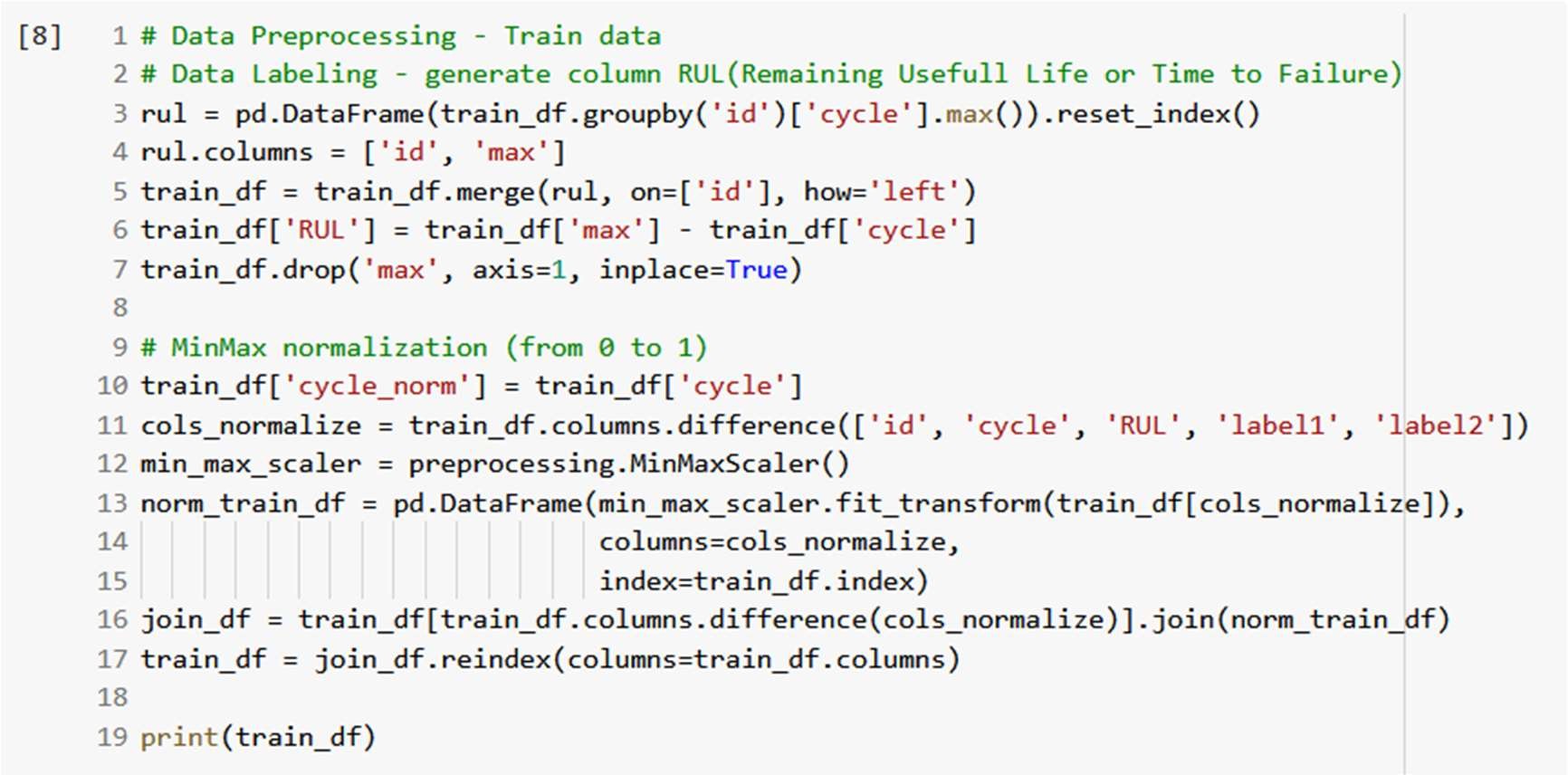


The dataset containing training and testing data are uploaded and the null values are removed in order to avoid false information.

