Predictive Maintenance for NASA Turbofan Jet Engines

Introduction to Data Science EN.553.636

Team Members:

Aswath Sivakumar

Dheeraj Dhanvee Kairamkonda

Priyanka Kotha

Krishnan Venkataraman

What is Predictive Maintenance?

- Data-driven, proactive methods – Designed to analyse condition of Machines/Equipments
- Predict – ‘when maintenance should be performed’ – so that it is optimum
- Predict possible failure before they occur

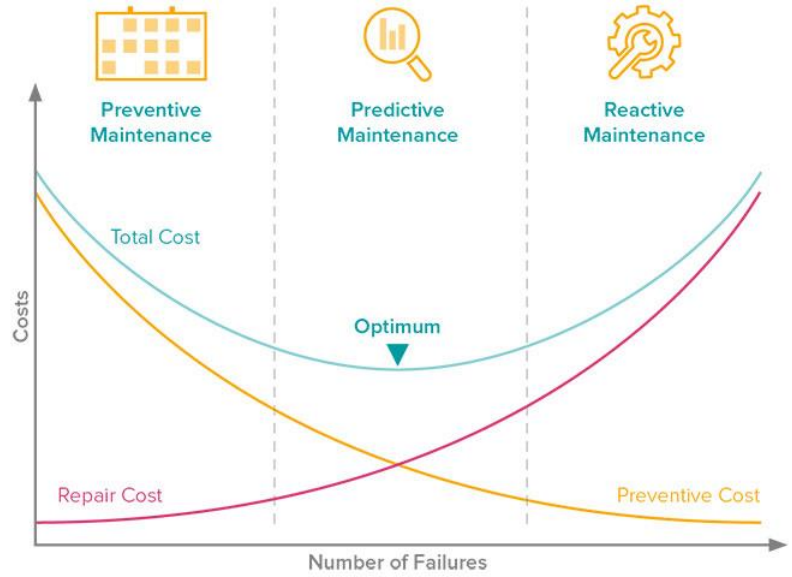


Figure referenced from [4]

Goal of our Project

- Provide predictive maintenance-based solutions for NASA turbofan jet engines
- We have taken 4 problem statements – Analysed them and tried to provide solutions for the same

Dataset

- NASA Turbo fan Jet engine degradation dataset
- 4 different sets generated over 100 different jet engines under different combinations of operational conditions and fault modes
- In all the four datasets – Jet engines are run till end of life. ‘N’ number of cycles unique for each jet engine depending on its wear and tear from starting

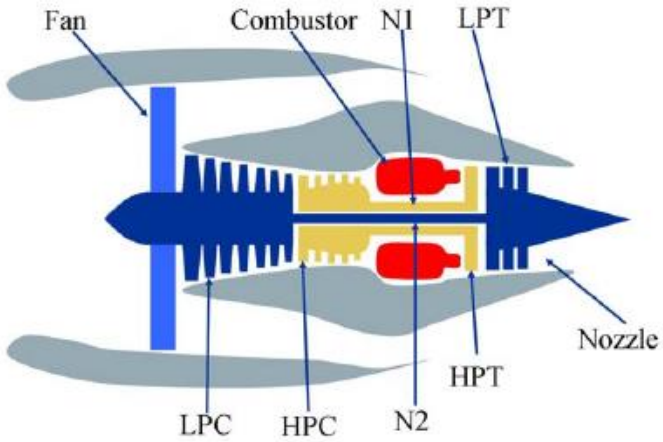


Figure showing simplified NASA turbofan jet engine - referenced from [3]

Concatenated Main dataset contains:

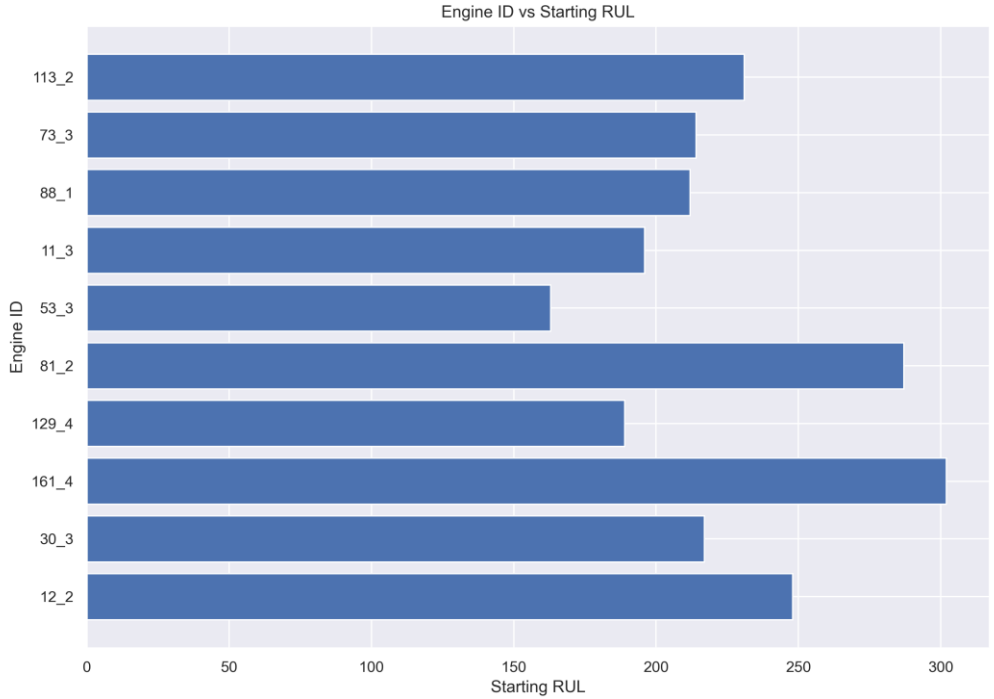
- **<u>27 Columns</u>** – Engine ID, Cycle number, Operation settings 1-3, **sensor measurements 1-21** , RUL(Remaining Useful Life) – Number of cycles remaining till failure from this current cycle
- **<u>160,359 Rows</u> -** Number of records

