

AirMs Services User Manual

- *A Complete Solution to
Aircraft Predictive
Maintenance*



DECEMBER 14

AirMs

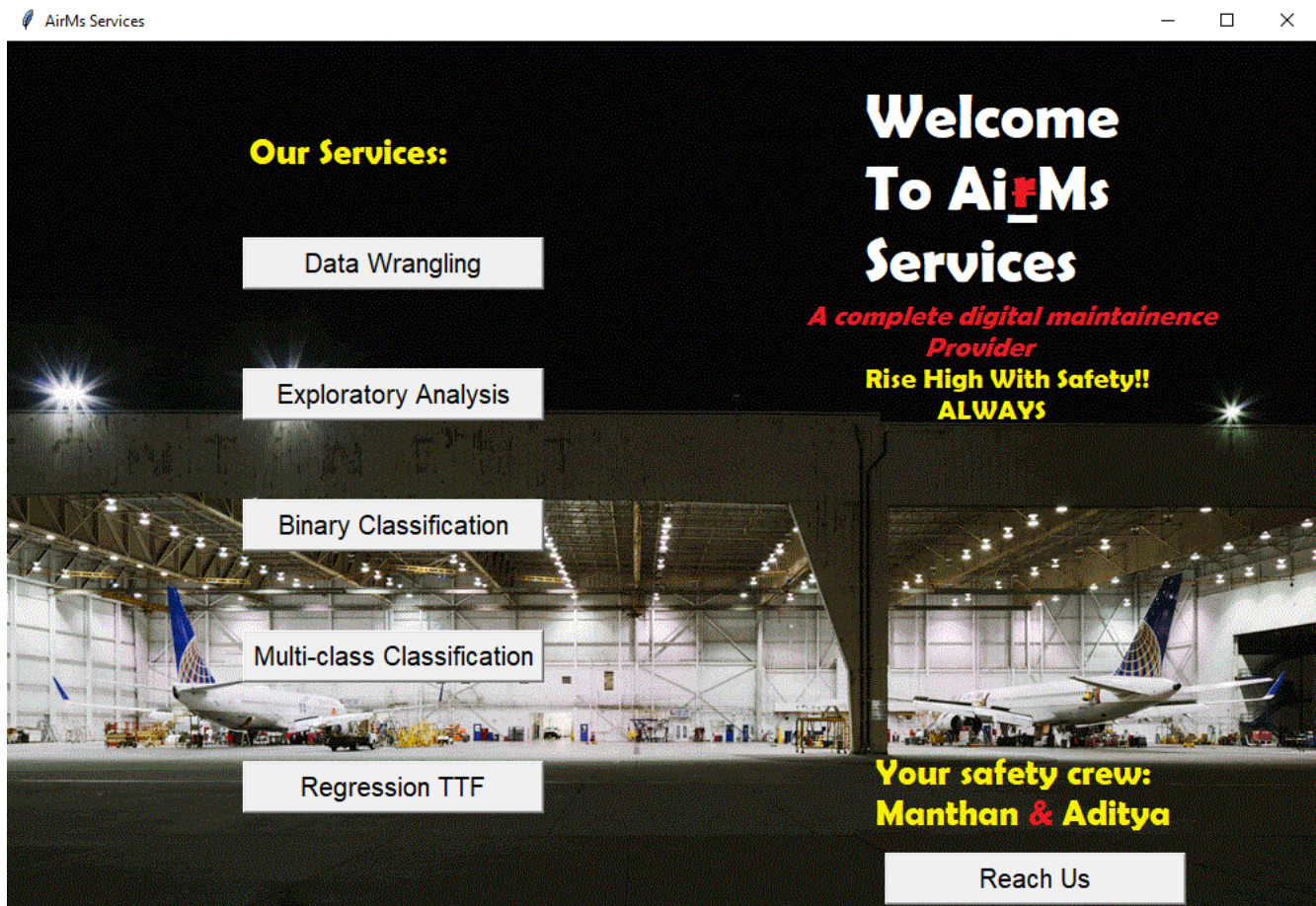
Authored by: Manthan Dhisale
Aditya Iyenger

