Navigation

Go to

Home

Navigation

Go to

Home

Navigation

Go to

Home

Estimate Remaining Useful Life of Turbofan engines

Photographer: Luke Sharrett/Bloomberg

One of the main challenges in the aviation industry is to reduce maintenance costs an downtime of machines by maintaining or improving safety standard. A large per cent costly delays are a result of unplanned maintenance such as when an aircraft has an behaviour on the previous flight, creating an operational disruption and even requirin change.

Since you don't know when failure will occur, you have to be conservative in your plan especially if you're operating safety-critical equipment like engines. But by schedulin maintenance very early, you're wasting machine life or spare part life that is still usab adds to costs to the owner.

However, if you can predict when machine failure will occur, you can schedule mainte before it. By implementing predictive maintenance we can minimize unnecessarily sc maintenance and production hours lost during maintenance (improve the overall avail equipment) and reduce the cost of spare parts and its consumables during the maint process.

In order to develop an algorithm that predicts the breakdown of a piece of equipment given time window (typically some number of days), we require enough historical dat us to capture information about events leading to failure.

About Data set:

In this post, we are using the C-MAPSS dataset, which is engine degradation simulati carried out at NASA by using C-MAPSS simulation software. C-MAPSS has created fou data sets simulated under different combinations of operational conditions and fault

Data sets are consist of three operational settings and 21 sensor measurements (tem pressure, fan speed, etc.) for several engines and for every cycle of their lifetime.

The engine is operating normally at the start of each time series and develops a fault a during the series. ***In the training set, the fault grows in magnitude until system failur set, the time series ends sometime prior to system failure.***

Problem statement:

The goal of predictive maintenance is to predict at the time t, using the data up to tha whether the equipment will fail in the near future.

This problem is can be formulated in two ways:

Classification: we are aims to predict the probability that the equipment will fail with specified time window.

Regression: A regression-based approach to which aims to estimate the remaining ti end of the equipment's useful life of an engine.

In this blog, we'll focus on the second dataset (FD002) in which all engines develop s with six operating conditions.

Navigation

Go to

Data Exploration

Data Exploration

☐ Show Train dataset

☐ Show Test dataset

Before we start EDA, we need to compute a target variable for this data set which is R

(RUL)Remaining Useful Life(the length of time a machine is likely to operate before it repair or replacement.

As mention about the definition,

```
RUL = a total number of the life cycle engine - Current engine life cycle
```

To calculate RUL in our training data set, we just did group by engine number(max nu cycles) and present engine life cycle.

For test data, the time series ends sometime prior to system failure, after that we do and we have ground-truth data (The true remaining cycles for each engine in the test

Example: let consider the engine one life cycle data. In a training data life cycle of eng from 1 and end 223. In test data, the engine runs some cycle like 120, we need to fin many cycles still remain.

To calculate RUL in our test data set, we did the same thing as training data by adding truth data value to test data.

☐ Show code

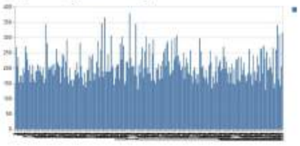
Data with RUL calculation

	Engine_No	Time_in_cycles	Altitude	Mach_number	TRA	[illegible]
0	1	1	10,447.4810	[illegible]	100	245.420
1	1	2	12,001.0014	[illegible]	100	247.222
2	1	3	7,419.6342	0.6218	60	255.966
3	1	4	12,003.9470	0.8416	100	247.222
4	1	5	7,020.1526	0.6201	60	255.966

Since the degradation in a system will generally remain negligible until after some pe operation time, the early and higher RUL values are probably unreasonable. We can't anything about the RUL before that point because we have no information about the i and how.

☐ Show Sketch of the RUL of Engines

Exploratory Data Analysis:

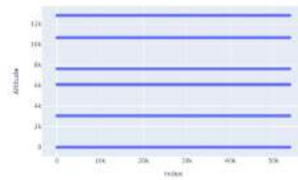


- Engines have different life durations.The average working time in train data is 2 with a minimum of 128 and a maximum of 378.
- 96% of engines are within 300 life cycle and 4% only engines lived above 300 l this engines we consider as an extreme case.

Operational Setting:

What column to you want to display

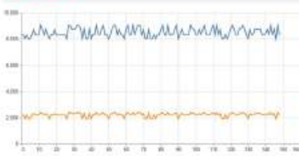
Altitude



As plots illustrate, all engines are operated six different altitude conditions with a ran 12000M.) and with 5 different operating speed(Mach number) of (0~0.84), with two t position (TRA) (60~100).

sensors data:

What column to you want to display



- sensors data remain same throughout engine's life.
- we understand that sensors data T2,T24,T30,T50,P2,P15,P30 highly correlated another (Based on Temperature and pressure graph look like sames).
- sensors Nf and Nc also correlated and also W31 and W32
- As we can see, all the sensors data is higly correlated. even data is highly corre keeping this features.