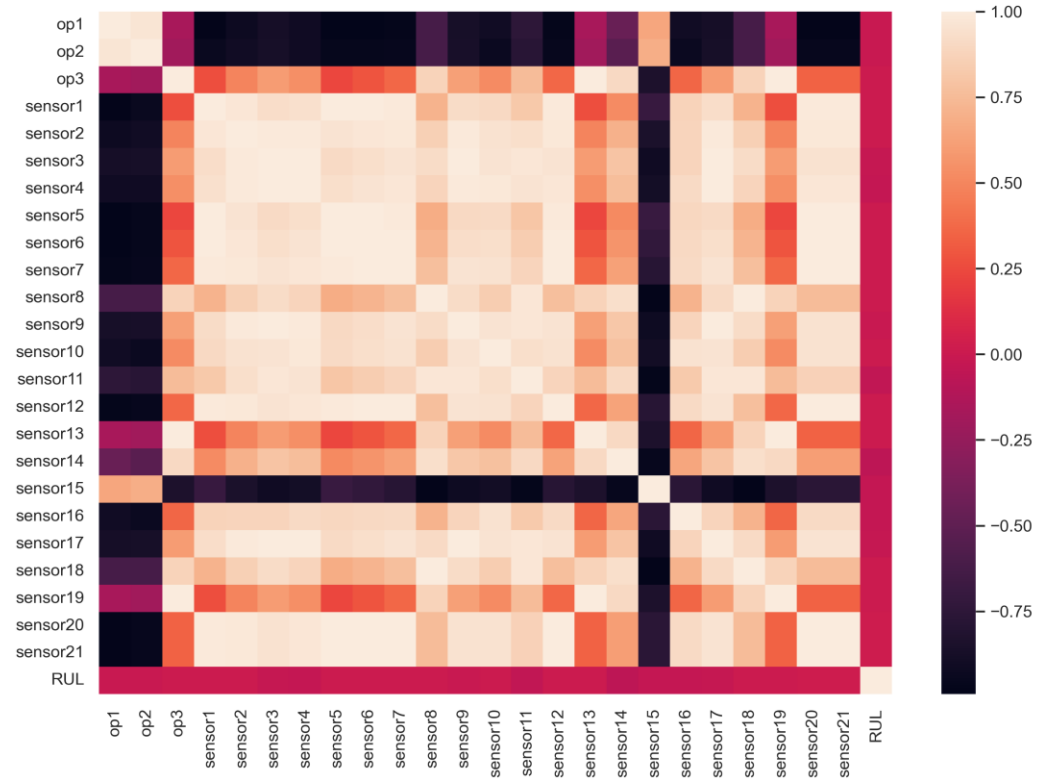
Engine_ID	Cycle	op1	op2	op3	sensor1	sensor2	sensor3	sensor4	sensor5	...	sensor15	sensor16	sensor17	sensor18	sensor19	sensor20	sensor21	RUL
1_1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	8.4195	0.03	392	2388	100.0	39.06	23.4190	191
1_1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	8.4318	0.03	392	2388	100.0	39.00	23.4236	190
1_1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	8.4178	0.03	390	2388	100.0	38.95	23.3442	189
1_1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	8.3682	0.03	392	2388	100.0	38.88	23.3739	188
1_1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	8.4294	0.03	393	2388	100.0	38.90	23.4044	187
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
249_4	251	9.9998	0.2500	100.0	489.05	605.33	1516.36	1315.28	10.52	...	8.4541	0.03	372	2319	100.0	29.11	17.5234	4
249_4	252	0.0028	0.0015	100.0	518.67	643.42	1598.92	1426.77	14.62	...	8.2221	0.03	396	2388	100.0	39.38	23.7151	3
249_4	253	0.0029	0.0000	100.0	518.67	643.68	1607.72	1430.56	14.62	...	8.2525	0.03	395	2388	100.0	39.78	23.8270	2
249_4	254	35.0046	0.8400	100.0	449.44	555.77	1381.29	1148.18	5.48	...	9.0515	0.02	337	2223	100.0	15.26	9.0774	1
249_4	255	42.0030	0.8400	100.0	445.00	549.85	1369.75	1147.45	3.91	...	9.1207	0.02	333	2212	100.0	10.66	6.4341	0

Snippet of Main Data frame

Dataset – Preliminary Analysis

- Analysed the mean,sd,min,max values for each column
- Analysed the correlation between each columns with one another
- Analysed how the starting RUL is varying between each engines





Problem 1

Given the sensor values and the operation parameters of an engine at a point in time,

Can we predict the Remaining useful life (RUL) ?

