Research Stay: Predictive Maintenance of Industrial Equipment

Juan Echeagaray

School of Engineering and Sciences
Instituto Tecnológico y de Estudios Superiores de Monterrey

October 10, 2023

Agenda

Maintenance scheduled by data analytics on historical data

- Enhance operational efficiency
- Sustainability and cost reduction
- Competitive advantage
- **Safety at the forefront**

Figure: Turbine failure mid-flight [1]

Predictive maintenance faces 2 main challenges:

Implementation

Need a model to estimate the RUL of a set of machinery, in an efficient and reliable manner

Interpretability

Need to *understand* why a model produces a given RUL estimate, becomes more important in critical environments

Objectives

This research project aims to develop:

1. PdM framework to predict RUL for a designated fleet of machinery
2. Model which ensures reproducibility, stability, robustness and confidence
3. Tools to interpret and visualize the model's predictions

Figure: Sample of a uniform fleet of cars

Objectives
Scope

The previous objectives are to be accomplished subject to the following constraints and assumptions:

- RUL prediction of an uniform fleet of machines
- Availability of a labeled dataset with run to failure sequences of each machine

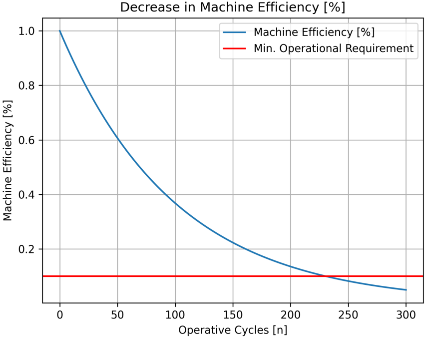


Figure: Efficiency loss as a machine operates

Objectives
Loss Function

The 2021 PHMAP challenge proposed the loss function (1); the average of the RMSE and NASA's scoring function [2].

$$
\begin{aligned}
\mathcal{L}(y, \hat{y}) &= \frac{1}{2}\left(\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} + \frac{1}{m}\sum_{i=1}^{m}\exp(\alpha \cdot (y_i - \hat{y}_i)) - 1\right) \\
\alpha &= \begin{cases} \frac{-1}{10} & \text{if } y_i - \hat{y}_i \leq 0 \\ \frac{1}{13} & \text{if } y_i - \hat{y}_i > 0 \end{cases}
\end{aligned} \tag{1}
$$

As a remark, (1) is an asymmetric loss function with a higher penalty for overestimates.

Flight conditions and readings from a fleet of turbofan engines, derived from NASA's CMAPSS model, including real flight conditions and relates the degradation process to the operating history of the machine. [3]

- Used in the PHMAP 2021 Data Challenge [4]
- State of the art prognosis dataset (akin to MNIST and CIFAR for CV)

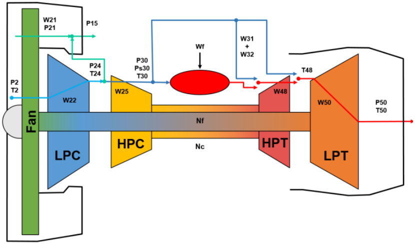


Figure: CMAPSS turbofan engine schematic

Exploratory Data Analysis

Data Overview

- Split into 10 h5 files
- Sampling frequency of 1 second
- Contains sensor readings, environmental descriptors, auxiliary variables, virtual sensor readings and RUL values
- Corrupted file in the dataset

Symbol	Description	Units
alt	Altitude	ft
Mach	Flight Mach number	-
TRA	Throttle-resolver angle	%
T2	Total temperature at fan inlet	°R
Wf	Fuel flow	pps
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T48	Total temperature at HPT outlet	°R
T50	Total temperature at LPT outlet	°R
P15	Total pressure in bypass-duct	psia
P2	Total pressure at fan inlet	psia
P21	Total pressure at fan outlet	psia
P24	Total pressure at LPC outlet	psia
Ps30	Static pressure at HPC outlet	psia
P40	Total pressure at burner outlet	psia
P50	Total pressure at LPT outlet	psia
RUL	Remaining Useful Life	cycles
unit	Unit number	-
cycle	Flight cycle number	-
Fc	Flight class	-
hs	Health state	-