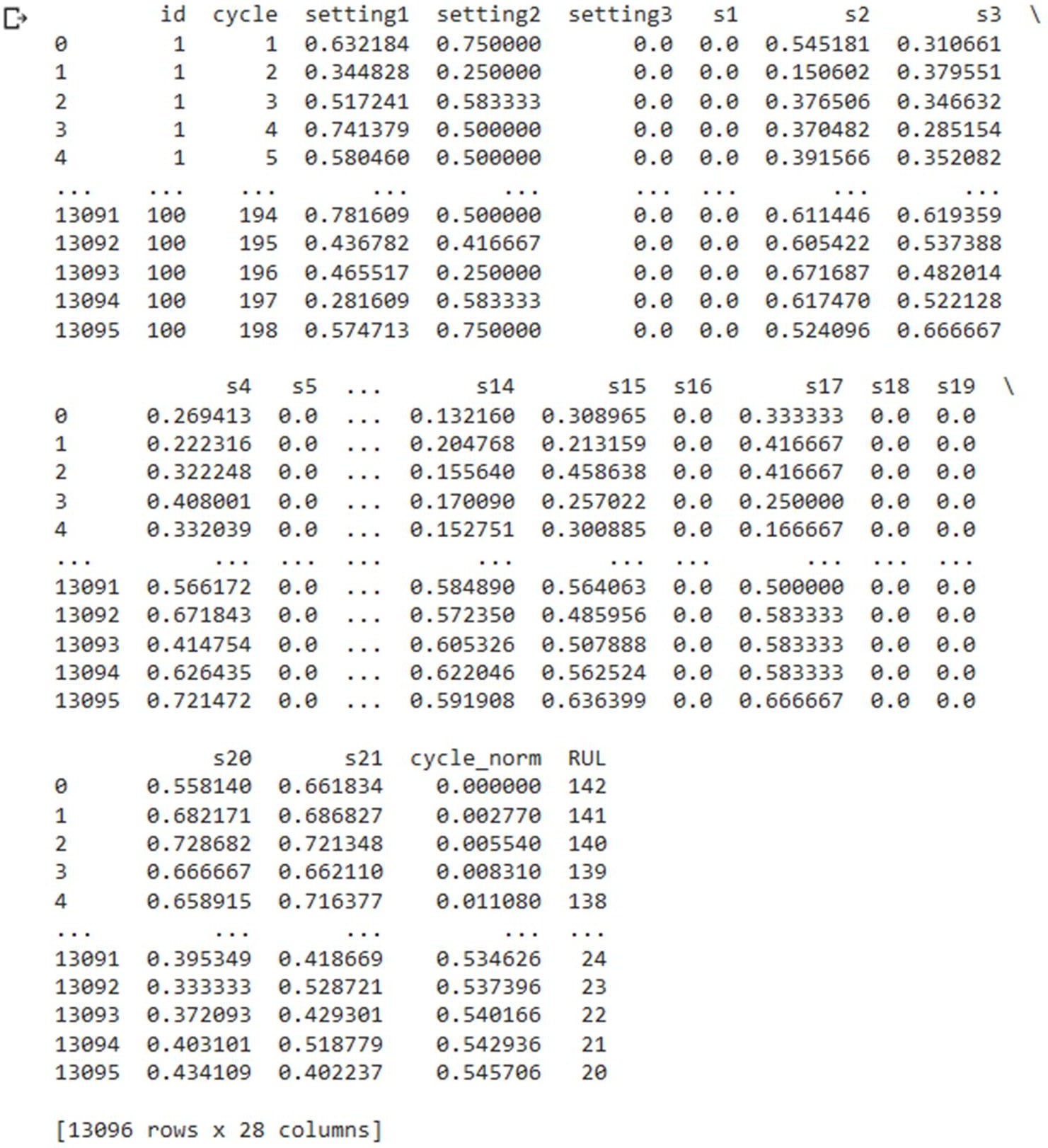
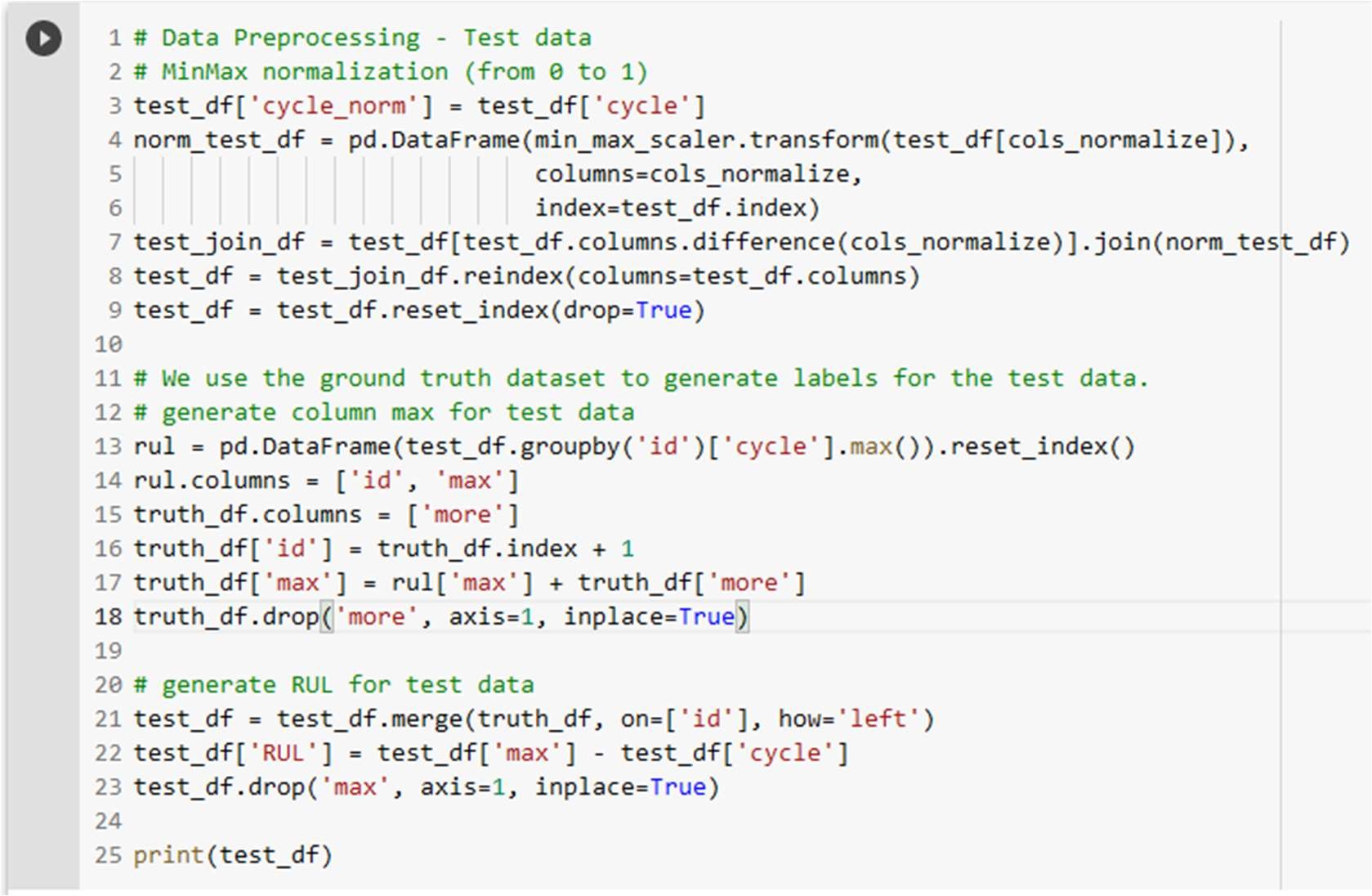