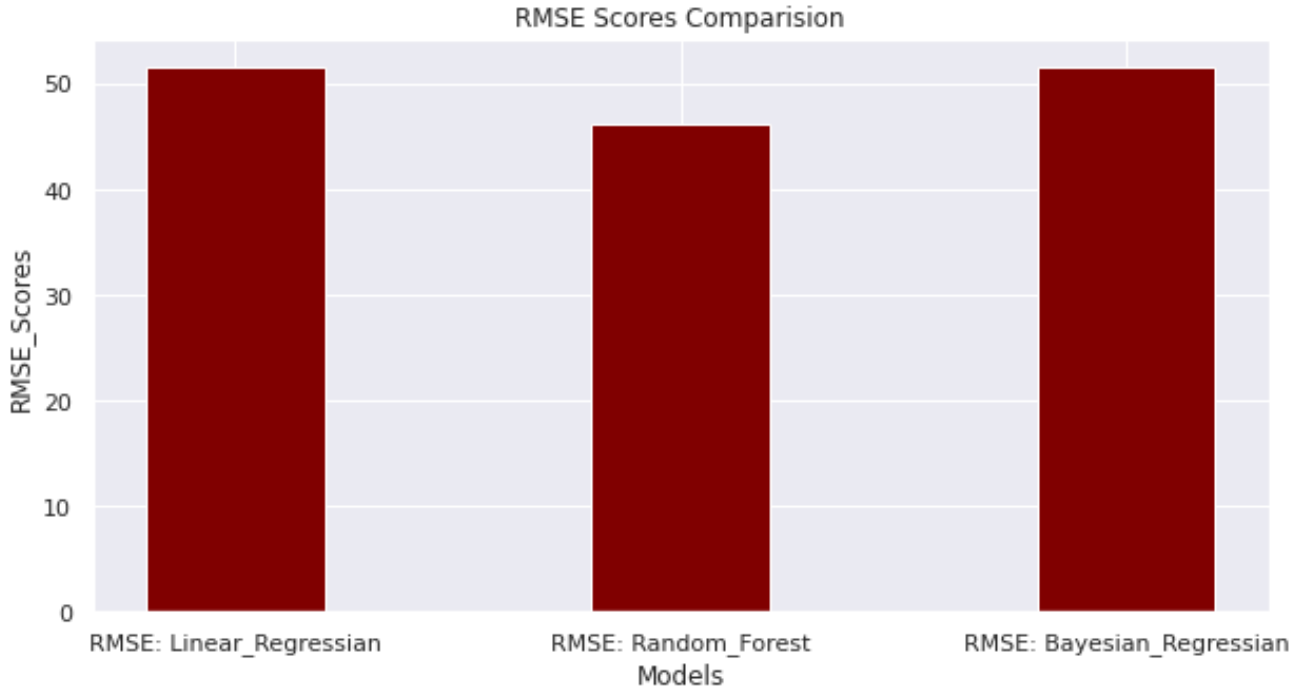
Proposed Solution

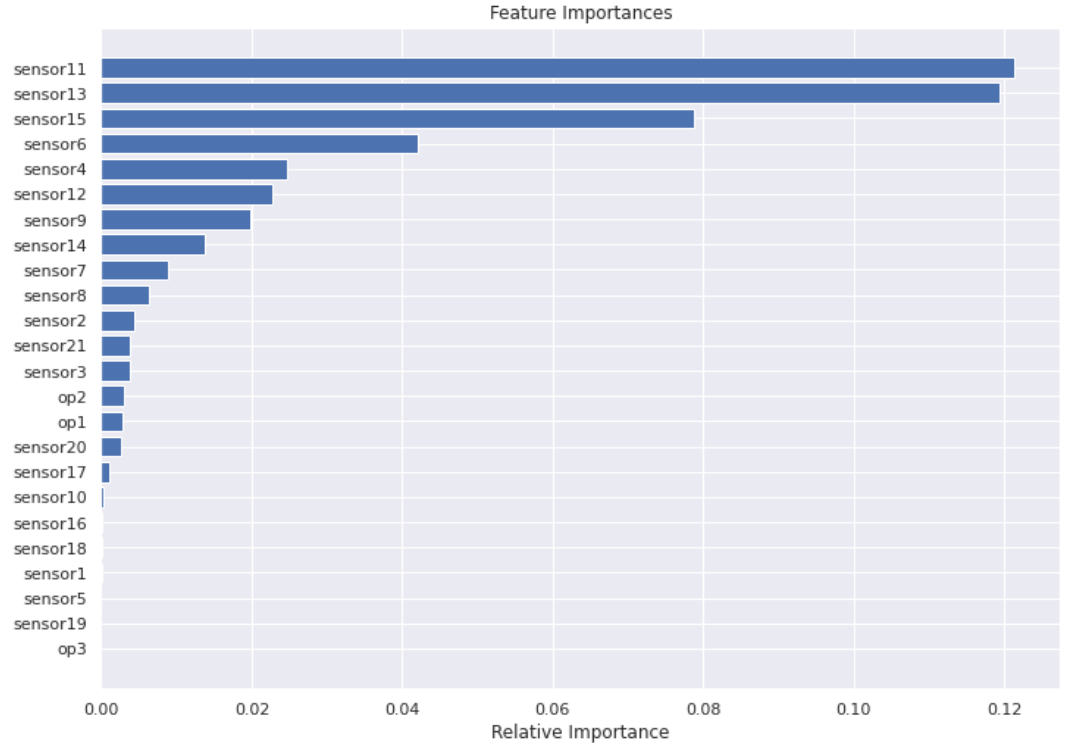
Regression Problem 1 – Overall Approach

- Cleaning and structuring the features and doing train-test split
- Building regression models and optimising them
  - Linear regression
  - Random Forest based regression
  - Bayesian regression
- Comparing the models and evaluating the test results – RMSE score
- Visualising the results

Evaluation And Analysis

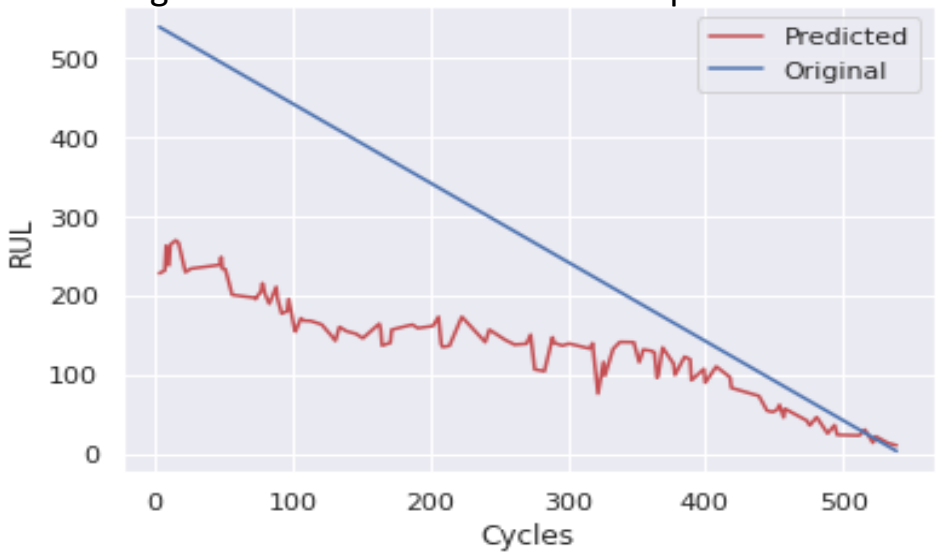
RMSE/Models	Linear Regression	Random forest	Bayesian Regression
Train RMSE	51.55	43.15	50.6
Test RMSE	51.66	46.12	51.32

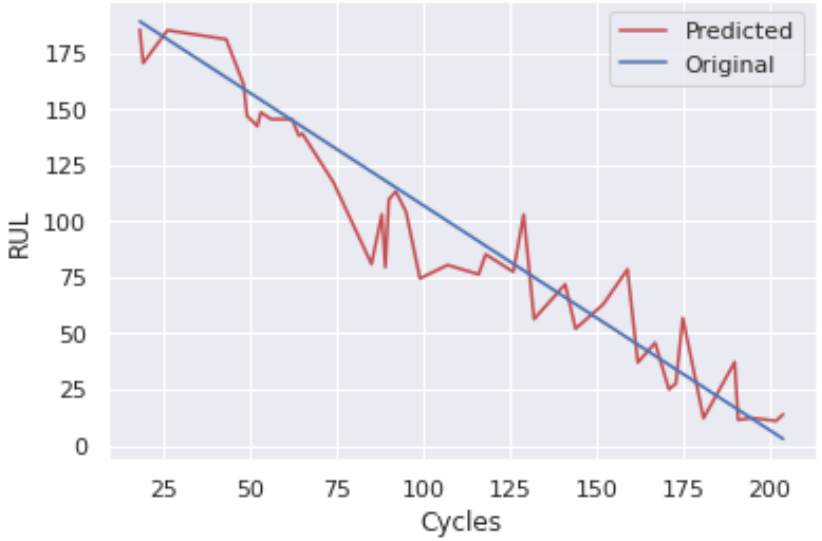




Visual Results

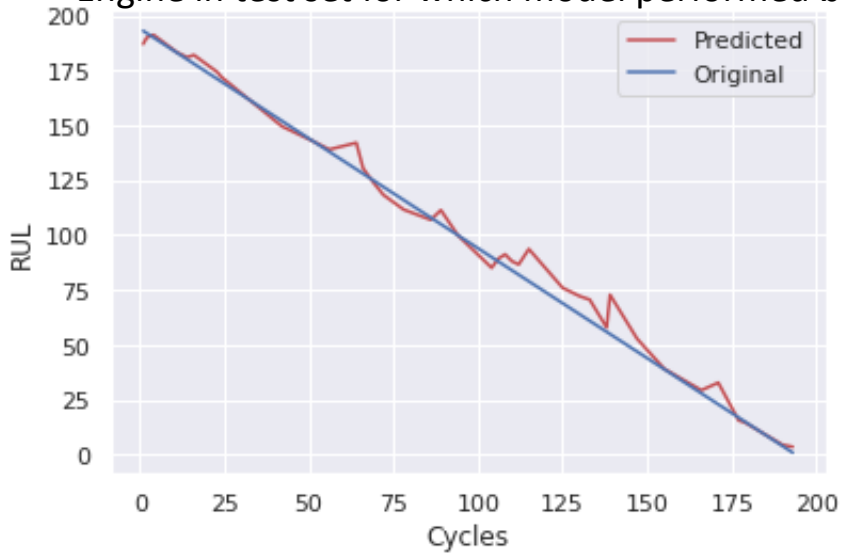
Engine in test set for which model performed **worst**