Let's plot PDF individual sensors for any given engines.

which engine you want comparison

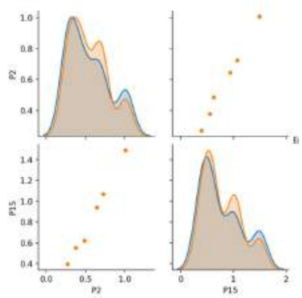
1 ✕ 5 ✕

Plot PDF of below sensors data

◉ Pressure sensors
○ Temperature sensors
○ Speed sensors
○ Other parameters

Select from Pressure sensors

P2 ✕ P15 ✕



Engine number is: 1 , total number of engine cycle(RUL) is: 149

Engine number is: 5 , total number of engine cycle(RUL) is: 154

when we are comparing this engines sensors data,we can't tell differentiate between by looking into PDF plots for this dataset

Navigation

Go to

Data Exploration

Navigation

Go to

Data Exploration

Navigation

Go to

Data Exploration

Navigation

Go to

Data Exploration

Navigation

Go to

Data Exploration

Navigation

Go to

Data Exploration

Navigation

Go to

Data Exploration

Navigation

Go to

Feature Engineering

Navigation

Go to

Feature Engineering

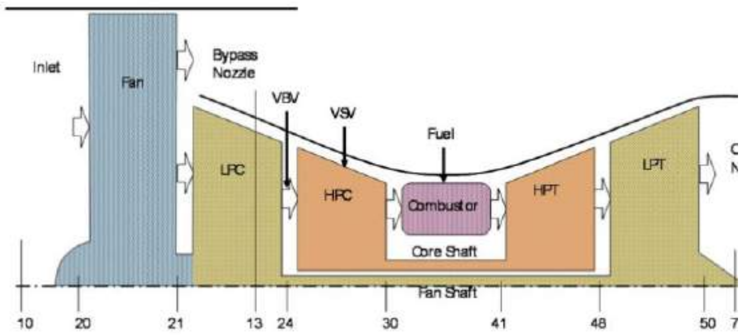
Navigation

Go to

Feature Engineering

Feature engineering

Based on my domain knowledge i come up with 40 new features which can improve n performance.This features are performance parameters of turbofan engines like, Thrust,Propulsion efficiency,Thermal efficiency,Turbine Entry Temperature(TET),etc...



From C-MAPSS paper

Schematic diagram showing the components,station numbers and sensor data taken C-MAPSS engine.A brief description some of the major modules is summarized below

Exhaust Gas Temperature (EGT)(T90): expressed in Kelvin, is the temperature at the exhaust and a measure of an engine's efficiency in producing its design level thrust; tl the EGT the more wear and deterioration affect an engine. High EGT can be an indicat degraded engine performance. An exceedance in EGT limits can lead to immediate da engine parts and/or a life reduction of engine parts. With this in mind it then becomes important to keep the EGT as low as possible for as long as possible.

Turbine Inlet Temperature(T41): The general trend is that raising the turbine inlet te increases the specific thrust of the engines with a small increase in fuel consumption rate.However, high temperatures can damage the turbine, as the blades are under lar centrifugal stresses and materials are weaker at high temperature.

Thermal efficiency is a prime factor in turbine performance.The three most importan affecting the thermal efficiency are turbine inlet temperature(T41), compression ratio and the component efficiencies of the compressor and turbine. Other factors that affe efficiency are compressor inlet temperature(P21) and combustion efficiency.Maximu efficiency can be obtained by maintaining the highest possible exhaust temperature.s life is greatly reduced at high turbine inlet temperatures, the operator should not exce exhaust temperatures specified for continuous operation.

All this feature are created using given sensor data with some assumptions. I am not discuss all the feature hear. Please ref to [1],[2]

More about my Feature engineering GitHub

Improvement:

- From EDA we understand that data set having different operating conditions,so create clustering and do labeling on top that create onehotencoder but this onl FD002 and FD004 datasets
- If possible we can create feature engineering with autoencoder.
- If we can remove noise from data, we can achieve good results.Let's try noise r effectively by using the autoencoder or other noise reduction techniques

Navigation

Go to

Model Building

select C parameter to controls the penalty strength

1e-07 1

Select C values is `1e-07`

Navigation

Go to

Model Building

select C parameter to controls the penalty strength

1e-07 1

Select C values is `1e-07`

Select regularization

none

Navigation

Go to

Model Building

select C parameter to controls the penalty strength

1e-07 1

Select C values is `1e-07`

Select regularization

none

Navigation

Go to

Model Building

select C parameter to controls the penalty strength

1e-07 1

Select C values is `1e-07`

Select regularization

none

Navigation

Go to

Model Building

select C parameter to controls the penalty strength

1e-07 1

Select C values is `1e-07`

Select regularization

none

Navigation

Go to

Model Building

select C parameter to controls the penalty strength

1e-07 1

Select C values is `1e-07`

Select regularization

none

Select regularizer is none

Model Building

As mention early in home page we are going to solve both regression and classificatio

After completing feature engineering, we got total 63 feature and applied random for algorithm to select feature and we got 23 feature.