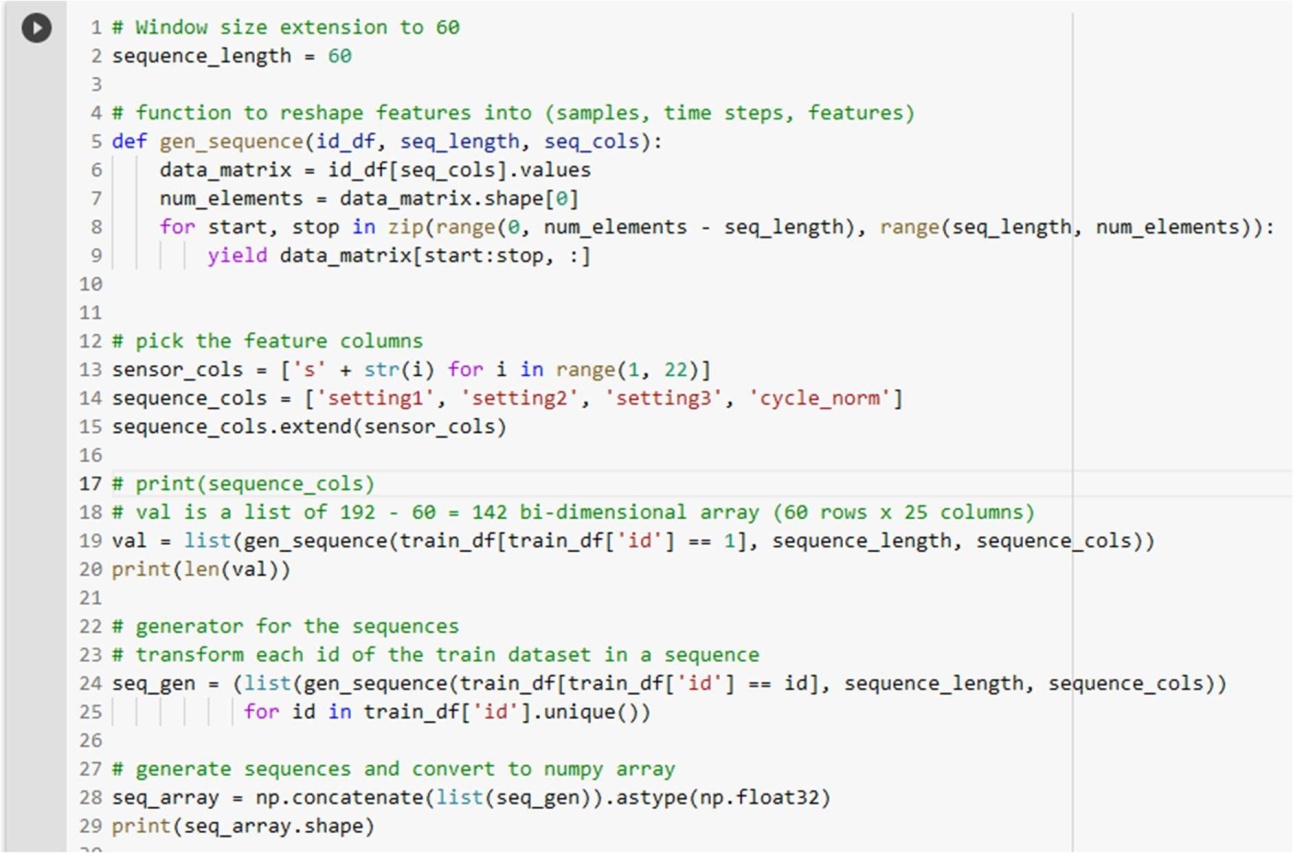


Then the Remaining Useful Life is calculated using the number of cycles given in the dataset for training dataset.