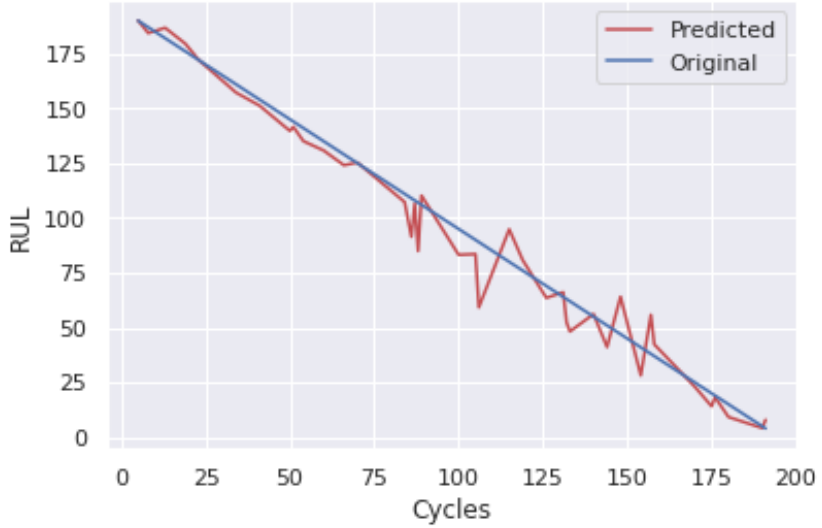




Random Engine from test set

Engine in test set for which model performed **best**





Random Engine from test set

Problem 2

Instead of predicting the RUL, Can we predict the percentage Health left?

Intuition behind the Idea:
Since each engine is different (could be different versions, different manufacturers for the parts) , if we could predict health – then that health would be more personalised to that engine
“Engine 1” – 10% health could 100 cycles | “Engine 2” – 10 % health could be 50 cycles
This would gives us **more control outside the model** . Something like introducing a calibration factor based on engine make or versions.

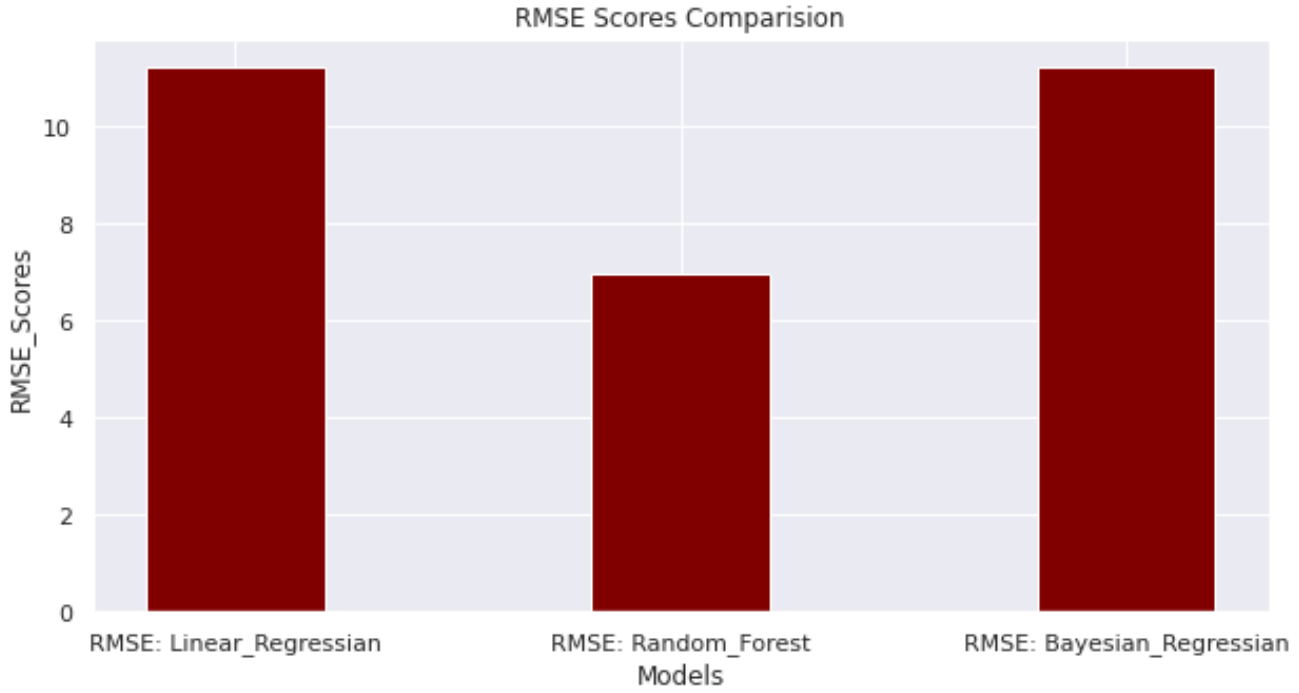
Proposed Solution

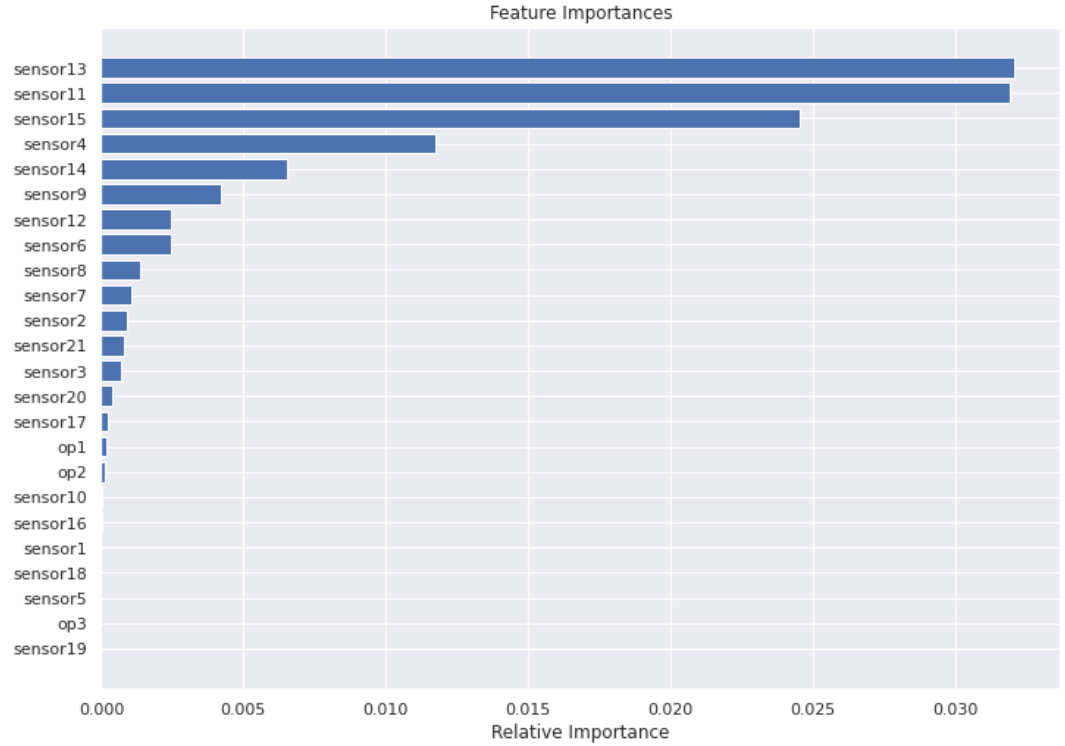
Regression Problem 2 – Overall Approach

- Creating a health% label based on RUL and total life of engine at each cycle
- Cleaning and structuring the features – doing train-test split
- Building regression models – optimising and evaluate them
  - Linear regression
  - Random Forest based regression
  - Bayesian regression