AirMs

Index

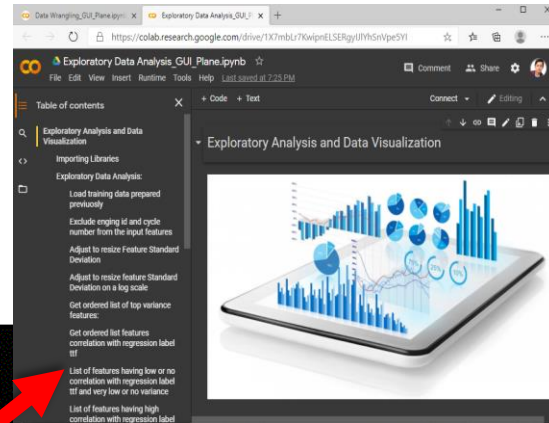
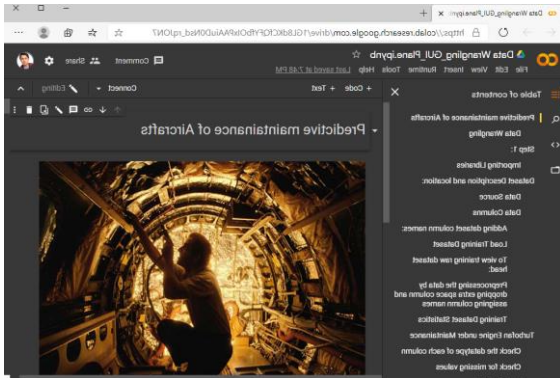
1. Graphic User Interface.....	3
2. Interface Breakdown.....	4
3. Data Wrangling GUI.....	5
4. Exploratory Analysis GUI.....	7
5. TTF (Time To Failure Prediction).....	9
6. Binary Classification for predicting failure.....	10
7. Multi-class Classification for predicting the failure class.....	11
8. Our Unique Selling Propositions.....	12

Graphic User Interface



A complete user-easy and interactive interface, through which a User can access to 5 Google Colab Notebooks online.

Also Users can reach the Safety Crew (AirMs Leaders) through Reach Us features, where they can drop a mail or view the leaders profile.



Our Services:

Data Wrangling

Exploratory Analysis

Binary Classification

Multi-class Classification

Regression TTF

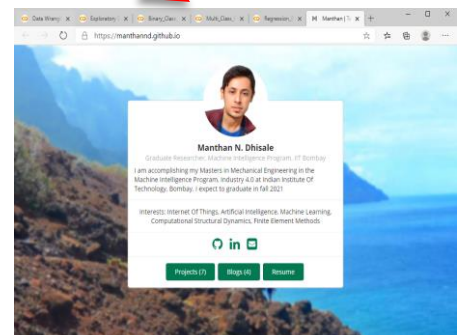
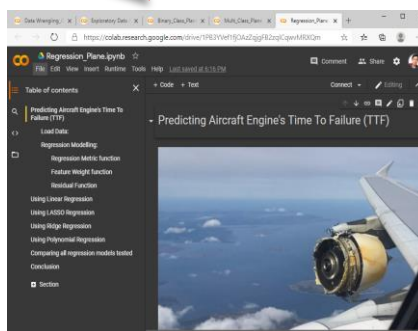
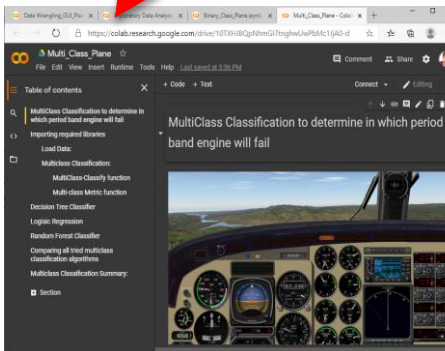
To AirMs Services

A complete digital maintenance Provider

Rise High With Safety!!

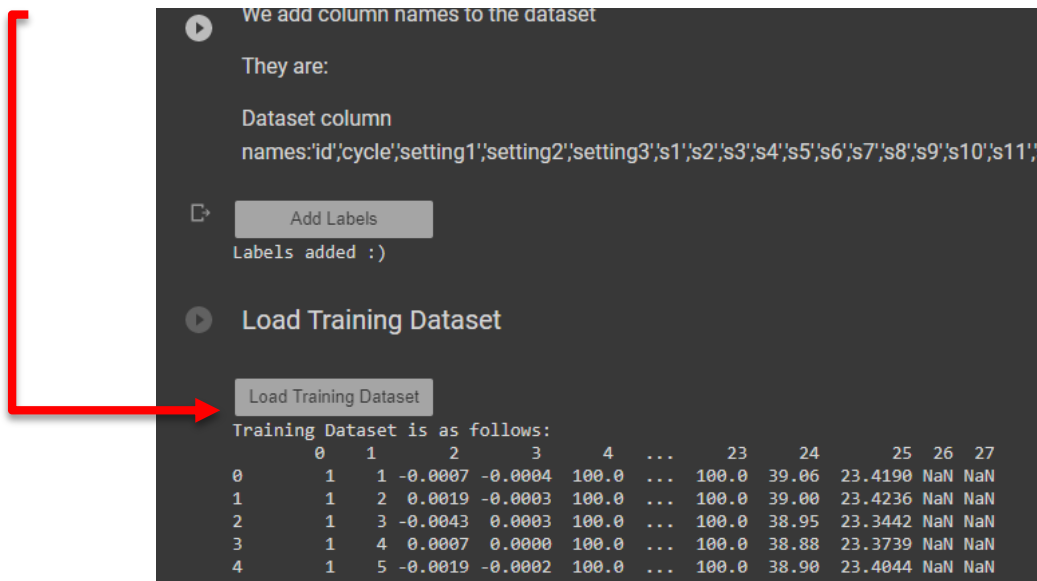
Your safety crew: Manthan & Aditya

Reach Us



Data Wrangling (A Code free Interface with Colab Form Features for GUI)

