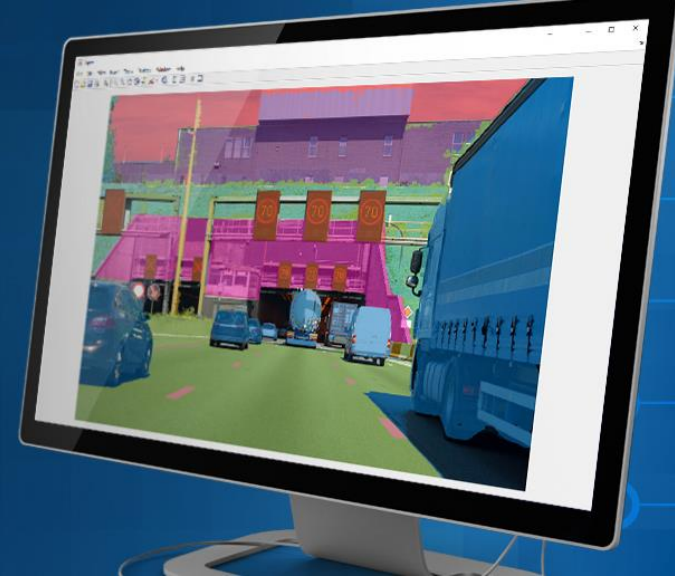


# End-to-end algorithm development for predictive maintenance

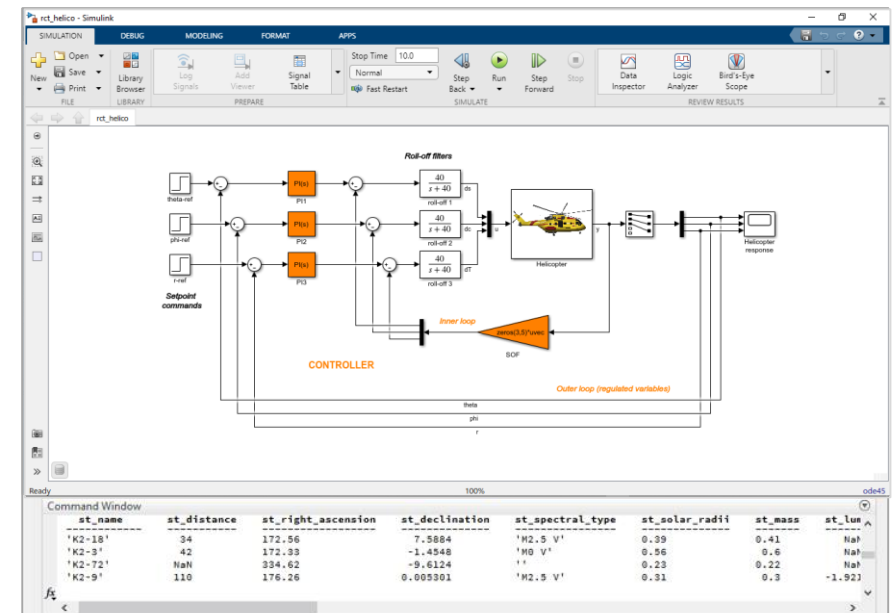
Antti Löytynoja, M.Sc.  
aloytyno@mathworks.com  
*Senior Application Engineer*  
**MathWorks**

# Our Products MATLAB® & SIMULINK®



- MATLAB is a programming environment for algorithm development, data analysis, visualization, and numeric computation.
- Simulink is a graphical environment for designing, simulating, and testing systems.
- More than 100 add-on products for specialized tasks.

## Computer Simulation Toolbox





- Millions of engineers and scientists worldwide use MATLAB and Simulink.