Evaluation And Analysis

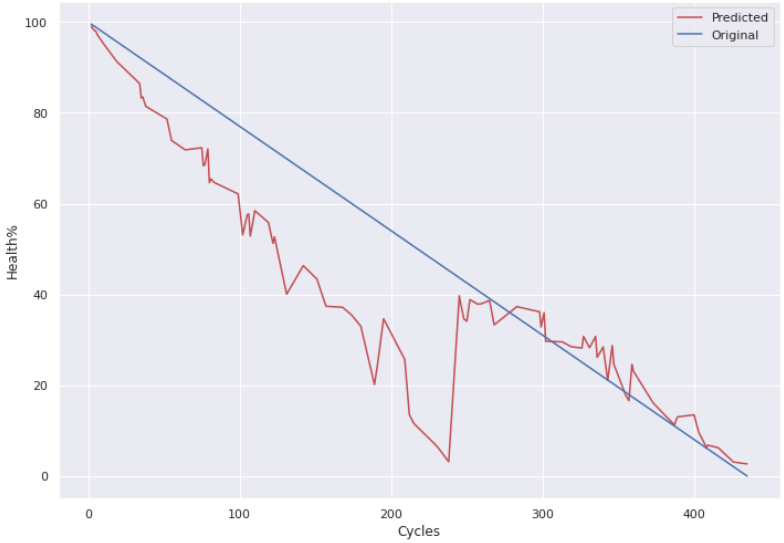
RMSE/Models	Linear Regression	Random forest	Bayesian Regression
Train RMSE	11.24	4.81	11.24
Test RMSE	11.25	6.98	11.25

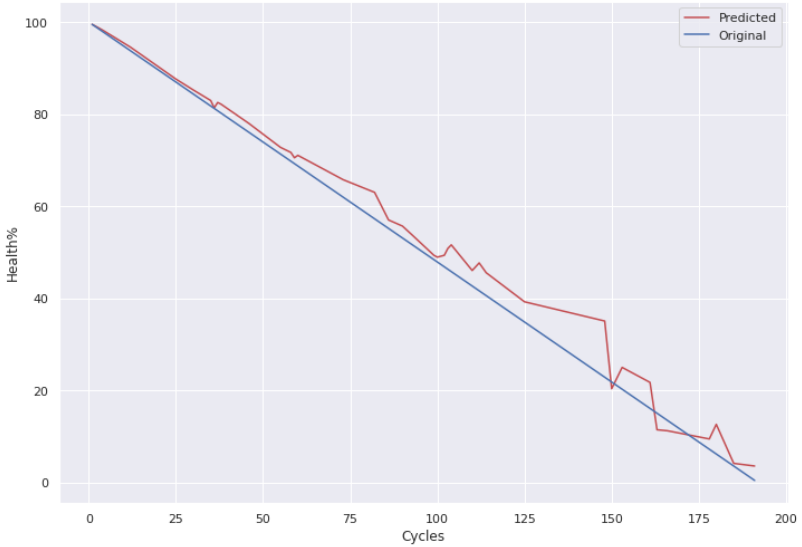




Visual Results

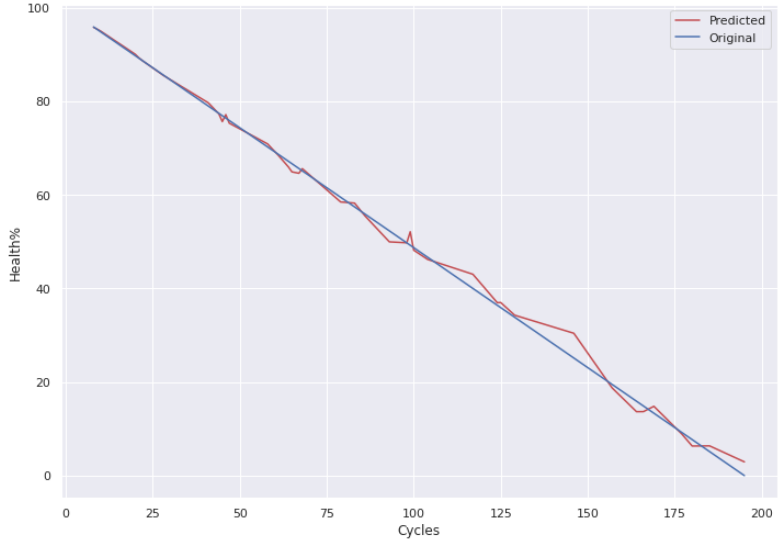
Engine in test set for which model performed **worst**

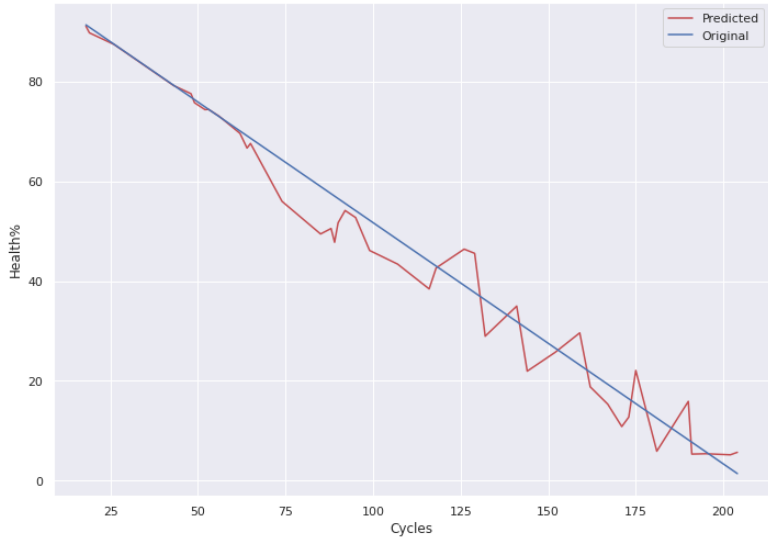




Random Engine from test set

Engine in test set for which model performed **best**





Random Engine from test set

Problem 3

Can we predict the Status of Engine at any point?

