# Predictive Maintenance of Turbojet Engine

# Problem Statement:--

**Description**

Prognostics and health management is an important topic in industry for predicting state of assets to avoid downtime and failures. This data set is the version of the very well known asset degradation modeling from NASA. It includes Run-to-Failure simulated data from turbo jet engines. Engine degradation simulation was carried out using C-MAPSS. Four different were sets simulated under different combinations of operational conditions and fault modes. Records several sensor channels to characterize fault evolution. The data set was provided by the Prognostics CoE at NASA Ames.

**Prediction Goal**

In this dataset the goal is to predict the remaining useful life (RUL) of each engine in the test dataset. RUL is equivalent of number of flights remained for the engine after the last datapoint in the test dataset.

# Data Description.

The engine is operating normally at the start of each time series, and develops a fault at some point during the series. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure. The objective is to predict Remaining Useful Life (RUL).

The data are provided as a zip-compressed text file with 26 columns of numbers, separated by spaces. Each row is a snapshot of data taken during a single operational cycle, each column is a different variable. The columns correspond to:

**1) unit number**

**2) time, in cycles**

**3) operational setting 1**

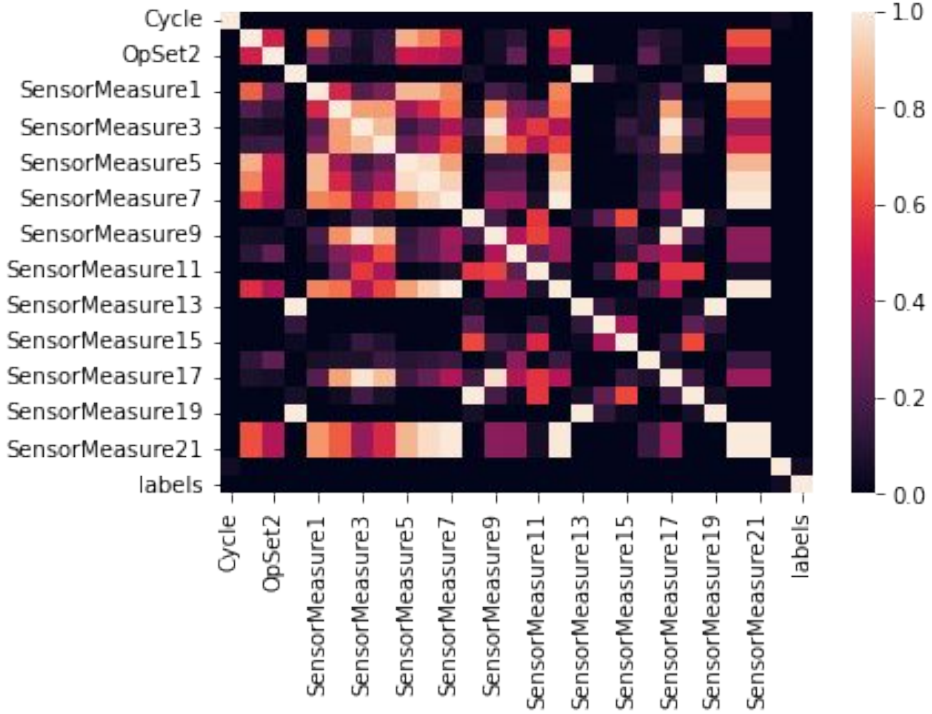
**……..**

**26) operational setting 21**

# Checking data types of the columns

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 160359 entries, 0 to 61248