1. Colab Buttons for Loading data and training:



We add column names to the dataset

They are:

Dataset column
names:'id','cycle','setting1','setting2','setting3','s1','s2','s3','s4','s5','s6','s7','s8','s9','s10','s11';

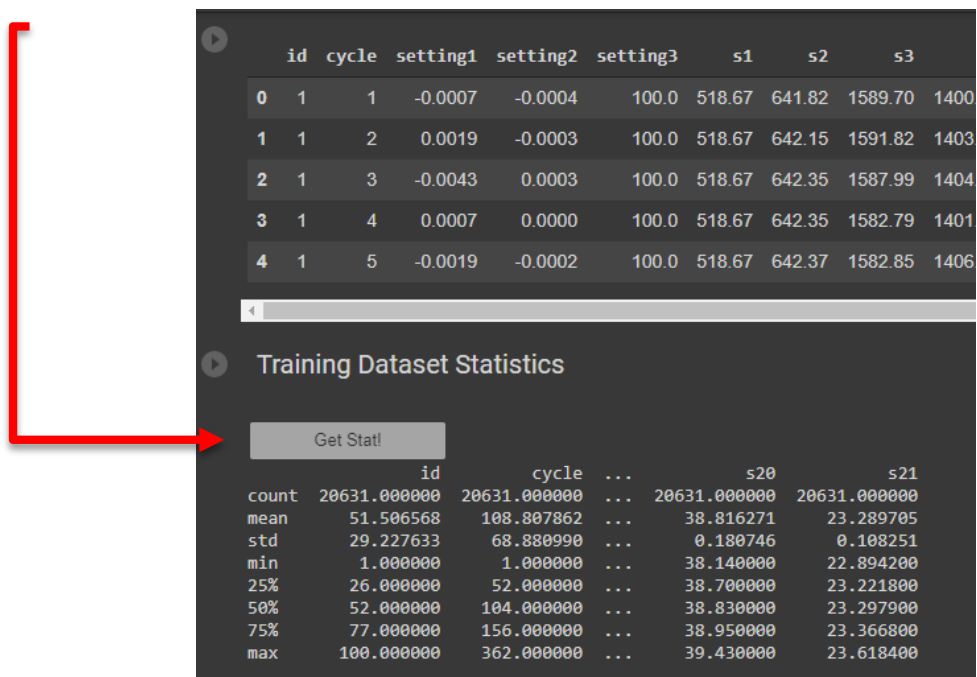
Labels added :)

Load Training Dataset

Training Dataset is as follows:

	0	1	2	3	4	...	23	24	25	26	27
0	1	1	-0.0007	-0.0004	100.0	...	100.0	39.06	23.4190	NaN	NaN
1	1	2	0.0019	-0.0003	100.0	...	100.0	39.00	23.4236	NaN	NaN
2	1	3	-0.0043	0.0003	100.0	...	100.0	38.95	23.3442	NaN	NaN
3	1	4	0.0007	0.0000	100.0	...	100.0	38.88	23.3739	NaN	NaN
4	1	5	-0.0019	-0.0002	100.0	...	100.0	38.90	23.4044	NaN	NaN