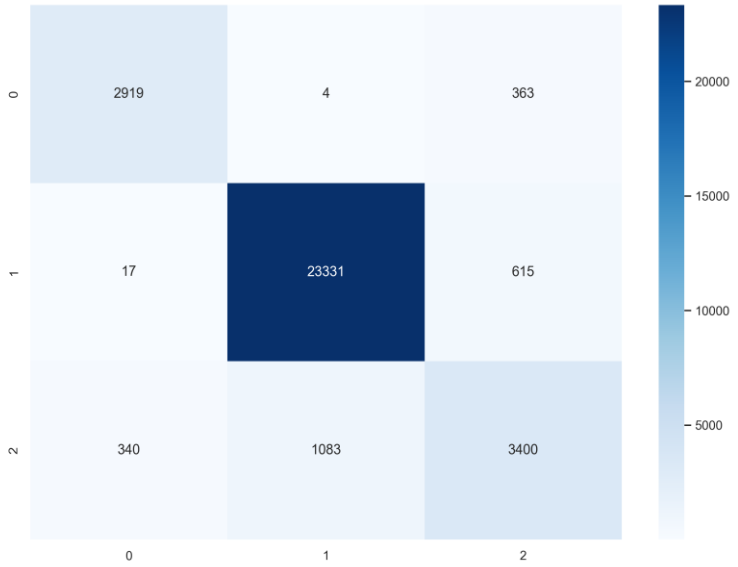
Provide an Alarm based system – Normal , Warning , Critical

Proposed Solution

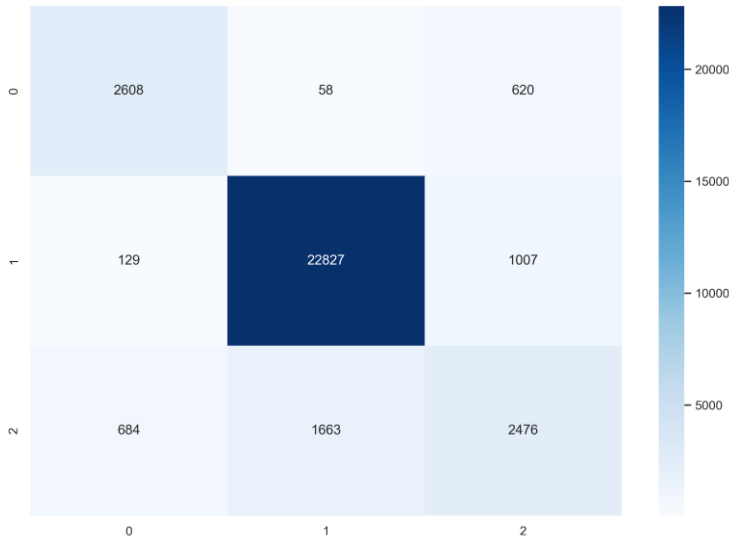
Classification Problem – Overall Approach

- Creating working, warning and critical labels using health%
- Cleaning and structuring the features – doing train-test split
- Building and optimising classification models
  - Random forest
  - KNN
  - SVM
- Evaluating and comparing models