Table: General description of dataset variables [4]

Exploratory Data Analysis

Flight Class Distribution

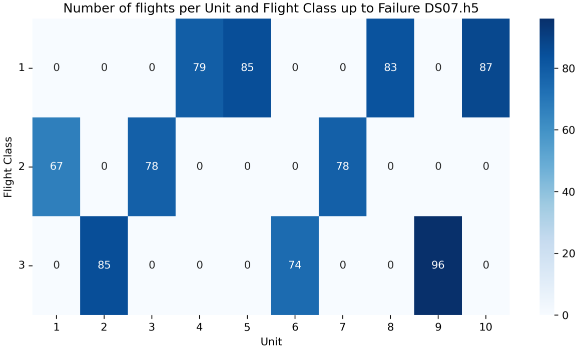


Figure: Flights recorded for each class per unit

Flight classes encode flight duration[1], and there are no inter-class units

[1] FC1 ranges between 1 and 3 hours, FC2 between 3 and 5, and FC3 is greater than 5

Exploratory Data Analysis
Sample Operating Conditions

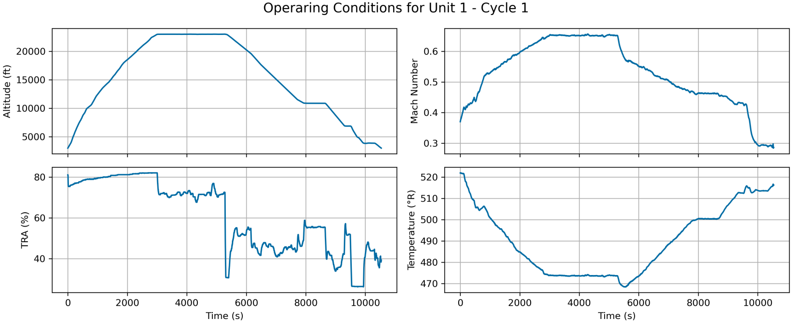


Figure: Operating conditions for the first flight of Unit 1

Exploratory Data Analysis

Operating Conditions per Flight Class

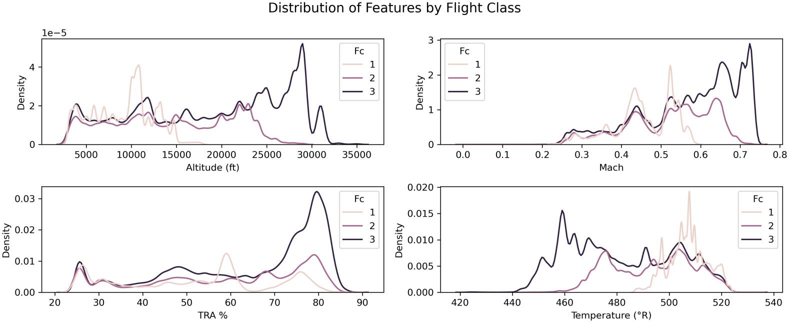


Figure: Distribution of operating conditions per Flight Class

Distinctive operating conditions per FC where FC2 can be seen as a mixture of FC1 and FC3

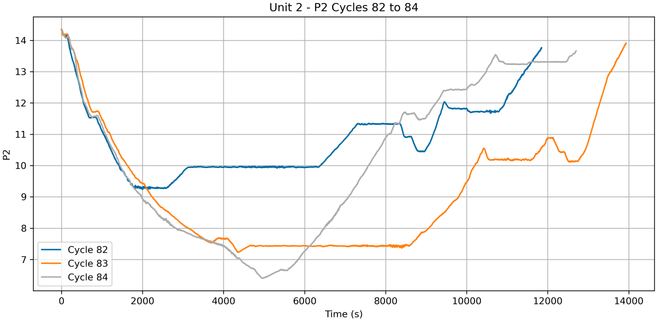


Figure: Total Pressure at fan inlet (P2) for the last 3 cycles of unit 2

Smoothness degradation with a presence of new plateaus near the end of the flight