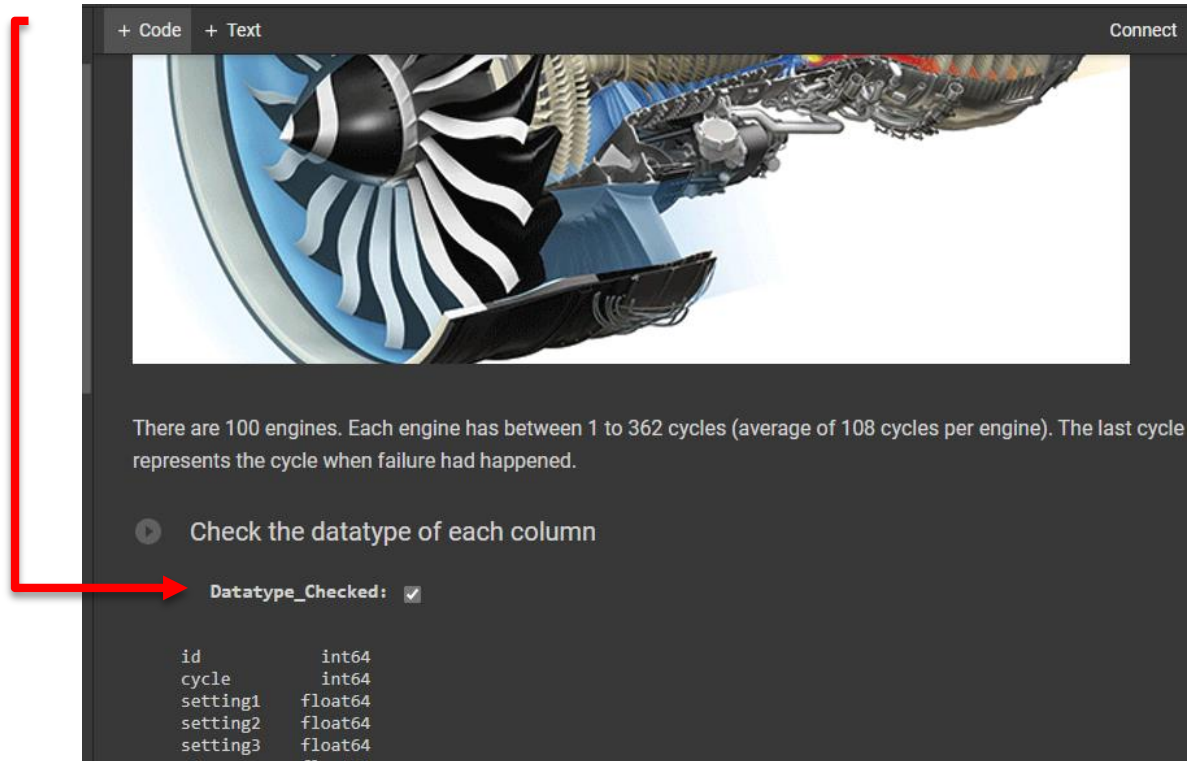
2. Stat Button:



Training Dataset Statistics

	id	cycle	...	s20	s21
count	20631.000000	20631.000000	...	20631.000000	20631.000000
mean	51.506568	108.807862	...	38.816271	23.289705
std	29.227633	68.880990	...	0.180746	0.108251
min	1.000000	1.000000	...	38.140000	22.894200
25%	26.000000	52.000000	...	38.700000	23.221800
50%	52.000000	104.000000	...	38.830000	23.297900
75%	77.000000	156.000000	...	38.950000	23.366800
max	100.000000	362.000000	...	39.430000	23.618400