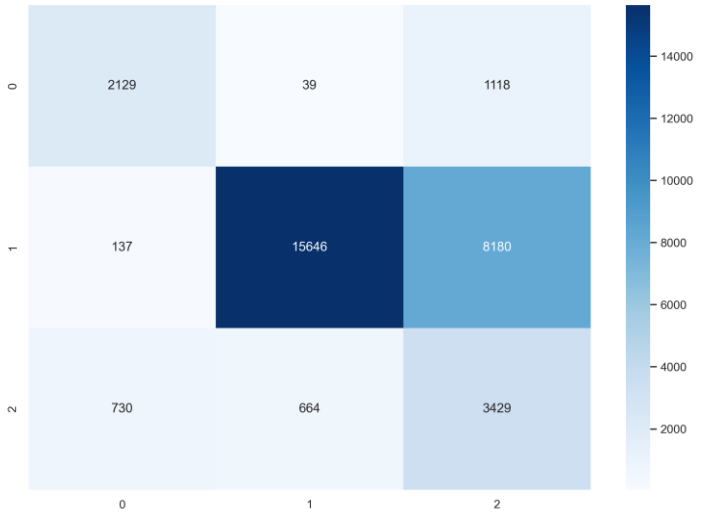
Evaluation



Random Forest Model - CF



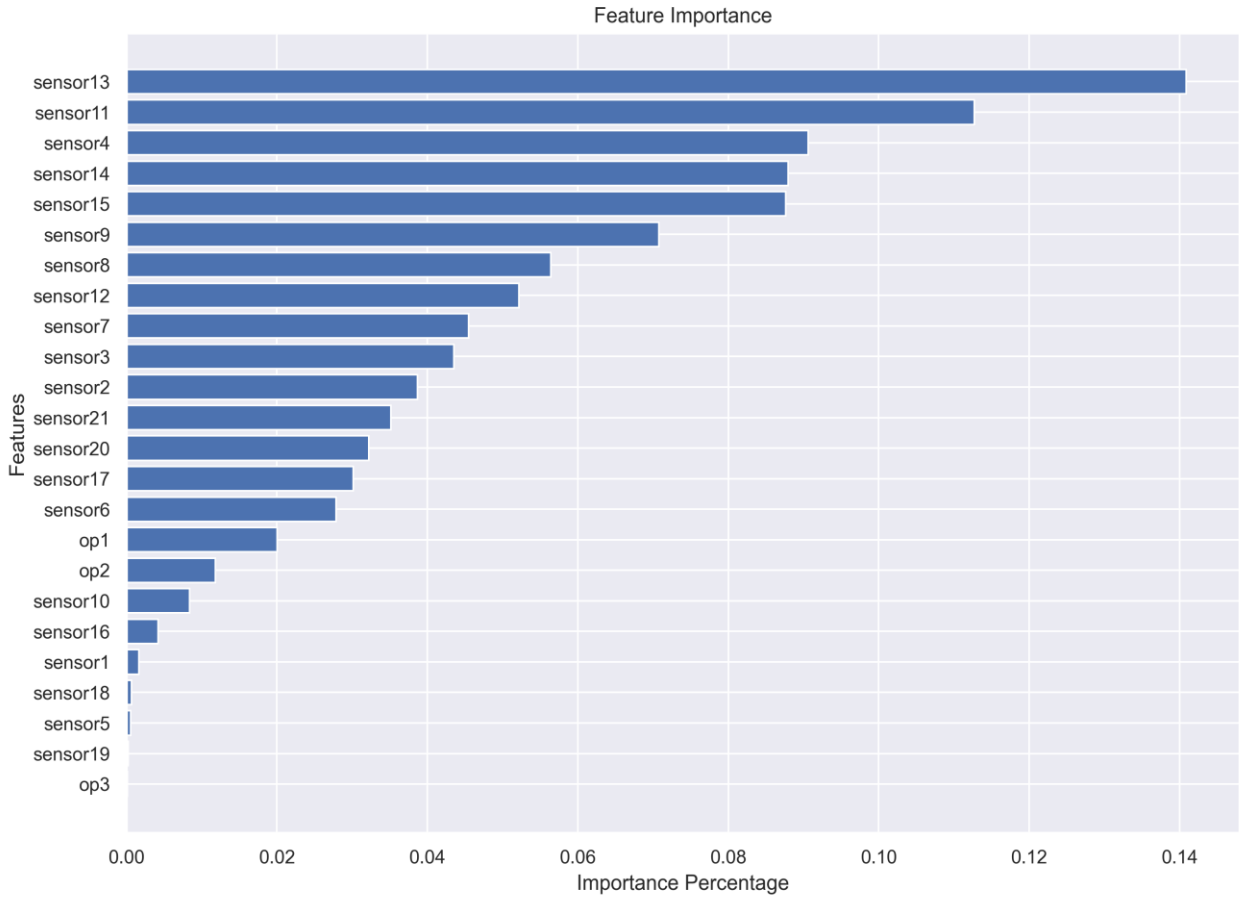
KNN Model - CF



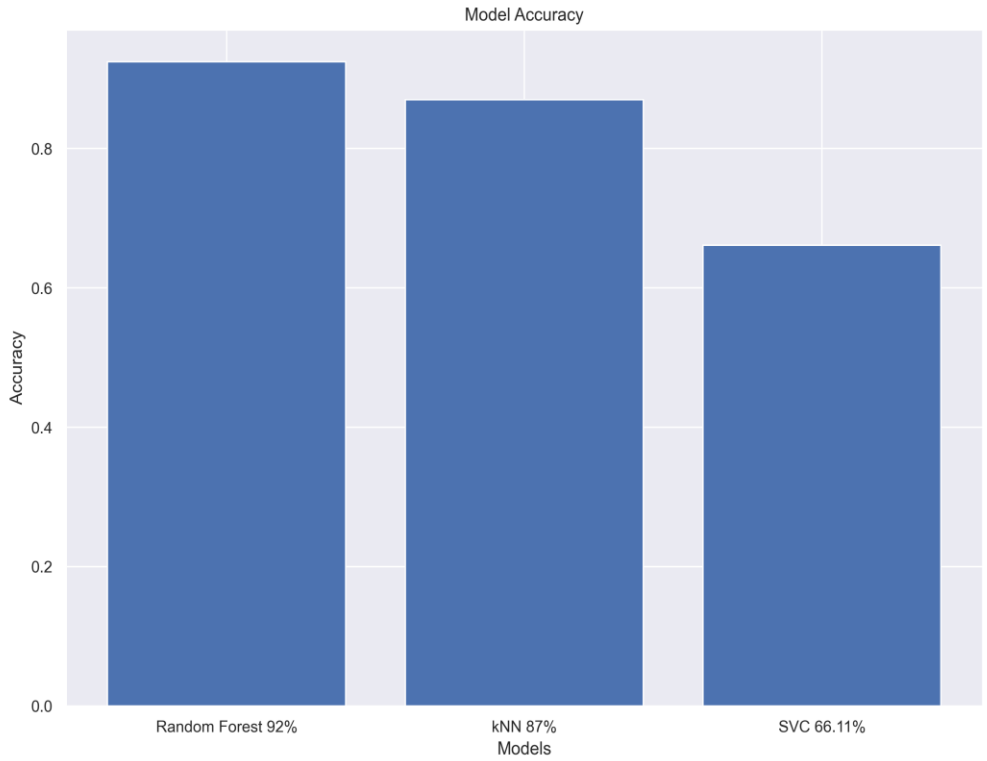
SVM Model - CF

- From the CFs we see that accuracy: RF =92.44% | KNN = 87.02% | SVM =66.11%
- 0 - Critical
- 1 - Normal
- 2 - Warning

Results



Feature Importance Analysis for the proposed RF model



Model Accuracies

Problem 4

We have so many sensor values ,operation modes etc. in our data,

Can we reduce the dimensions?

Can we visualize how the engines are distributed over the entire space?

Can we find anything more interesting ? That could help previous models?

Proposed Solution

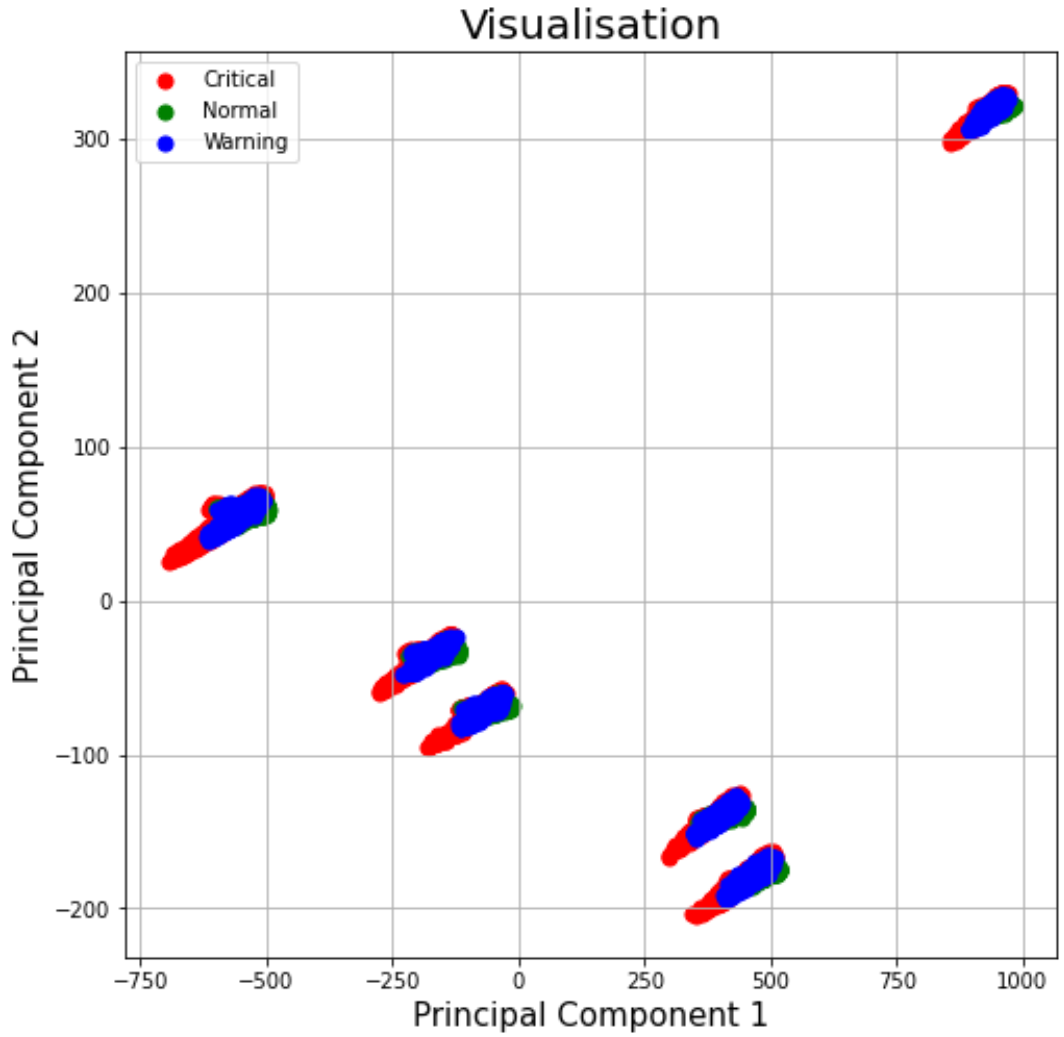
PCA – Overall Approach

- Applied PCA to the overall dataset
- Analysed how many components explain the variance at each %
- Visualised the results
- Generated useful information that could help our confidence in prediction