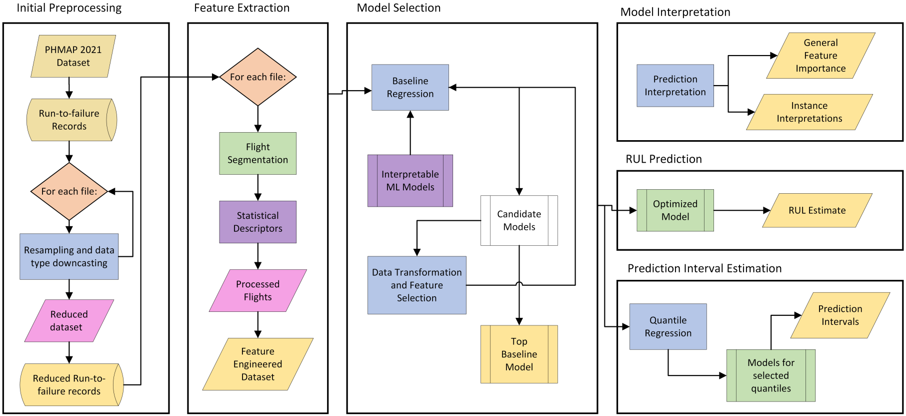


Figure: Proposed Methodology

Data Resampling and Downcasting

Sensor readings every second can lead to large arrays, think about sampling frequency (1 Hz, 1 KHz, 1 MHz, etc. . .) and data types.

- Resampling to catch the general shape
- Downcasting to a datatype with lower memory footprint (64bit float to 32bit float)

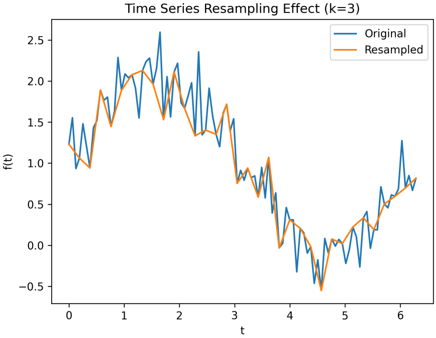


Figure: Time Series Resampling and Downcasting Effect

Problem

Dataset consists of a set of multivariate time series of varying length with a distinct number of events per series

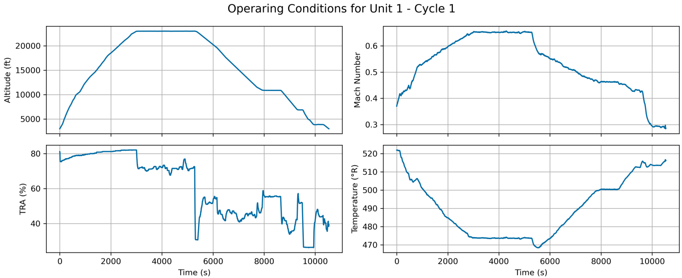


Figure: Operating conditions of Plane 1 during its first flight

Time Series Segmentation

Basics

For a given signal $Y = \{\boldsymbol{y}_t\}_{t=1}^{t=T}$ of T samples, where $y_t \in \mathbb{R}^d$, we assume there exists a set $\mathcal{T} = \{t_1^*, t_2^*, \ldots, t_n^*\}$ coding the $n-1$ stages of Y

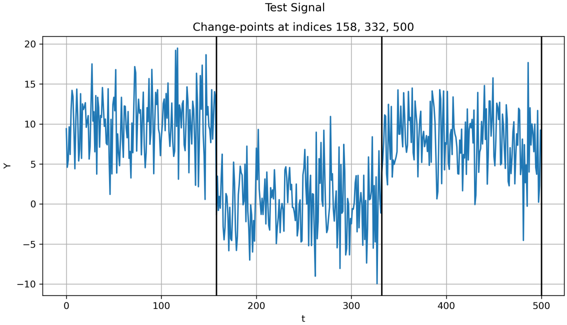


Figure: Piecewise constant signal with normal noise

1

[1]The last index eases the software implementation

Greedy sequential algorithm that estimates 1st change-point as:

$$\hat{t}_1 := \underset{1 \leq t < T-1}{\arg\min}\ C(y_{0\ldots t}) + C(y_{t\ldots T}) \tag{2}$$

It then repeats the operation on both left and right segments until the specified number of splits is achieved.