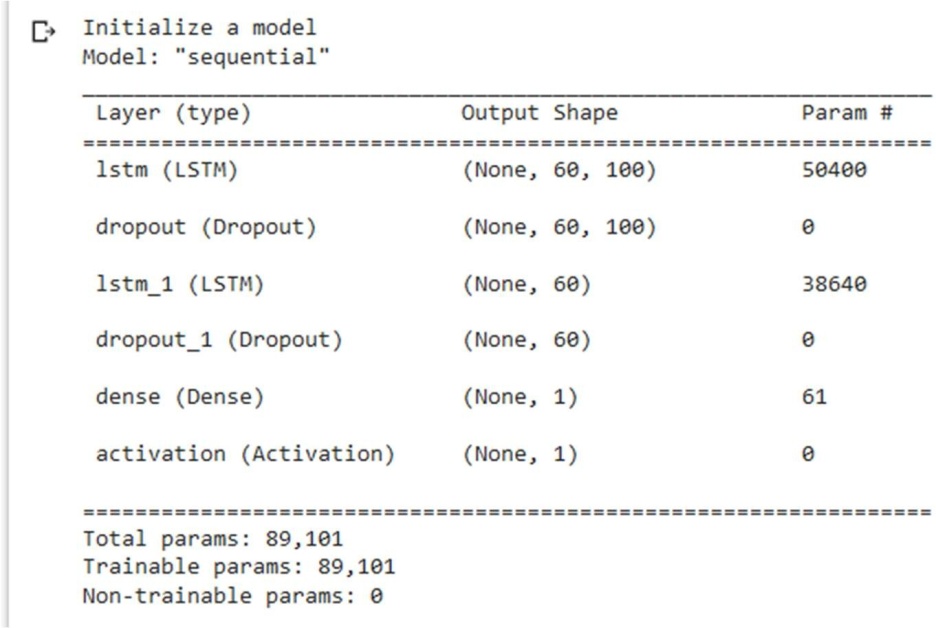
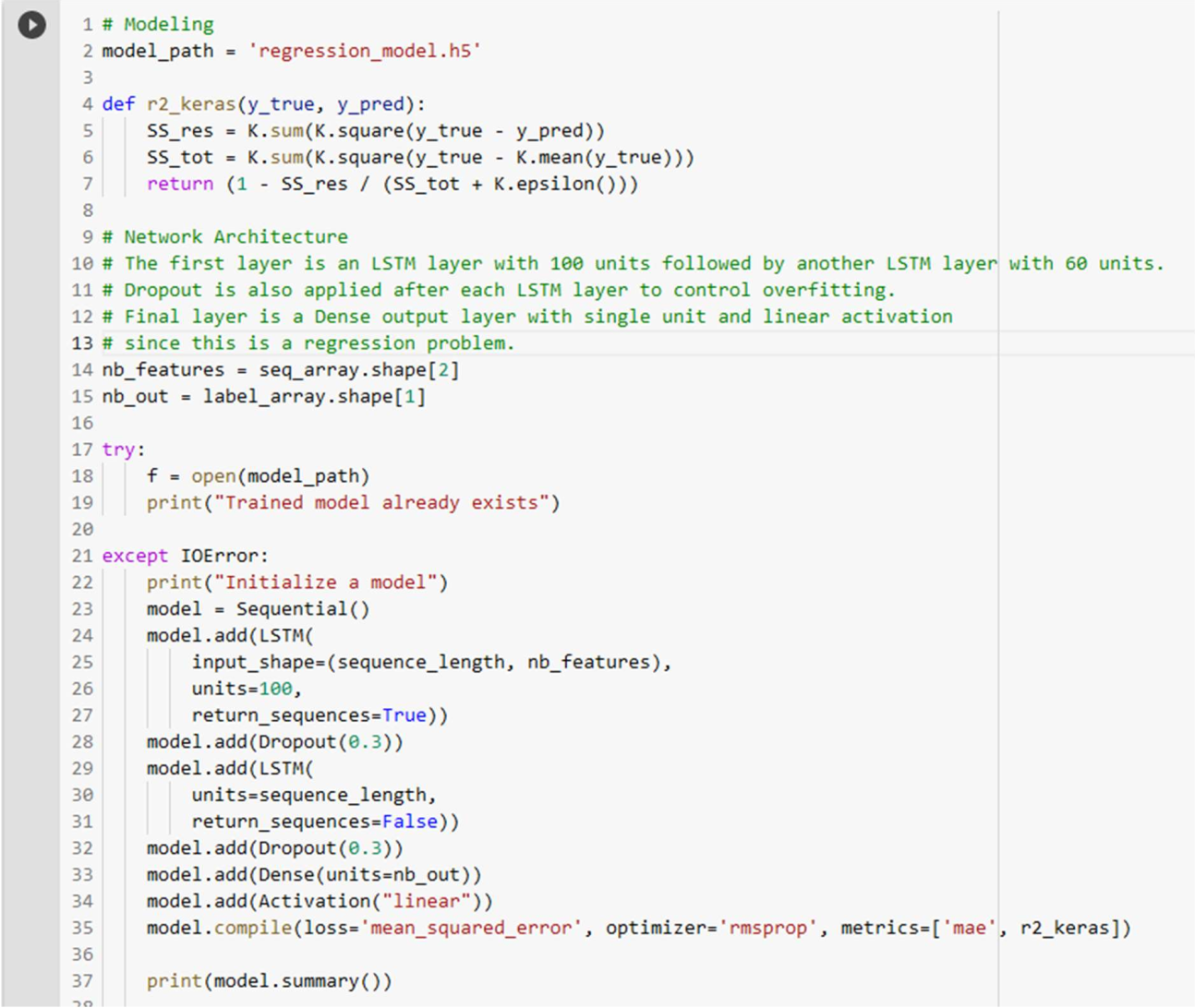
Data values are normalized in order to make data uniform across various engines with different RUL range.