PCA Applied on the Dataset

We see that 3 principal components explain almost 99% of the variance in the data

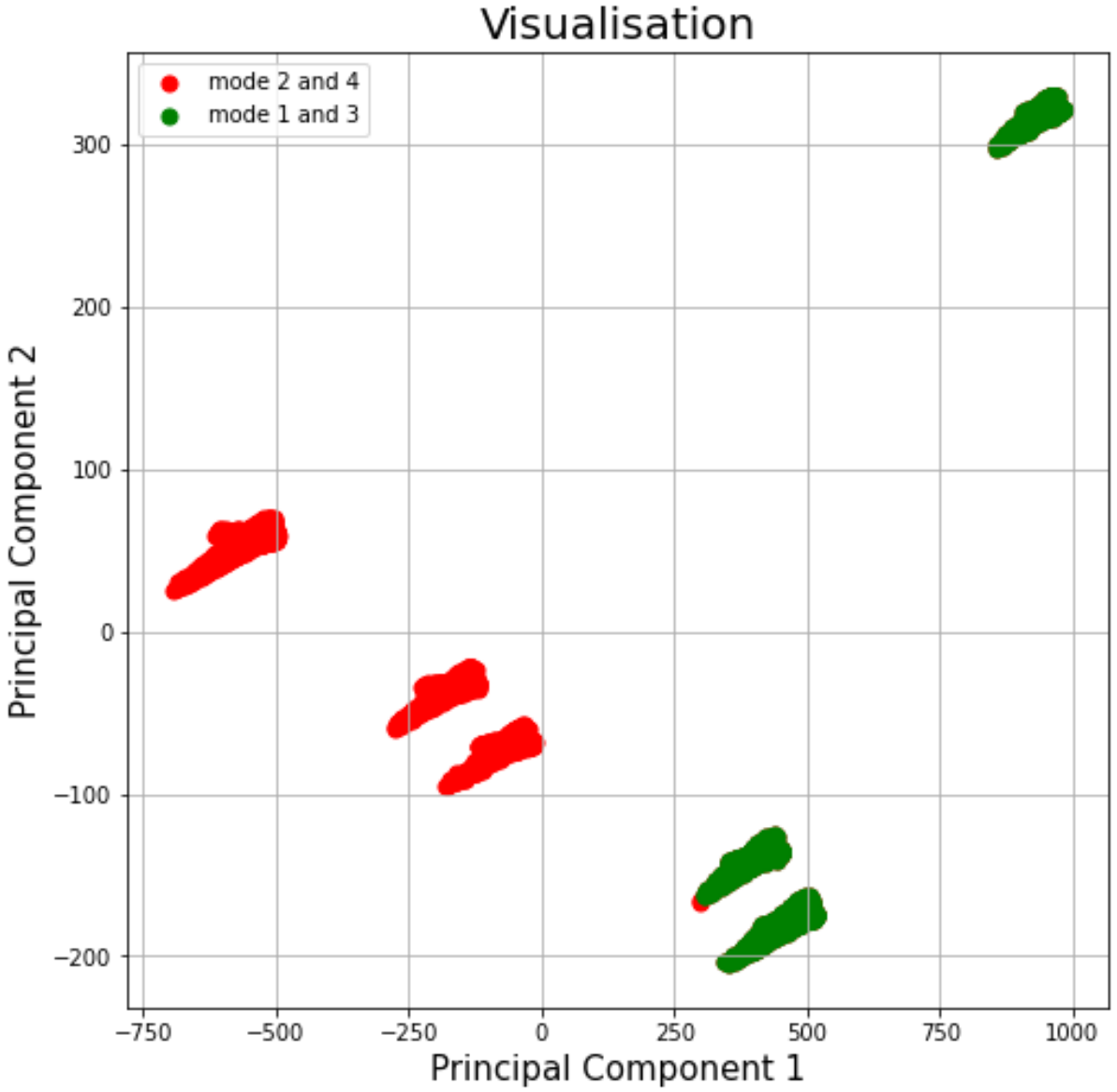
1st PC – 90.487% | 2nd PC – 6.83 % | 3rd PC – 2.342 %



Observations:

- The Data can be reduced from **25 features to 3 features** – to explain about 99% variance
- From the plots even though we see multiple clusters, in each cluster we see boundaries between normal , warning and critical
- **We see 6 clusters . What are they ?**

Understanding the PCA plot



Analysis:

- From the clusters we filtered out each cluster based on their PC 1 and PC2 ranges
- For each cluster we identified which engine they belong to
- Surprisingly we see that there are clusters made up of engines which have been run in different modes – like different environment settings

- This analysis helps us in a way that , if we could first find the location our test engine (its cluster). Then we can use that information for **our regression and classification models** for calibration or confidence of prediction estimation.

Conclusion

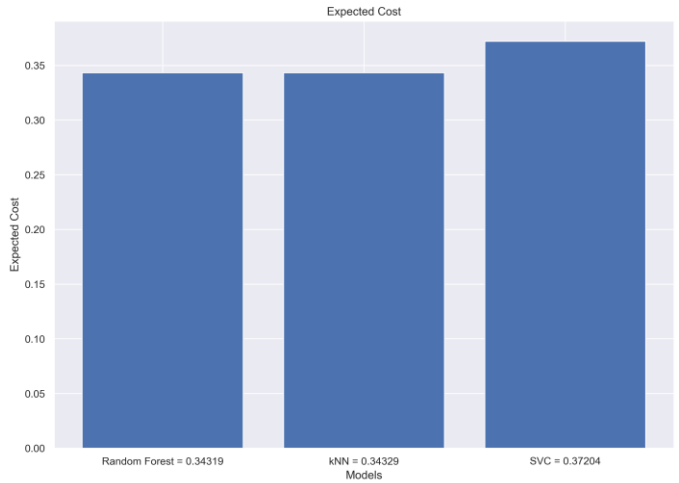
Business Perspective:

- From business perspective, expected cost (or expected benefit) is a popular metric of evaluating these types of models – reference [5]
- In this study, we assign 0.5 benefit of true negatives, 0.99 benefit of true positives, 0.4 cost of false positives, and 0.99 cost of false negatives – These weights can be adjusted based on the business needs

$$Expected\ Cost = P(\mathbf{p}) \times [P(TP) \times benefit(TP) + P(FN) \times cost(FN)] + P(\mathbf{n}) \times [P(TN) \times benefit(TN) + P(FN) \times cost(FP)]$$

From all the models built, we see that the **Random forest classifier gives the lowest cost as well as the highest accuracy**.

Overall both the regression as well the classification solutions proposed, **should help in performing better with respect to maintenance tasks.**



References

1. NASA, "Prognostics Center of Excellence Data Repository",http://ti.arc.nasa.gov/projects/data_prognostics
2. https://www.kaggle.com/datasets/behrad3d/nasa-cmaps
3. Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation. https://ti.arc.nasa.gov/m/pub-archive/154/154_Saxena.pdf.
4. https://www.wipotec-ocs.com/us/dynamic-weighing-systems-cep/predictive-maintenance - Fig source in slide 2
5. F. Provost and T. Fawcett (2013). Data Science for Business (Chapter 7). O'Reilly Media, Inc.

Thank You !