Data columns (total 26 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   ID                 160359 non-null  int64
 1   Cycle              160359 non-null  int64
 2   OpSet1             160359 non-null  float64
 3   OpSet2             160359 non-null  float64
 4   OpSet3             160359 non-null  float64
 5   SensorMeasure1     160359 non-null  float64
 6   SensorMeasure2     160359 non-null  float64
 7   SensorMeasure3     160359 non-null  float64
 8   SensorMeasure4     160359 non-null  float64
 9   SensorMeasure5     160359 non-null  float64
 10  SensorMeasure6     160359 non-null  float64
 11  SensorMeasure7     160359 non-null  float64
 12  SensorMeasure8     160359 non-null  float64
 13  SensorMeasure9     160359 non-null  float64
 14  SensorMeasure10    160359 non-null  float64
 15  SensorMeasure11    160359 non-null  float64
 16  SensorMeasure12    160359 non-null  float64
 17  SensorMeasure13    160359 non-null  float64
 18  SensorMeasure14    160359 non-null  float64
 19  SensorMeasure15    160359 non-null  float64
 20  SensorMeasure16    160359 non-null  float64
 21  SensorMeasure17    160359 non-null  int64
 22  SensorMeasure18    160359 non-null  int64
 23  SensorMeasure19    160359 non-null  float64
 24  SensorMeasure20    160359 non-null  float64
 25  SensorMeasure21    160359 non-null  float64
dtypes: float64(22), int64(4)
memory usage: 33.0 MB
```

We will display the mutual correlations of the signs on the "heat map", for this we will prepare an additional sign "RUL", showing the number of cycles to failure in the training data