Data is reshaped into a sequence of length 60. Each sequence contains samples, time steps, features. Engine whose values are less than sequence length are omitted, because it cannot be used for prediction. LSTM model takes sequence of input and find RUL for first value of next sequence.

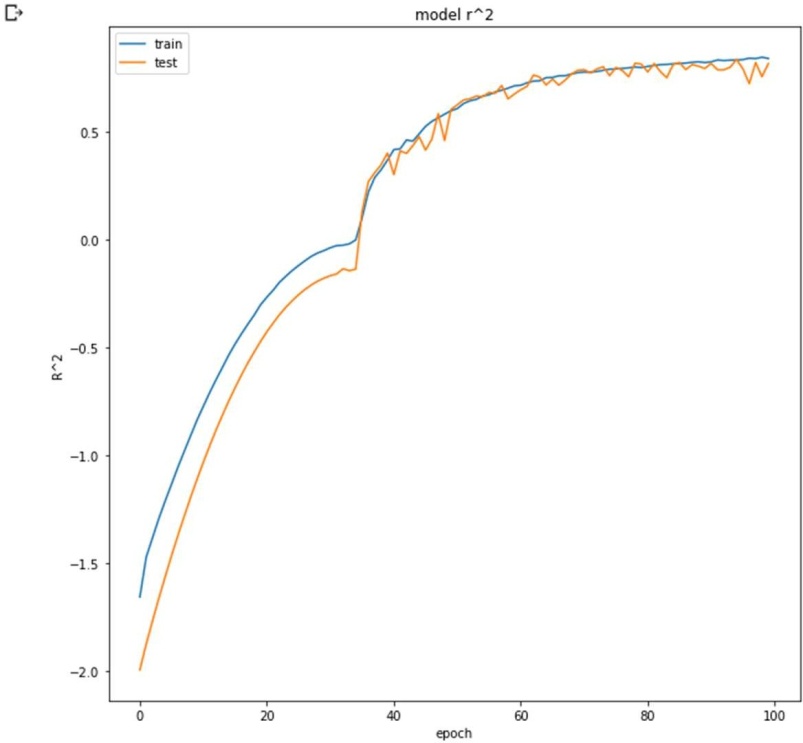
# MODULE-2: BUILDING LSTM MODEL

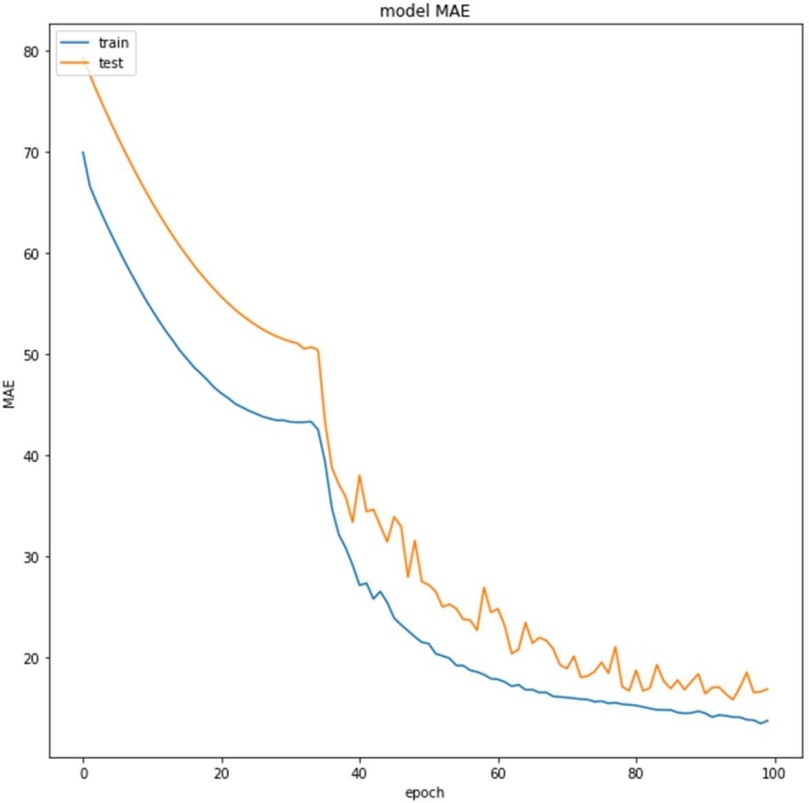
2 layers of LSTM cells are built, with 100 units in the first layer and 50 units in the second layer. In the final layer, sigmoid activation function is included.