We have 4 different data set with different operating condation.Please Select data set

Please choose a Data set Type

FD001

Scaling:

Sklearns' MinMaxScaler can be used to create a scaler fitted to our train data. The def settings create a scaler to scale our training features between 0–1. The scaler is then both our x_train and x_test set.

Shape of train data : `(20631, 26)`

Shape of test data : `(13096, 26)`

Data processing:

By default we are having Train and Test datasets.we keep Train data for train and split into cross_validation and test data, we pick the last sequence data points as test data data points as cross_validation.

we are taken RUL and Label columns as dependent variable, remaining 23 columns ar independent variable and remaining one columns is Engine_No column.

For classification approach, generating the label by ourself in this way, from 0 to 30 cy labeled as 1(fail) and rest (>30) as 0(Not fail). For regression problem we keep RUL co without any changes.

	Engine_No	T50	Nf	Ps30	phi	NRf	Nfc	B
0	1	0.3898	0.2424	0.3690	0.6333	0.2059	0.1996	0.36
1	1	0.3526	0.2121	0.3810	0.7665	0.2794	0.1628	0.41
2	1	0.3785	0.2727	0.2500	0.7983	0.2186	0.1718	0.35
3	1	0.3312	0.3182	0.1667	0.8891	0.2901	0.1749	0.16
4	1	0.4046	0.2424	0.2560	0.7463	0.2353	0.1747	0.08

After split data.we process data for machine learning and deep leaing algorithm little which is followed below.

Choose the algorithm

Please choose a Model Type

Machine learning

Please choose a Problem Type

Classification

For machine learning algorithm we send data to algorithm as a individual data point.

Shape of Train data after split: `(20631, 23)`

Shape of Train labeling data : `(20631,)`

Shape of Cross validation data after split: `(12996, 23)`

Shape of Cross validation labeling data : `(12996,)`

Shape of Test data after split: `(100, 23)`

Shape of Test labeling data : `(100,)`

☐ Show code for ML data processing

Please choose a Model from list

Logistic regression

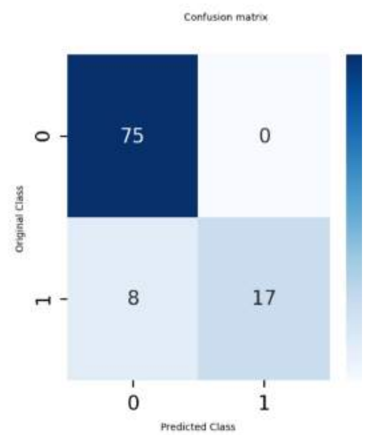
Let's train Logistic regression on FD001 data set

The train log loss is: `0.09552238858812181`

The cross validation log loss is: `0.03642935829761147 6`

The test log loss is: `0.2258768212223636`

---------------------- Confusion matrix ----------------------



Number of mis-classified points : `8.0`

precision: `0.9638144578313253`

recall: `1.0`

fscore: `0.9493670886075949`

Negative class labels: `75` positive class labels: `25`

Conclusion

The task of predicting RUL is certainly interesting.But there is still quite a bit of room f improvement with better training, models, and feature engineering.

Our model give good results,but that's not good enough for aviation standards we nee improve model performance as follows:

- Deep learning models are not performing as we expected maybe deep learning performs well if have large data sets.
- As we know that data generated by sensors, it means that the data set is conta with sensor noise. we can improve model performances by applying denoising t

You can find the full code on my github page [here]

Made with Streamlit

Model prediction

Please choose a Data set

Model prediction

Please choose a Data set

FD001

Which engine you want predict

8

Select any engine from `1` to `100`

At give some point of time engine fail or not

1 165

you are completed `159` and you are predicting what happened in next 30 cycle

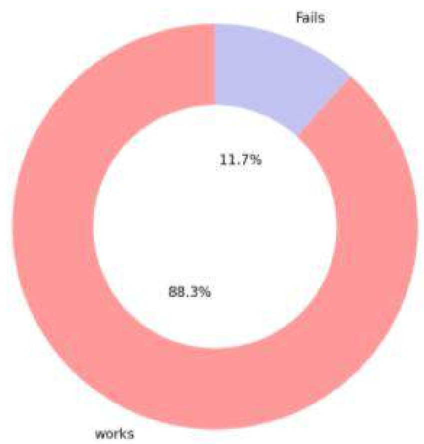
Model Prediction

Let's predict how models are performing by giveing unseen data.

Selected raw data file from

https://raw.githubusercontent.com/gethgle/predictive-maintenance/main/data/test_

Probability of Engine 8 failure with next 30 life cycle:



Remaining useful life of Engine 8:

Predicted cycles remaining: 89

Made with Streamlit