## Classification Problem Formulation

The challenge of this project was to predict the Remaining Useful Life of the engine by using the given sensor's data and operational conditions.

But in this project, we ll try to simplify that by converting it to a Classification Problem. Where, the classes/labels will of 3 types, i.e. **Good Condition, Moderate Condition and Warning Condition**.

**labels corresponding to each conditions**

- Good Condition - 0
- Moderate Condition - 1
- Warning Condition - 2

## Defining the labels

Here we will define the engine's condition with **Life Ratio (LR)**, which is the ratio between **Current Cycle** and the **End cycle/ End of Life(EOL)**. If **LR=0**, that means the component has just started its degradation and **LR=1** means, it is completely degraded.

- if LR <= 0.6 - Good Condition
- if 0.6 < LR <= 0.8 - Moderate Condition
- if 0.8 < LR - Warning Condition

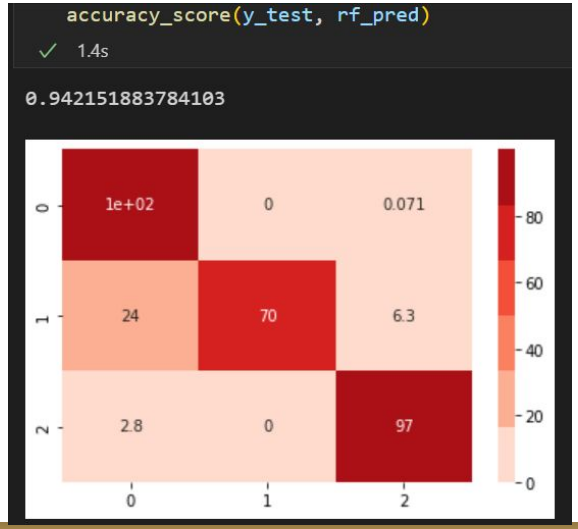
# Model training

Model Training was done with two machine learning classification models

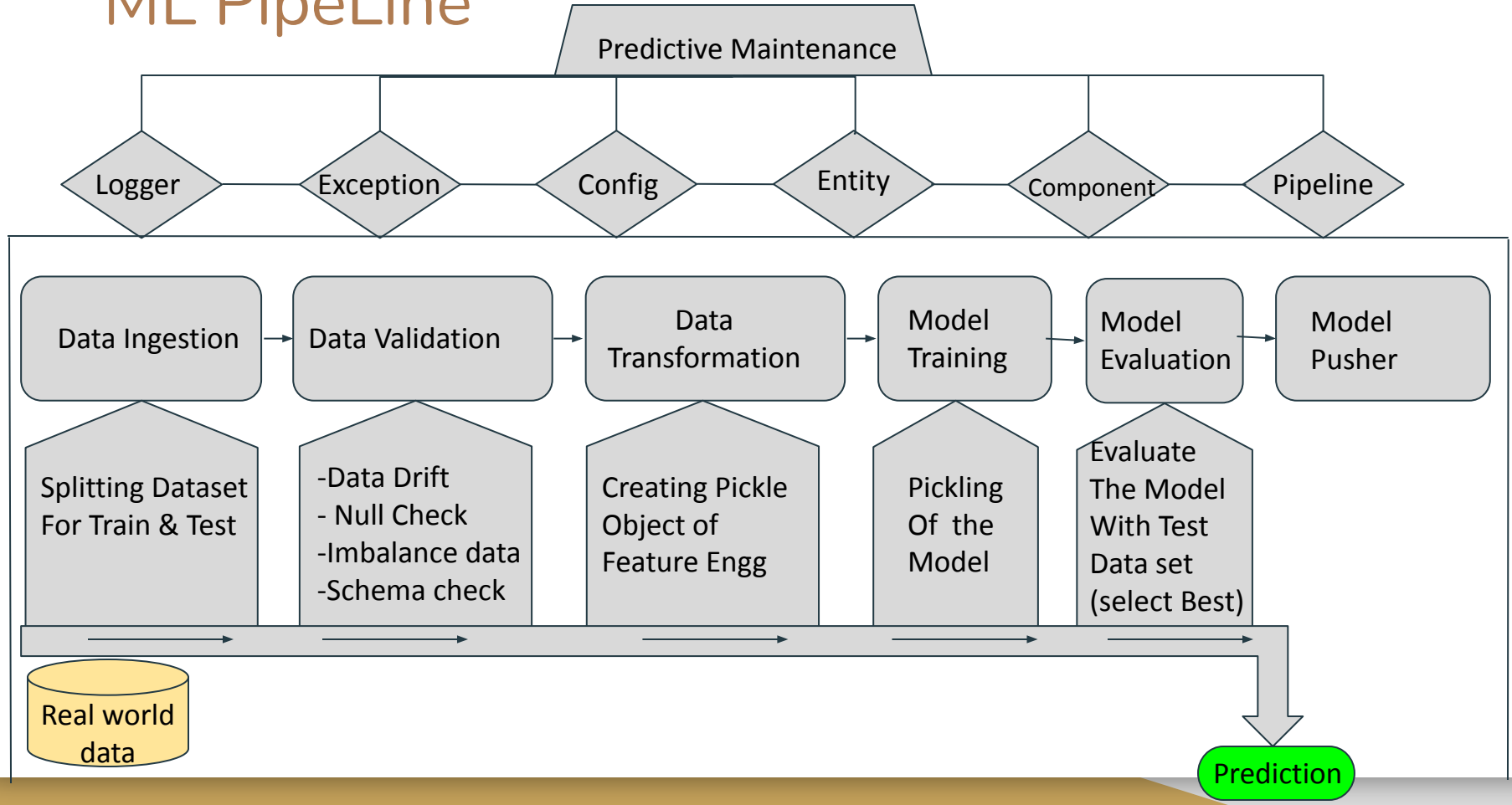
-Random Forest

-Logistic regression

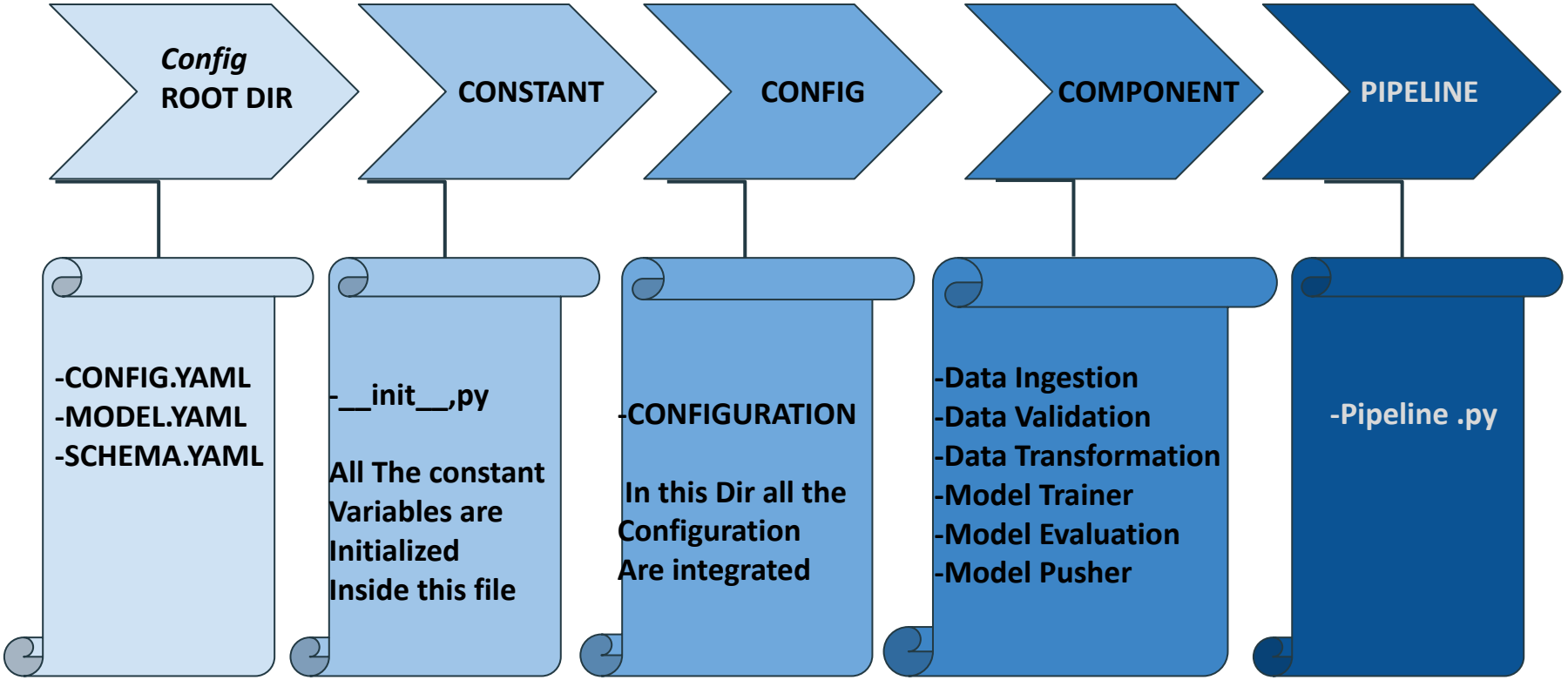
From this 2 model Random forest gave the better performance while training and testing



# ML PipeLine


Predictive Maintenance
Logger
Exception
Config
Entity
Component
Pipeline
Data Ingestion
Data Validation
Data Transformation
Model Training
Model Evaluation
Model Pusher
Splitting Dataset For Train & Test
-Data Drift
- Null Check
-Imbalance data
-Schema check
Creating Pickle Object of Feature Engg
Pickling Of the Model
Evaluate The Model With Test Data set (select Best)
Real world data
Prediction

# Code Flow


Config
ROOT DIR