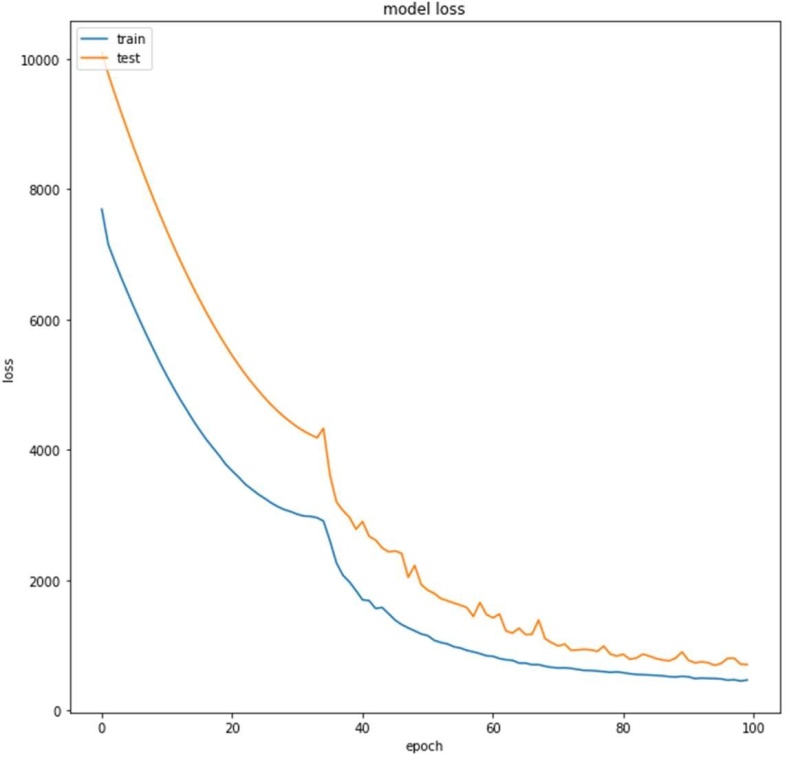


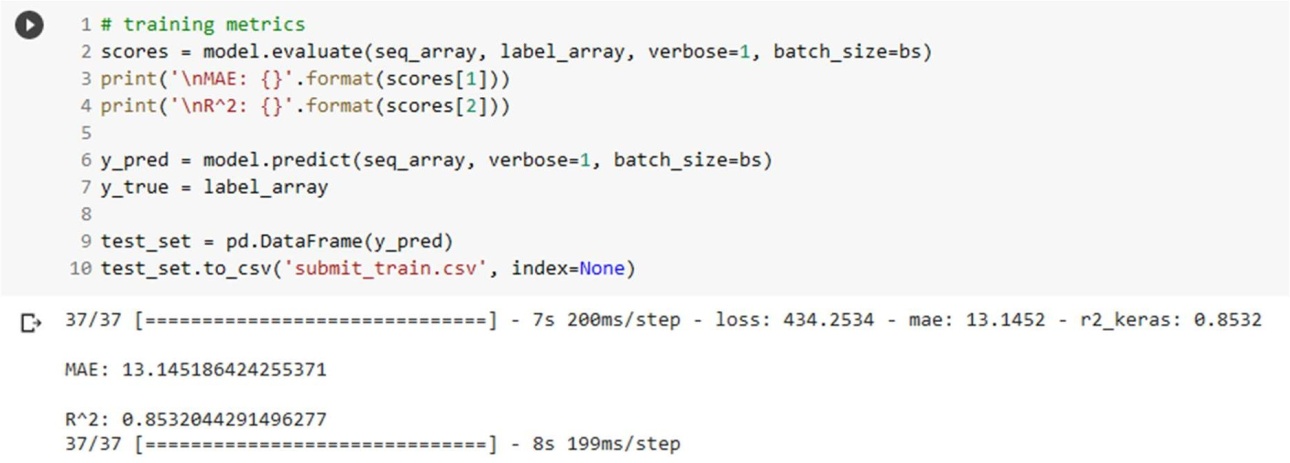
# MODULE-3:TRAINING LSTM MODEL

The model takes the dataset as the input and calculates the Remaining Useful Lifetime. The loss is calculated after each epoch.