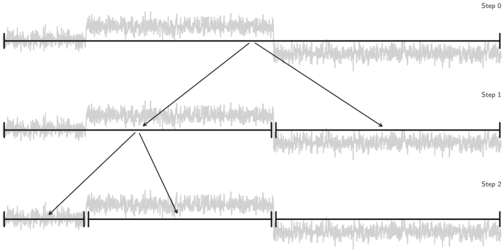


Figure: Overview of Binary Segmentation retrieved from [5]

An estimator of shifts in the central point of a distribution. Given a series $\{y_t\}_{t\in\mathcal{I}}$ where $y_t \in \mathbb{R}^d$:

$$C(y_{\mathcal{I}}) = \sum_{d} \sum_{t\in\mathcal{I}} ||y_t - \bar{y}||_2^2 \tag{3}$$

where $\bar{y}$ is the component wise mean of $y_{\mathcal{I}}$

- Needs scaling, considering not to penalize on segment length

Feature Extraction

Statistical Descriptors

Given a segment S_i which encompasses T_i timestamps, $S_i \in \mathcal{M}_d^{T_i}(\mathbb{R})$ with $T_i \geq \alpha$ where α is a constraint on the minimum size of S_i. We then use a set of descriptors $F_s = \{f \mid f : \mathcal{M}_d^{T_i}(\mathbb{R}) \to \mathbb{R}^d\}$ as an example:

Statistical Descriptors

Minimum	Std. Deviation
25th Percentile	Variance
Median	Kurtosis
75th percentile	Skew
Maximum	Coeff. of Variation
Mean	-

Applying each element of F_s results on S_i results in a feature vector $\hat{S}_i \in \mathbb{R}^{d \cdot |F_s|}$

Data Reduction

A flight $F_i \in \mathcal{M}_d^{T_i}(\mathbb{R})$ contains $d \cdot T_i$ elements, the feature vector $\hat{F}_i \in \mathbb{R}^{d|F_s|\tau}$ contains $d|F_s|\tau$ elements, where $|F_s|$ is the number of statistical descriptors used and τ is the number of segments chosen. As an example, use a flight with 5,000 timestamps, 10 features, 10 statistical descriptors and 4 segments:

$$|F_i| = 10 \cdot 5,000 = 50,000 \tag{4}$$

$$|\hat{F}_i| = 10 \cdot 10 \cdot 4 = 400 \tag{5}$$

$$\frac{|F_i|}{|\hat{F}_i|} = 125 \tag{6}$$

Methodology

Pros & Cons

Pros

- Large data reduction, see (6)
- Lower computational load
- Ease feature interpretation
- Reproducible

Cons

- Mixed or uniform MVT segmentation?
- Not a fixed number of stages in each flight
- Which descriptors to use?

RUL Prediction

Interpretable ML models

Given the reduced dataset, one now must select a model to estimate RUL. But how can you choose without prior information?

1. Define a common loss function
2. Apply data transformation techniques [2]
3. Select a suite of **interpretable** ML models
4. Baseline testing (w/o tuning) and ranking
5. Candidate model selection
6. Repeat!

[2] Any transformation must also be interpretable

Model Interpretation
Importance and Benefits

Interpretability enables **transparency** and **accountability**

- Analyze the weights of each feature in linear regression
- Check the splits in a decision tree
- Marginal changes in features affecting log-odds for logistic regression
- Shapley and LIME for "higher models"

Figure: Amazon removes its recruitment algorithm due to gender bias towards men [6]

Model Interpretation

Properties of Shapley values

- **Efficiency:**

$$\sum_{j=1}^{p} \phi_j = \hat{f}(x) - \mathbb{E}[\hat{f}(X)]$$

- **Symmetry:**

$$val(S \cup \{j\}) = val(S \cup \{k\}) \quad \forall S \subseteq \{1, \ldots, p\} - \{j, k\}$$
$$\Rightarrow \phi_j = \phi_k$$

- **Dummy:**

$$val(S \cup \{j\}) = val(S) \quad \forall S \subseteq \{1, \ldots, p\}$$
$$\Rightarrow \phi_j = 0$$

- **Additivity:**

$$\phi_j^{f+g} = \phi_j^f + \phi_j^g$$

All predictions have an inherent level of uncertainty. "A prediction interval provides an estimate range within which a future observation is likely to fall" [7]

Machine Learning Perspective

In ML, they represent the range within which a predicted observation is likely to fall, they estimate the uncertainty of point estimates.