CONSTANT
CONFIG
COMPONENT
PIPELINE
-CONFIG.YAML
-MODEL.YAML
-SCHEMA.YAML
-__init__,py
All The constant
Variables are
Initialized
Inside this file
-CONFIGURATION
In this Dir all the
Configuration
Are integrated
-Data Ingestion
-Data Validation
-Data Transformation
-Model Trainer
-Model Evaluation
-Model Pusher
-Pipeline .py

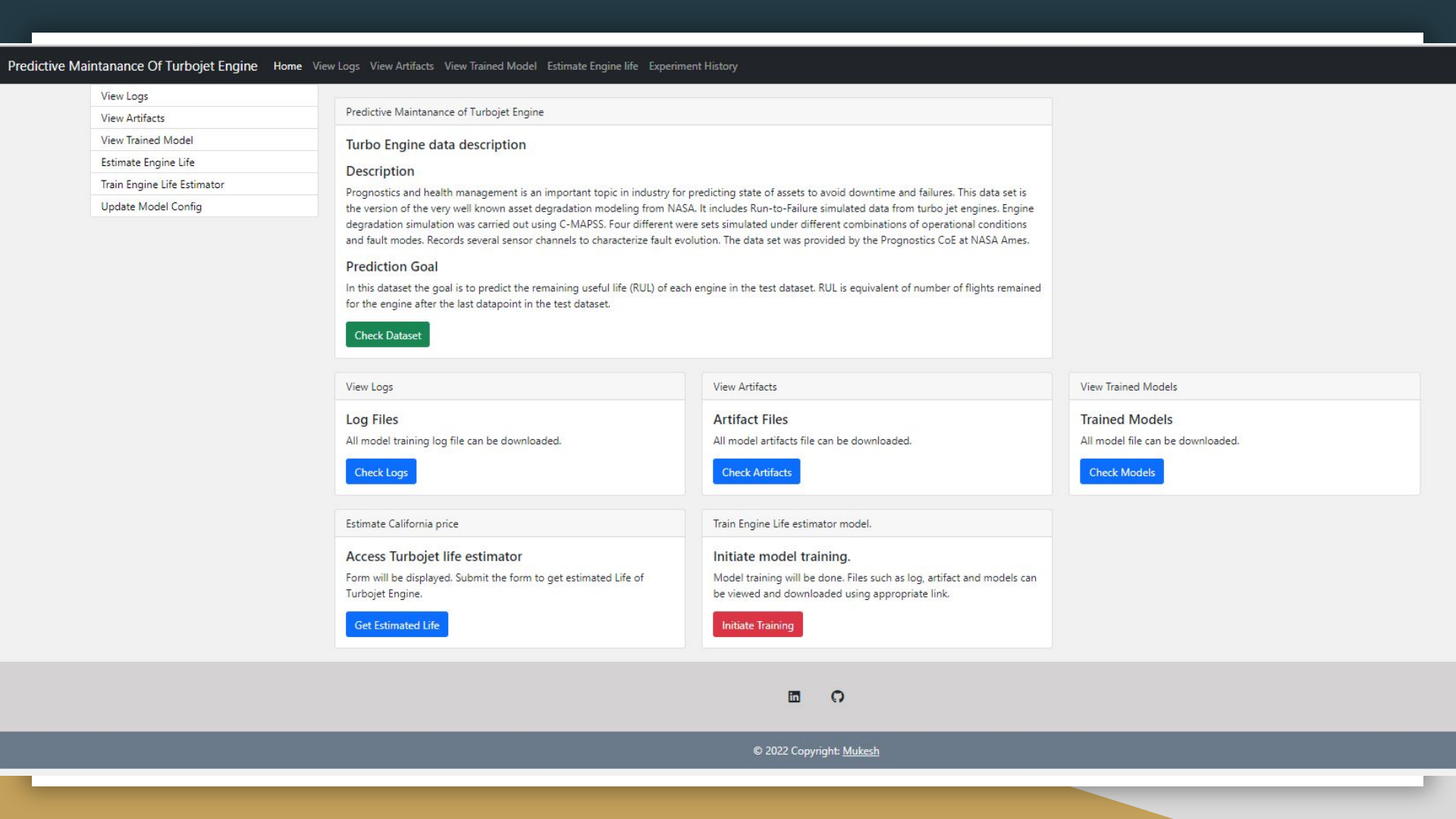
# Data Set Drift Monitoring

Drift is detected for 0.00% of features (0 out of 27). Dataset Drift is NOT detected.

Search

| Feature | Type | Reference Distribution | Current Distribution | Data Drift | Stat Test | Drift Score |
|---|---|---|---|---|---|---|
| Cycle | num |  |  | Not Detected | Wasserstein distance (normed) | 0.005307 |
| OpSet2 | num |  |  | Not Detected | Wasserstein distance (normed) | 0.004958 |
| Life_Ratio | num |  |  | Not Detected | Wasserstein distance (normed) | 0.004953 |
| SensorMeasure11 | num |  |  | Not Detected | Wasserstein distance (normed) | 0.004444 |
| SensorMeasure4 | num |  |  | Not Detected | Wasserstein distance (normed) | 0.004272 |
| SensorMeasure17 | num |  |  | Not Detected | Wasserstein distance (normed) | 0.004241 |
| SensorMeasure14 | num |  |  | Not Detected | Wasserstein distance (normed) | 0.004182 |
| SensorMeasure10 | num |  |  | Not Detected | Wasserstein distance (normed) | 0.00416 |
| SensorMeasure5 | num |  |  | Not Detected | Wasserstein distance (normed) | 0.004154 |
| SensorMeasure6 | num |  |  | Not Detected | Wasserstein distance (normed) | 0.00414 |

10 rows 1-10 of 27


Predictive Maintanance Of Turbojet Engine
Home
View Logs
View Artifacts
View Trained Model
Estimate Engine life
Experiment History