**4 million+**  
users in 185 countries



**100,000+**  
businesses, governments,  
and universities



All of the top 10  
automotive and  
aerospace companies

Knowing AI is an asset

# **The Chairman of Nokia on Ensuring Every Employee Has a Basic Understanding of Machine Learning — Including Him**

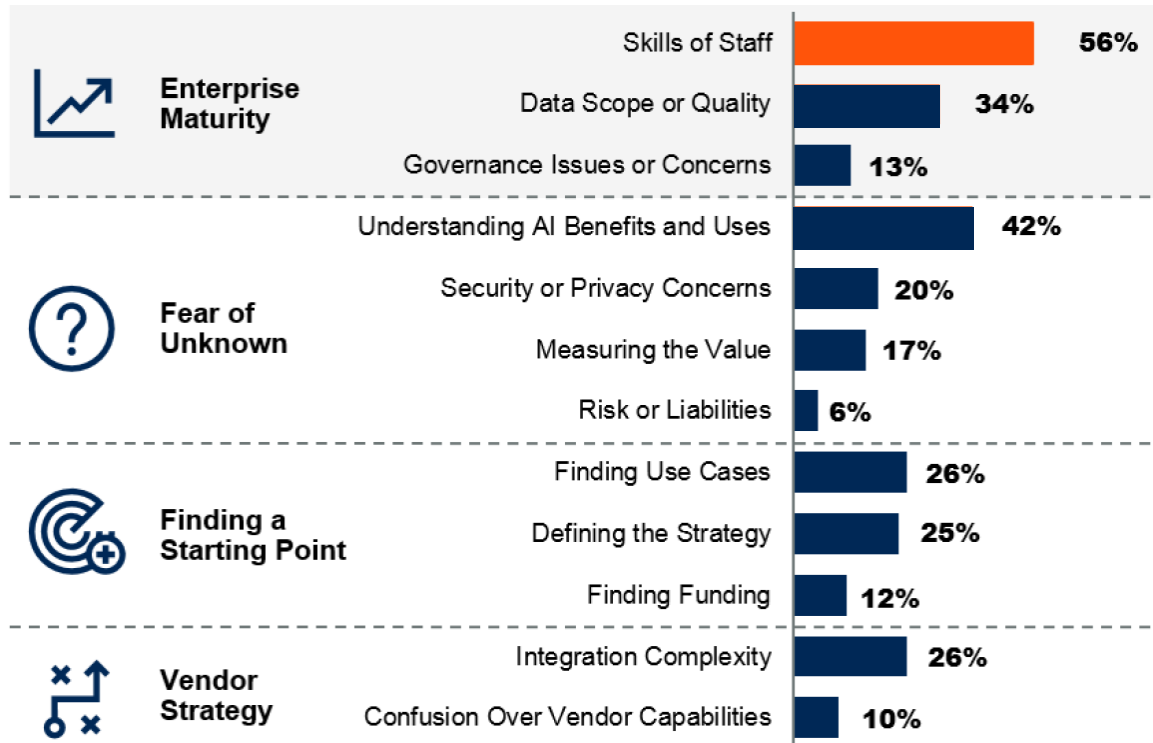
by [Risto Siilasmaa](#)

October 04, 2018

- *Harvard Business Review*

# AI skills is a major concern

## Top Three Challenges to AI and ML Adoption



**Skills of staff** is the top challenge for organizations surveyed by Gartner\*

\* Source: "AI and ML Development Strategies, Motivators and Adoption Challenges," Gartner Research Note, published 19 June 2019

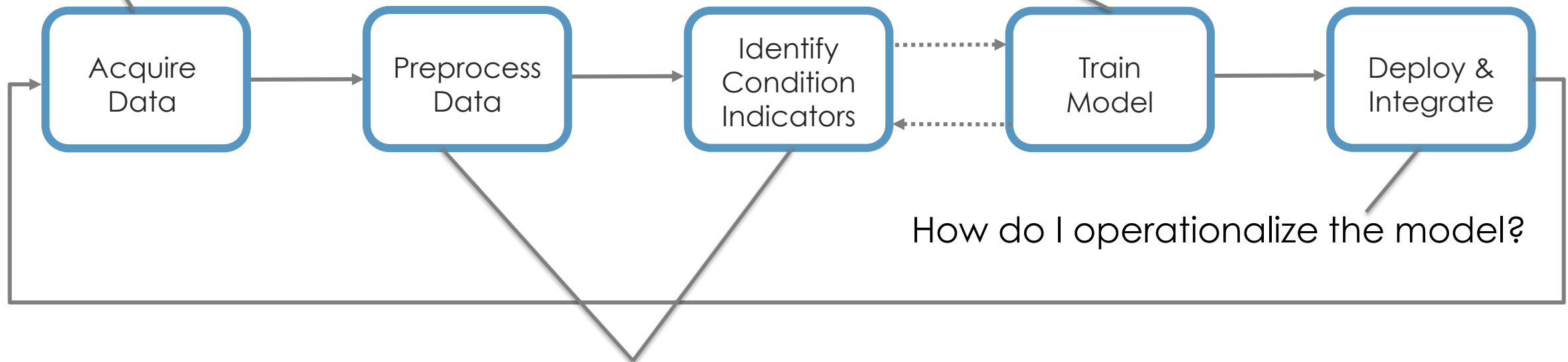
**Tools for every skill level** needed

n = 106  
 Gartner Research Circle members, excluding "unsure"  
 Source: Gartner AI and ML Development Strategies Survey  
 Q: What are the top three challenges or barriers to the adoption of AI and ML within your organization?  
 Rank up to three.  
 ID: 390794

# Workflow for Developing a Predictive Maintenance Algorithm

What if you don't have failure data?