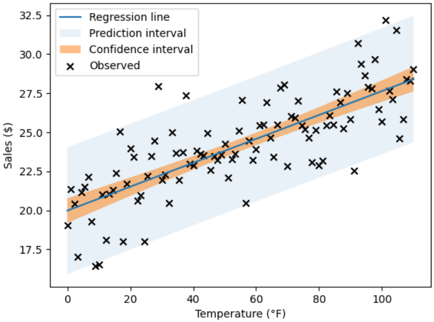


Figure: Difference between CIs and PIs [8]

Let τ be the selected quantile, y the target variable and $\hat{y}$ the predicted quantile. The **Pinball Loss** is defined as:

$$\mathcal{L}_\tau(y, \hat{y}) = (y - \hat{y})\tau \mathbb{1}\{y \geq \hat{y}\} + (\hat{y} - y)(1 - \tau)\mathbb{1}\{\hat{y} > y\} \tag{7}$$

$$\mathcal{Q}_\tau(Y|X) = \underset{q(X)}{\arg\min}\, \mathbb{E}[\mathcal{L}_\tau(Y, q(X))] \tag{8}$$

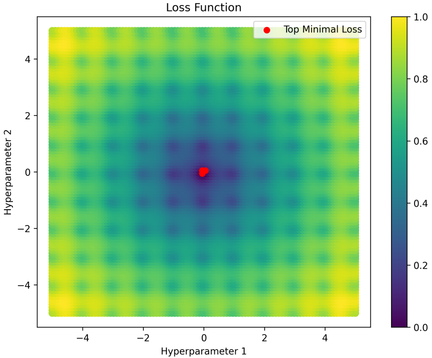
Koenker introduced the concept of quantile regression in 1978, and continued its development up to the Conditional Quantile Functions [9] referenced in (8) which minimizes (7)

Hyperparameter Optimization

Overview

Each dataset and model is different, and there is no rigorous proof on how to determine the optimal hyperparameters for a given problem. We adopt approximated methods divided into 3 categories [10]:

- Grid Search
- Random Search
- **Bayesian Optimization**



Given a search history, it suggests a hyperparameter for the next trial.

- Treats each hyperparameter **independently**
- Processes search history as tuples of (parameter, loss)
- Updates the definition of "good" and "bad" losses
- Defines $g(x)$ and $b(x)$ for good and bad losses

Heuristic

Select the hyperparameter which maximices:

$$x_s = \arg\max_x \frac{g(x)}{b(x)} \tag{9}$$

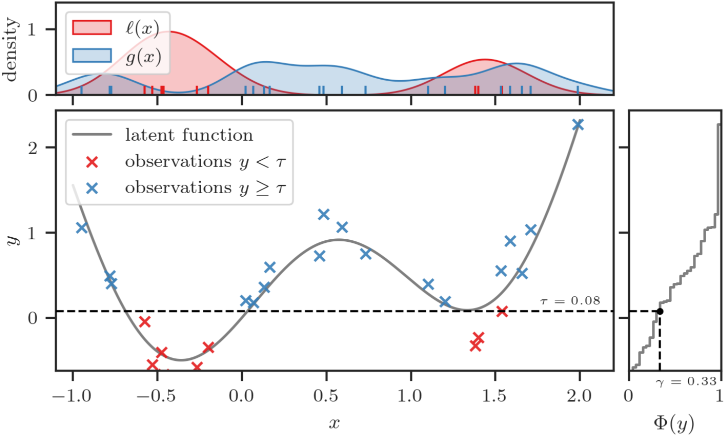


Figure: Example of TPE trial suggestion [11]

Segmentation of Environmental Descriptors

For a test case we apply Binary Segmentation with L2 norm and min. size for segment of 20% the length of the array