View Logs
View Artifacts
View Trained Model
Estimate Engine Life
Train Engine Life Estimator
Update Model Config
Predictive Maintanance of Turbojet Engine
Turbo Engine data description
Description
Prognostics and health management is an important topic in industry for predicting state of assets to avoid downtime and failures. This data set is the version of the very well known asset degradation modeling from NASA. It includes Run-to-Failure simulated data from turbo jet engines. Engine degradation simulation was carried out using C-MAPSS. Four different were sets simulated under different combinations of operational conditions and fault modes. Records several sensor channels to characterize fault evolution. The data set was provided by the Prognostics CoE at NASA Ames.
Prediction Goal
In this dataset the goal is to predict the remaining useful life (RUL) of each engine in the test dataset. RUL is equivalent of number of flights remained for the engine after the last datapoint in the test dataset.
Check Dataset
View Logs
Log Files
All model training log file can be downloaded.
Check Logs
View Artifacts
Artifact Files
All model artifacts file can be downloaded.
Check Artifacts
View Trained Models
Trained Models
All model file can be downloaded.
Check Models
Estimate California price
Access Turbojet life estimator
Form will be displayed. Submit the form to get estimated Life of Turbojet Engine.
Get Estimated Life
Train Engine Life estimator model.
Initiate model training.
Model training will be done. Files such as log, artifact and models can be viewed and downloaded using appropriate link.
Initiate Training
© 2022 Copyright: Mukesh