3. Checkbox decisions:

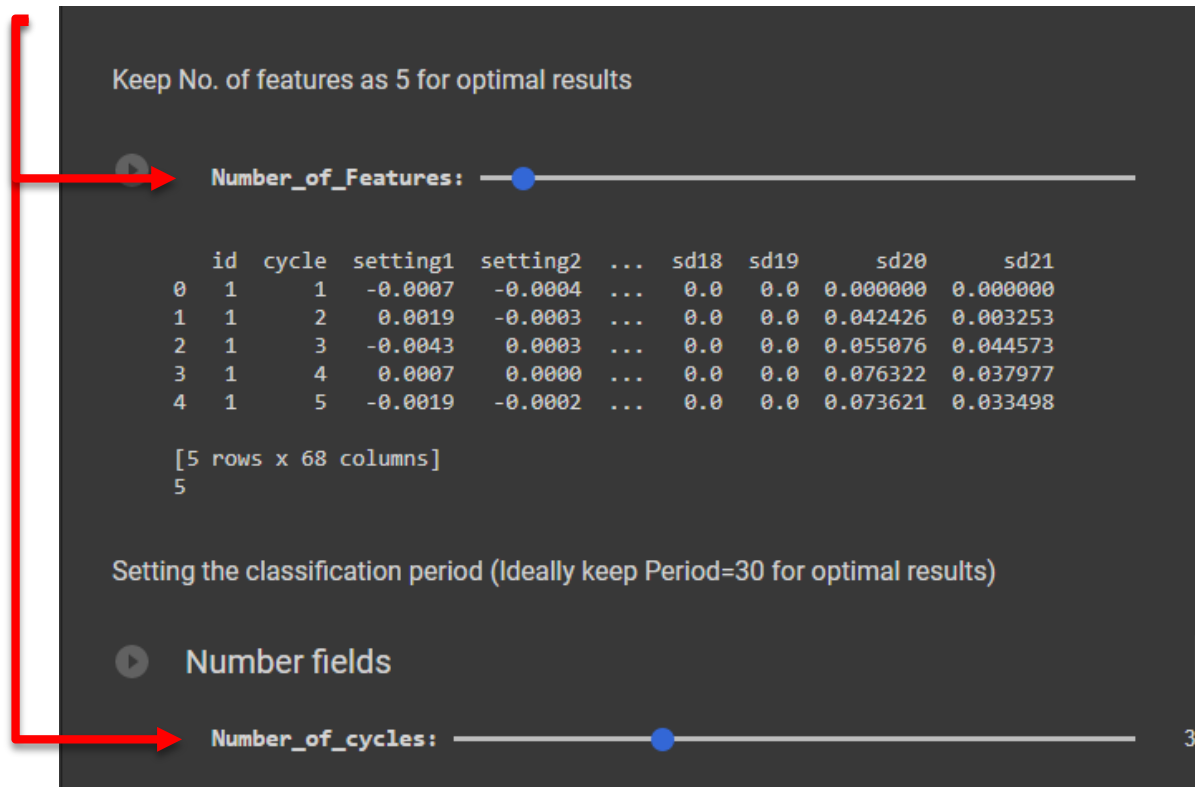


The interface shows a 3D model of an engine component. Below the model, there is a text box explaining the data: "There are 100 engines. Each engine has between 1 to 362 cycles (average of 108 cycles per engine). The last cycle represents the cycle when failure had happened."

A play button icon is followed by the text "Check the datatype of each column". Below this, a checkbox labeled "Datatype_Checked:" is checked. A red arrow points to this checkbox.

id	cycle	setting1	setting2	setting3
1	1	0.0007	-0.0004	0.0000
1	2	0.0019	-0.0003	0.0000
1	3	-0.0043	0.0003	0.0000
1	4	0.0007	0.0000	0.0000
1	5	-0.0019	-0.0002	0.0000

4. Scroll feature to define no. of cycles and no. of features:



The interface shows a data table with 5 rows and 68 columns. A red arrow points to the "Number_of_Features:" slider, which is set to 5. Below the table, it says "[5 rows x 68 columns]".

Setting the classification period (Ideally keep Period=30 for optimal results)

A play button icon is followed by the text "Number fields". Below this, a slider labeled "Number_of_cycles:" is set to 3. A red arrow points to this slider.

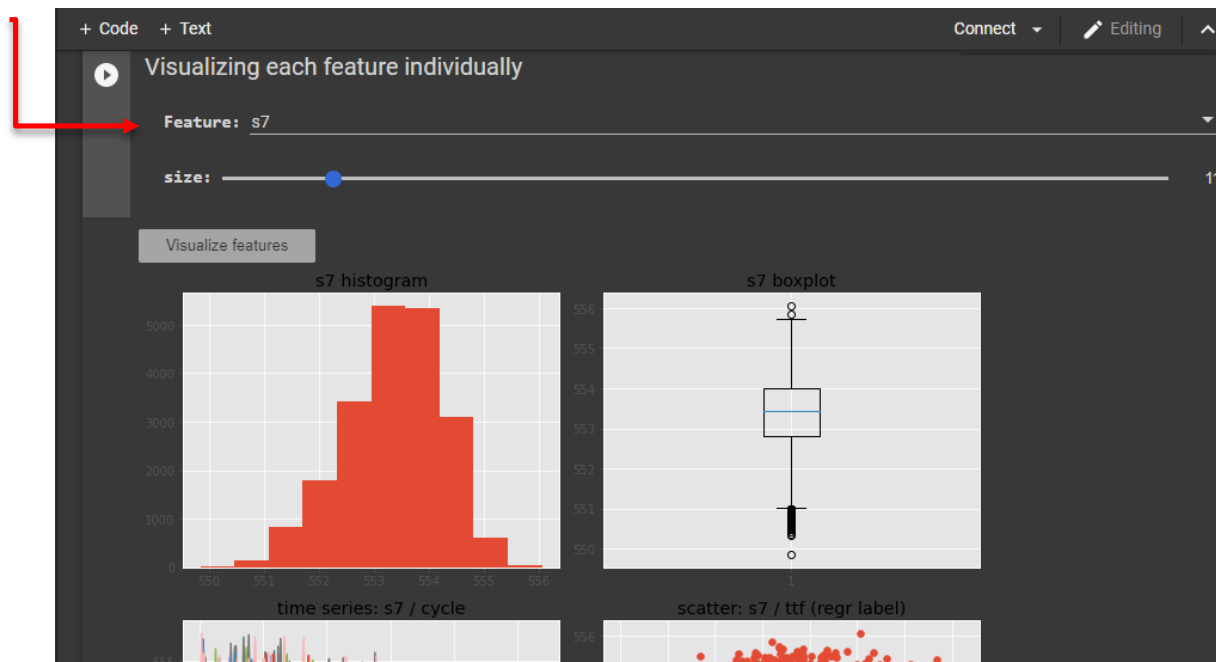
	id	cycle	setting1	setting2	...	sd18	sd19	sd20	sd21
0	1	1	-0.0007	-0.0004	...	0.0	0.0	0.000000	0.000000
1	1	2	0.0019	-0.0003	...	0.0	0.0	0.042426	0.003253
2	1	3	-0.0043	0.0003	...	0.0	0.0	0.055076	0.044573
3	1	4	0.0007	0.0000	...	0.0	0.0	0.076322	0.037977
4	1	5	-0.0019	-0.0002	...	0.0	0.0	0.073621	0.033498