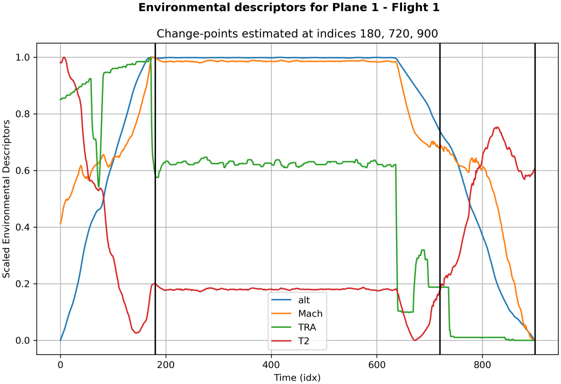


Figure: Environmental descriptors (scaled) for the first flight for plane 1

Flight History of a Plane

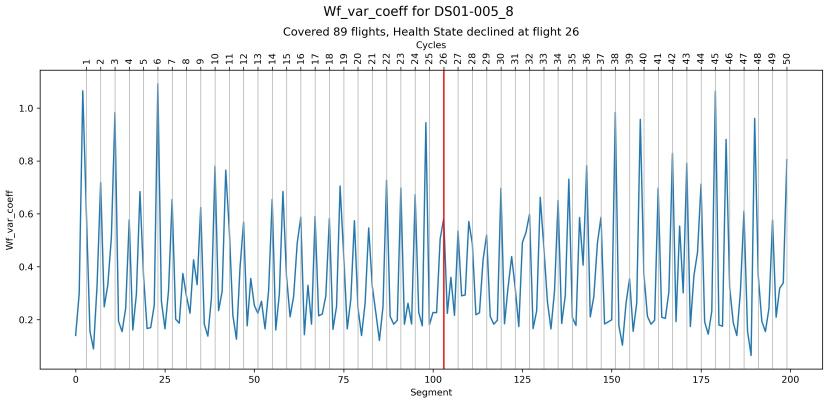


Figure: Coefficient of variation of the Fuel Flow for Unit 8. Flew 89 flights

Could there be a correlation between the early spikes and the relatively early health decline?

Baseline Regression Models

A suite of interpretable ML models is fit to 80% of the resulting dataset, tested on the remaining 20%

Model	Train Loss	Test Loss	Train R2	Test R2	Train RMSE	Test RMSE	Fit Time
Catboost	4.925814	6.81856	0.888255	0.793759	7.918826	11.036592	30.923586
XGBoost	3.992694	6.902025	0.928544	0.78922	6.332358	11.157376	6.510977
LGBM	**4.861673**	**6.983948**	**0.891184**	**0.784365**	**7.81434**	**11.285139**	**2.113664**
Random Forest	2.824662	7.209788	0.967582	0.77276	4.265207	11.584841	216.538052
Ridge	6.840865	7.275478	0.783071	0.766391	11.033299	11.746066	0.05542
Lasso Lars	7.30732	7.420427	0.753777	0.75725	11.754672	11.973661	0.064927
Lasso	7.307333	7.420434	0.753776	0.75725	11.754695	11.973673	0.122003
Elastic Net	7.363291	7.47165	0.750416	0.754214	11.834634	12.048292	0.237217
SVM	7.36937	7.60994	0.748435	0.744574	11.881503	12.282309	13.777651
Decision Tree	0.5	9.758508	1.0	0.625179	0.0	14.878505	3.323056
Dummy (Mean)	17.27968	17.824815	0.0	-0.004084	23.688969	24.351877	0.002806
Linear Regression	5.642092	87743.731314	0.852029	0.635906	9.112421	14.664064	0.257621

1. LGBM achieves similar accuracy as Catboost and XGBoost but in a fraction of their training time
2. Linear models present less overfitting but comparatively lower accuracy

Dimensionality Reduction?

PCA on raw descriptors needs 28 components to explain 99% of the total variance:

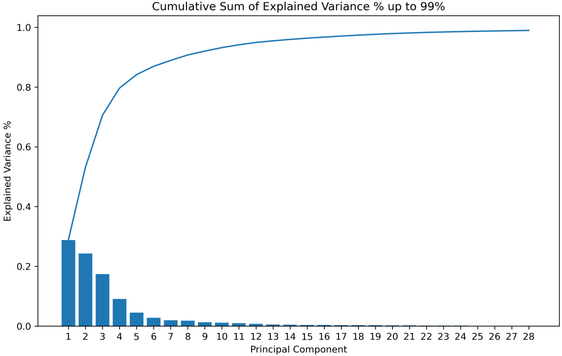


Figure: Explained variance ratio per Principal Component added up to 99%

Scaled PCA needs 93, nonetheless weights should be distributed fairly, doesn't imply better prediction performance!