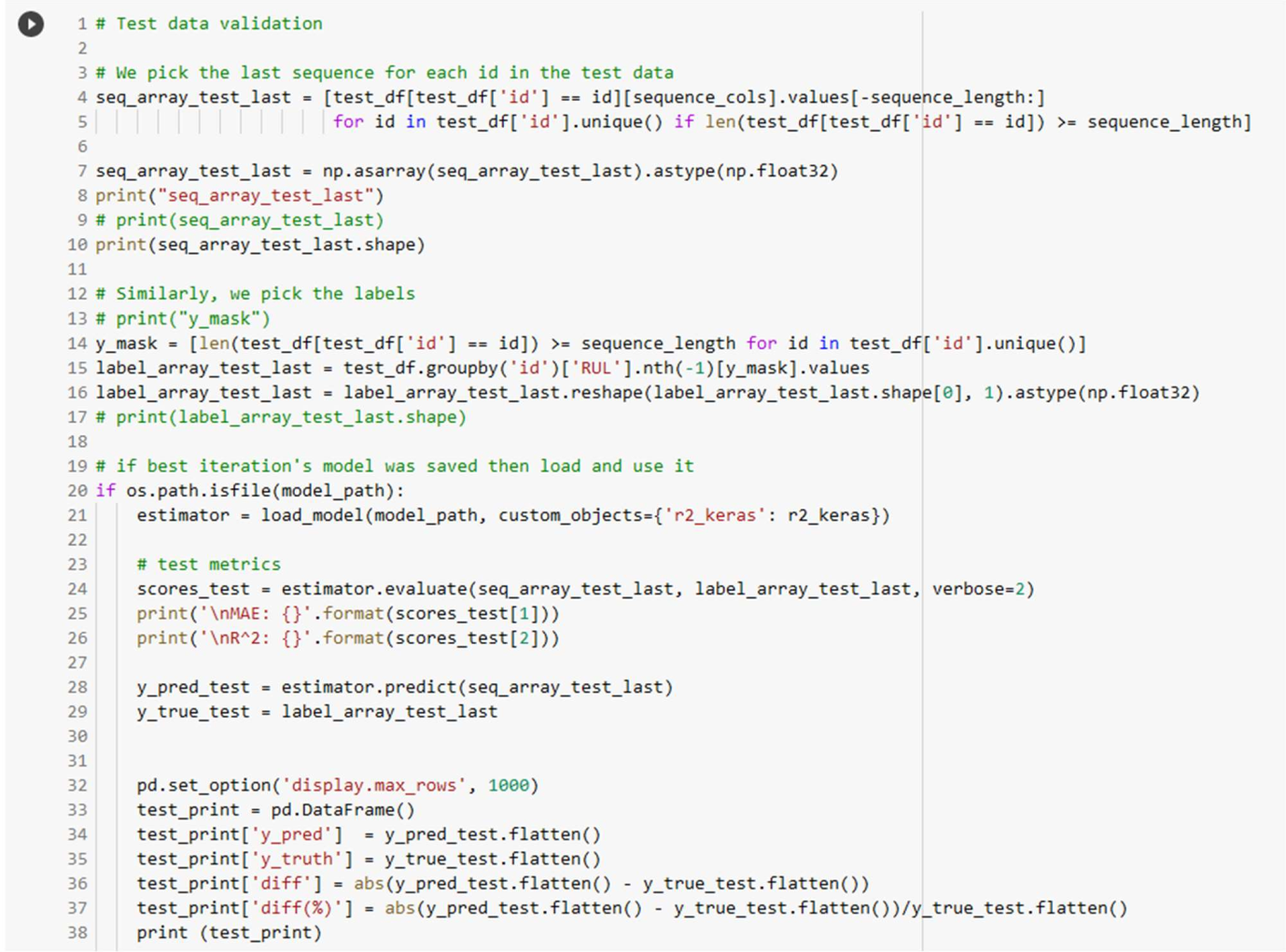


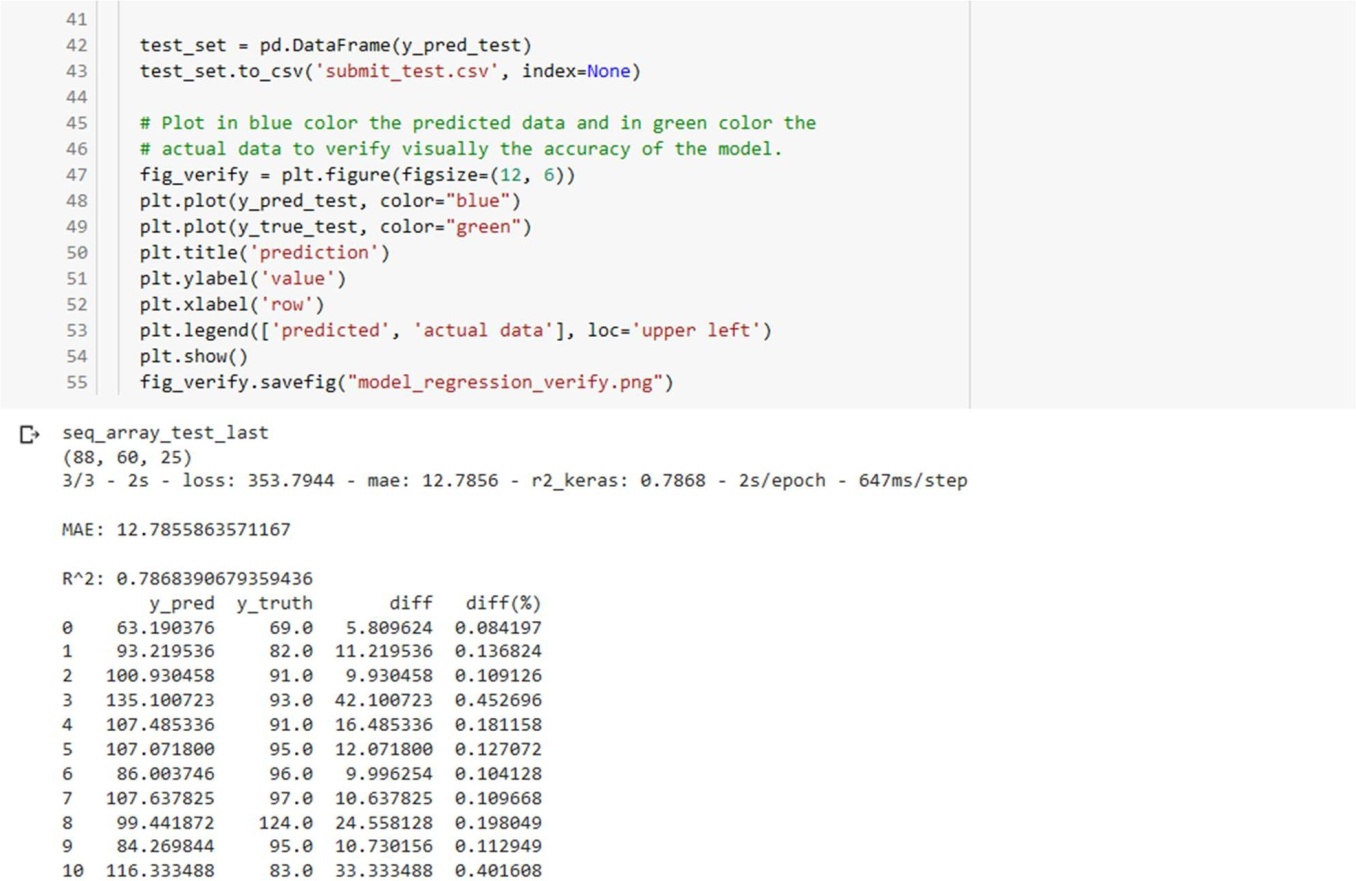


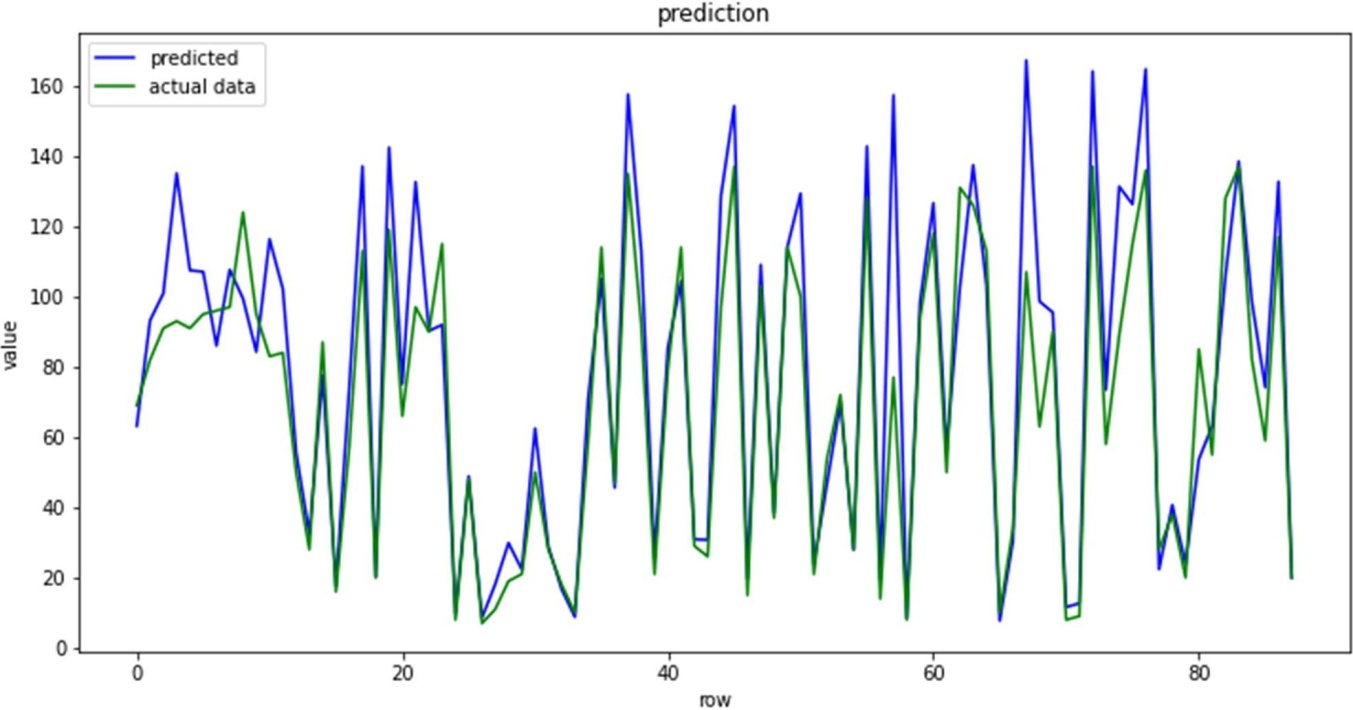


# MODULE-4:TESTING THE MODEL

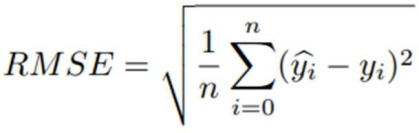
We finally apply the Training dataset whose ground truth Remaining useful lifetime is given in a separate dataset into out trained and optimized LSTM model , we obtain the output for each case of aircraft engine based on the aircraft id. The RUL predicted and actual ground truth is compared to analyze the result.







# PERFORMANCE MEASURES



where y^i and yi are respectively the value predicted by the model and the ground truth provided by the test set of the dataset of the generic i-th input sample, and n is the number of samples of the test set.

# REFERENCES

1. On the use of LSTM networks for Predictive maintenance, in smart industries." - 2019 IEEE International Conference of Smart Computing.
2. Machine Learning for Equipment failure Prediction and Predictive Maintenance (pm] - shad Griffon - Medium.com
3. Towards Data Science - Predictive maintenance of turbofan engines - by Keen Peters
4. Predicting the maintenance of aircraft engines using LSTM - International Journal trend in Scientific research and development. ISSN: 2456-6470
5. LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras –

Ryan – Medium.com

1. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) – Brandon Rohrer
2. Towards Data Science - Illustrated Guide to LSTM’s and GRU’s: A step by step explanation