Exploratory Analysis and Data Visualizations

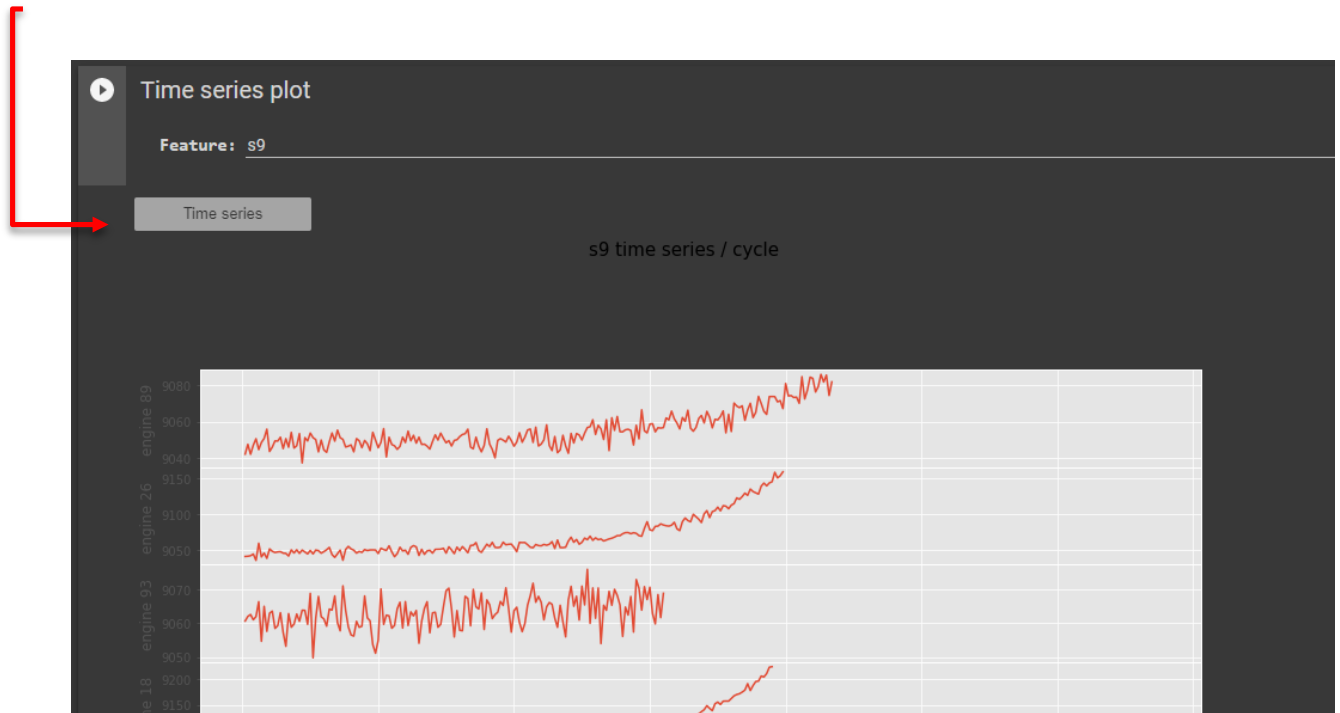
1. Scroll based Hyper-parameter Selection and plot size:



2. Selecting Sensor and getting the data trends

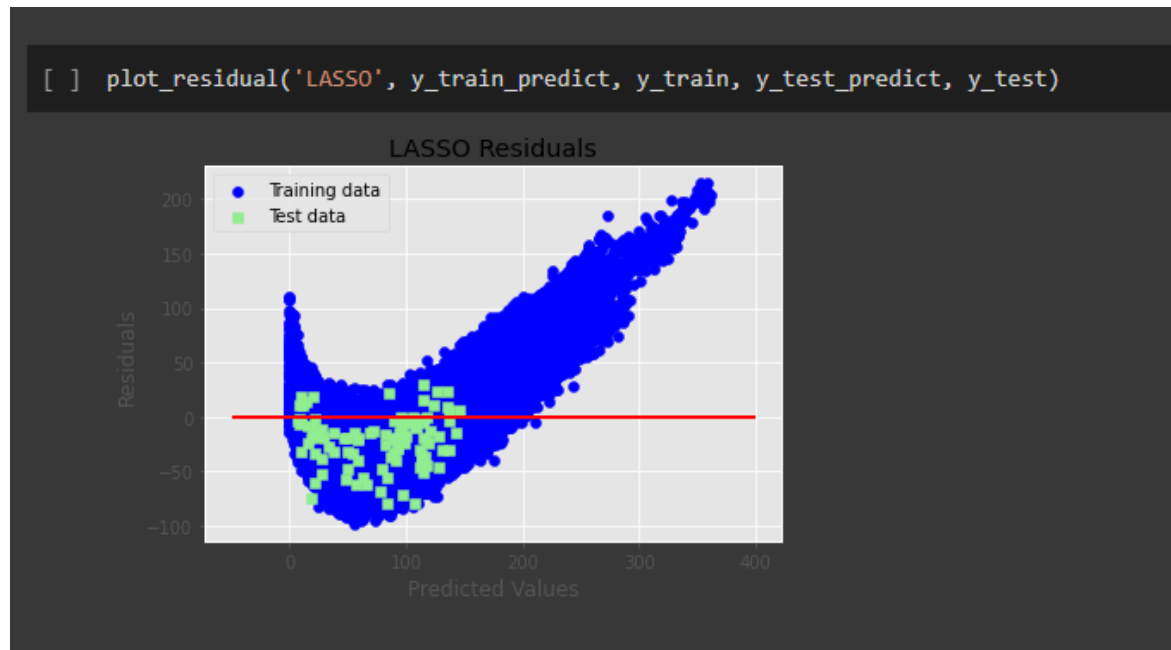


3. Time Plots button access



Time To Failure Prediction using Regression model

1. Visualization comparison of Train/Test



2. Performance comparison using various models

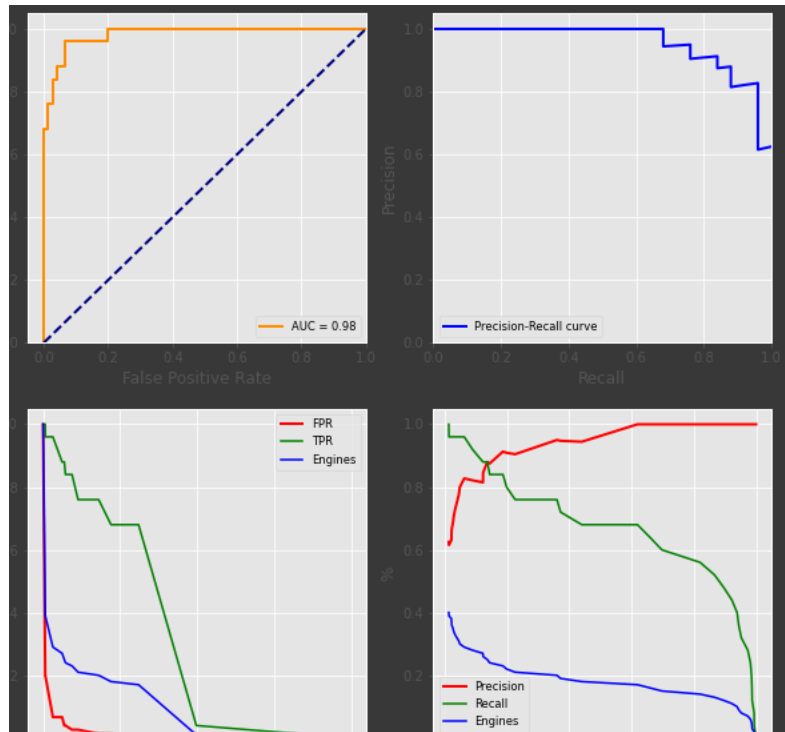
▾ Comparing all regression models tested

```
[ ] #regression metrics comparison before feature engineering
reg_metrics_bfe = pd.concat([linreg_metrics, lasso_metrics, rdg_metrics, polyreg_metrics], axis=1)
reg_metrics_bfe
```

	Linear Regression	LASSO	Ridge Regression	Polynomial Regression
Root Mean Squared Error	32.041095	31.966099	31.965740	32.530030
Mean Absolute Error	25.591780	25.551808	25.544620	24.944453
R^2	0.405495	0.408275	0.408289	0.387213
Explained Variance	0.665297	0.668206	0.667607	0.633327