Effects on Model Accuracy subject to prior Standarization

Using a LGBM regressor as a baseline model, the following treatments were tested on a validation dataset:

Transformation	Test Loss
No PCA	6.983
Raw, all PCs	5.867
Raw, 28 PCs	7.229
Standardized, all PCs	6.786
Standardized, 93 PCs	7.506

PCA enables an overall improvement on model loss, but the model doesn't rely on the *most* important components.

Sample Predictions

Using a basic LGBM model with 100 epochs yields:

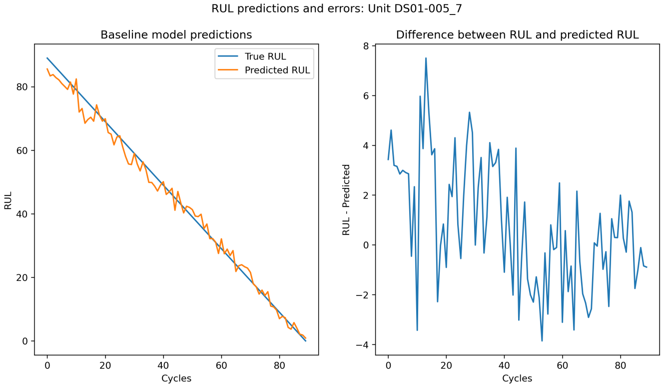


Figure: RUL estimates and deviations from real RUL for plane 7 of dataset DS01-005

Sample Predictions

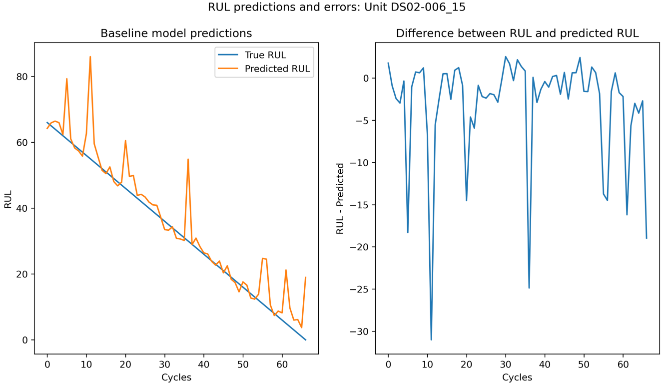


Figure: RUL estimates and deviations from real RUL for plane 15 of dataset DS02-006

Model interpretation

Next we show the overall importance of each Principal Component in RUL regression:

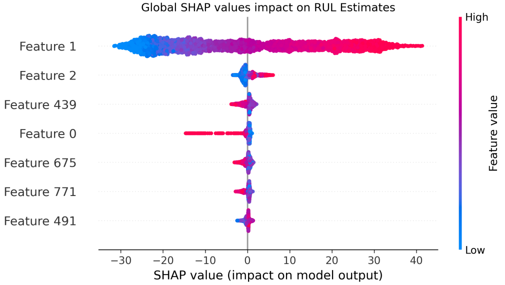


Figure: SHAP values for the Top 7 features (selected by the mean absolute value across all predictions)

The first 3 Principal Components appear in the Top 7, but are not the most important

Instance Explanation

We can also analyze in detail a single RUL estimate:

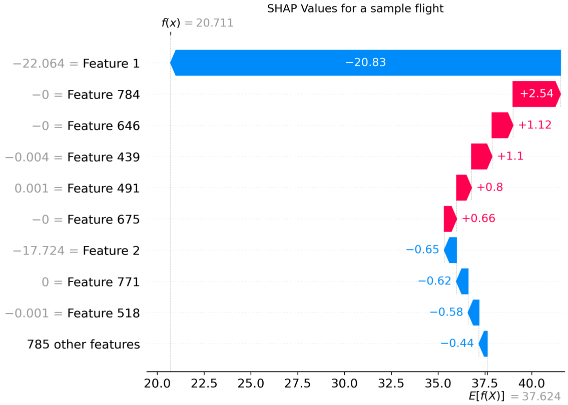


Figure: SHAP values for the Top 7 features (selected by the mean absolute value across all predictions)