Failure Prediction using Binary Classifier

1. Visualization Plots of Precision and Recall



2. Performance Comparison of various models

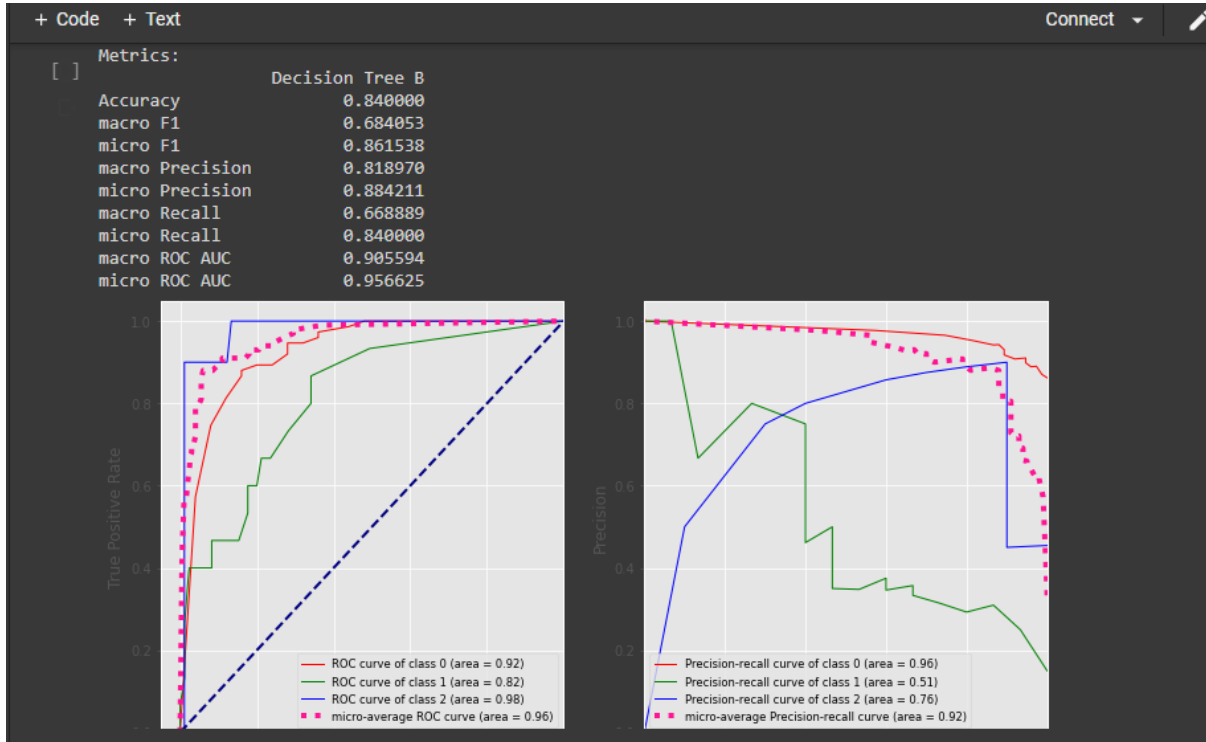
Comparing all Binary Classification algorithms tested

```
#compare all models
metrics_bn = pd.concat([metrics_lgr, metrics_dtr, metrics_rfc], axis=1)
metrics_bn
```

	Logistic Regression B	Logistic Regression A	Decision Tree B	Decision Tree A	Random Forest B	Random Forest A
Accuracy	0.910000	0.920000	0.880000	0.920000	0.910000	0.910000
Precision	1.000000	1.000000	0.933333	0.947368	0.944444	0.944444
Recall	0.640000	0.680000	0.560000	0.720000	0.680000	0.680000
F1 Score	0.780488	0.809524	0.700000	0.818182	0.790698	0.790698
ROC AUC	0.979733	0.981867	0.945067	0.962933	0.980267	0.982400

Failure Prediction using Multi-class Classifier

1. Visualization Plots of Precision and Recall



2. Performance Comparison of various models

Comparing all tried multiclass classification algorithms

```
[ ] #compare all models
metrics_mc = pd.concat([metrics_lgr, metrics_dtr, metrics_rfc], axis=1).T
metrics_mc
```

	Accuracy	macro F1	micro F1	macro Precision	micro Precision	macro Recall	micro Recall	macro ROC AUC	micro ROC AUC
Logistic Regression B	0.81	0.553947	0.835052	0.549455	0.861702	0.562222	0.81	0.945027	0.970900
Logistic Regression A	0.81	0.551750	0.852632	0.586919	0.900000	0.533333	0.81	0.941515	0.971800
Decision Tree B	0.84	0.684053	0.861538	0.818970	0.884211	0.668889	0.84	0.905594	0.956625
Decision Tree A	0.84	0.607906	0.857143	0.852146	0.875000	0.651111	0.84	0.949857	0.973550
Random Forest B	0.82	0.612536	0.854167	0.776749	0.891304	0.573333	0.82	0.964340	0.978550
Random Forest A	0.85	0.705759	0.867347	0.800813	0.885417	0.662222	0.85	0.967744	0.980600

Our Unique Selling Propositions

Our Predictive Maintenance Services:

Data Wrangling

Exploratory Analysis

Binary Classification (Whether Engines fail)

Multi-class Classification (Failure prediction for 3 periods)

Regression Modeling (Predicting Engines TTF)

***Your safety crew:
Manthan & Aditya***



***RISE HIGH WITH SAFETY
ALWAYS!!***