Example of estimated prediction intervals

Fitting a base LGBM for RUL estimates and 2 models for quantile regression using pinball loss:

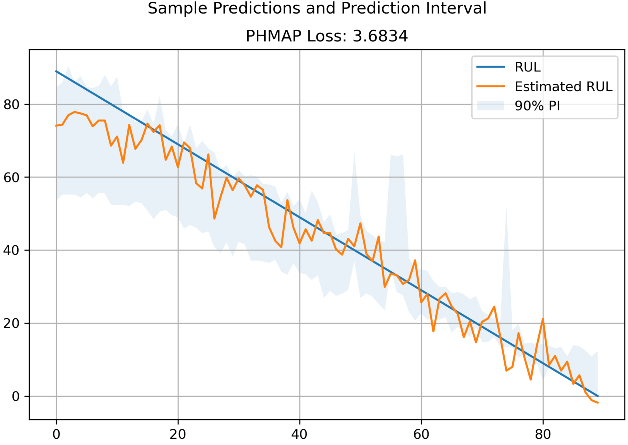


Figure: RUL estimates and a 90% prediction interval for a plane

TPE Hyperparameter Optimization

Defined a common search space with hyperparameters available for XGBoost Regressors. The optimization regime constrained is to 1500 iterations per model, for the following search space:

Parameter	Range
Learning Rate	$\exp(\mathcal{U}(-10, 0))$
Boosting Rounds	$\mathcal{U}(100, 10000)$
Max Depth	$\mathcal{U}(2, 25)$
Min Child Weight	$\mathcal{U}(2, 50)$
Subsample	$\mathcal{U}(0.2, 1)$
Subsample by Tree	$\mathcal{U}(0.2, 1)$
Subsample by Node	$\mathcal{U}(0.2, 1)$
Subsample by Level	$\mathcal{U}(0.2, 1)$
Gamma	$\exp(\mathcal{U}(-10, 10))$
Alpha	$\exp(\mathcal{U}(-10, 10))$
Lambda	$\exp(\mathcal{U}(-10, 10))$

TPE optimizes the mean cross-validated PHMAP test loss across 5 folds

References I

- BBC News, "Boeing 777: Dozens grounded after denver engine failure,", Feb. 22, 2021. [Online]. Available: https://www.bbc.com/news/world-us-canada-56149894.
- A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 international conference on prognostics and health management*, IEEE, 2008, pp. 1–9.
- M. Arias Chao, C. Kulkarni, K. Goebel, and O. Fink, "Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics," *Data*, vol. 6, no. 1, p. 5, 2021.
- PHM Society. (Oct. 21, 2021), 2021 phm conference data challenge - phm society data repository, [Online]. Available: https://data.phmsociety.org/2021-phm-conference-data-challenge/.

References II

C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107 299, 2020.

G. S. Reporter, "Amazon ditched ai recruiting tool that favored men for technical jobs,", Oct. 11, 2018. [Online]. Available: https://www.theguardian.com/technology/2018/oct/10/amazon-hiring-ai-gender-bias-recruiting-engine.

H. Peters, "Prediction Intervals in Machine Learning - Heinrich Peters - Medium,", Jul. 2023. [Online]. Available: https://medium.com/@heinrichpeters/prediction-intervals-in-machine-learning-a2faa36b320c.

L. Cialdella, *Understanding the difference between prediction and confidence intervals for linear models in Python*, Sep. 2020. [Online]. Available: https://lmc2179.github.io/posts/confidence_prediction.html.

References III

R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

AWS. (), What is hyperparameter tuning? - hyperparameter tuning methods explained - aws, [Online]. Available: https://aws.amazon.com/what-is/hyperparameter-tuning/#:~:text=Hyperparameter%20tuning%20allows%20data%20scientists,the%20model%20as%20a%20hyperparameter..

L. C. Tiao, A. Klein, M. W. Seeger, E. V. Bonilla, C. Archambeau, and F. Ramos, "BORE: Bayesian optimization by density-ratio estimation," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 10 289–10 300. [Online]. Available: http://proceedings.mlr.press/v139/tiao21a.html.