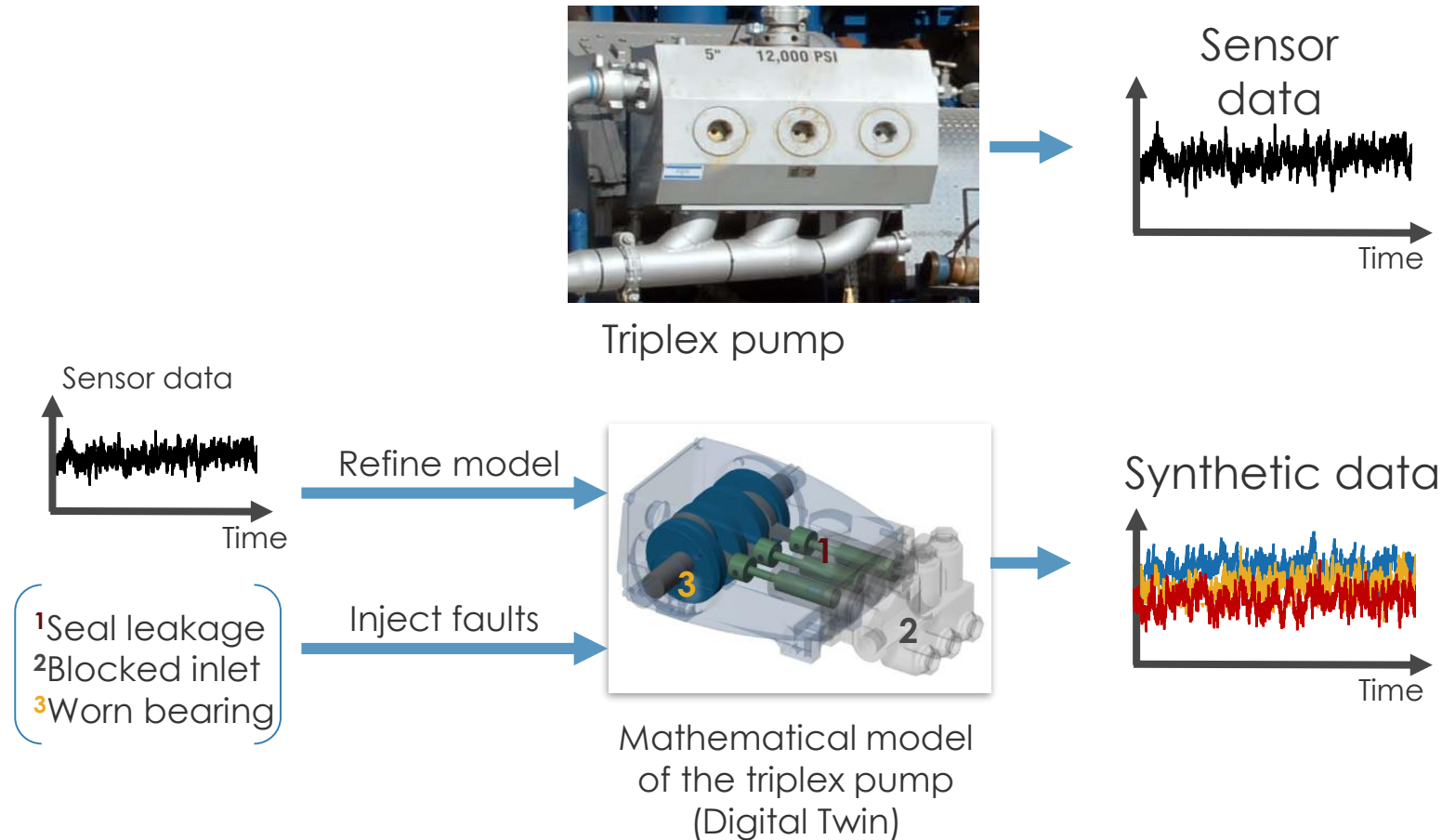
Which model should I use?



How do I operationalize the model?

How do you find good condition indicators?

# 1. What if you don't have failure data?



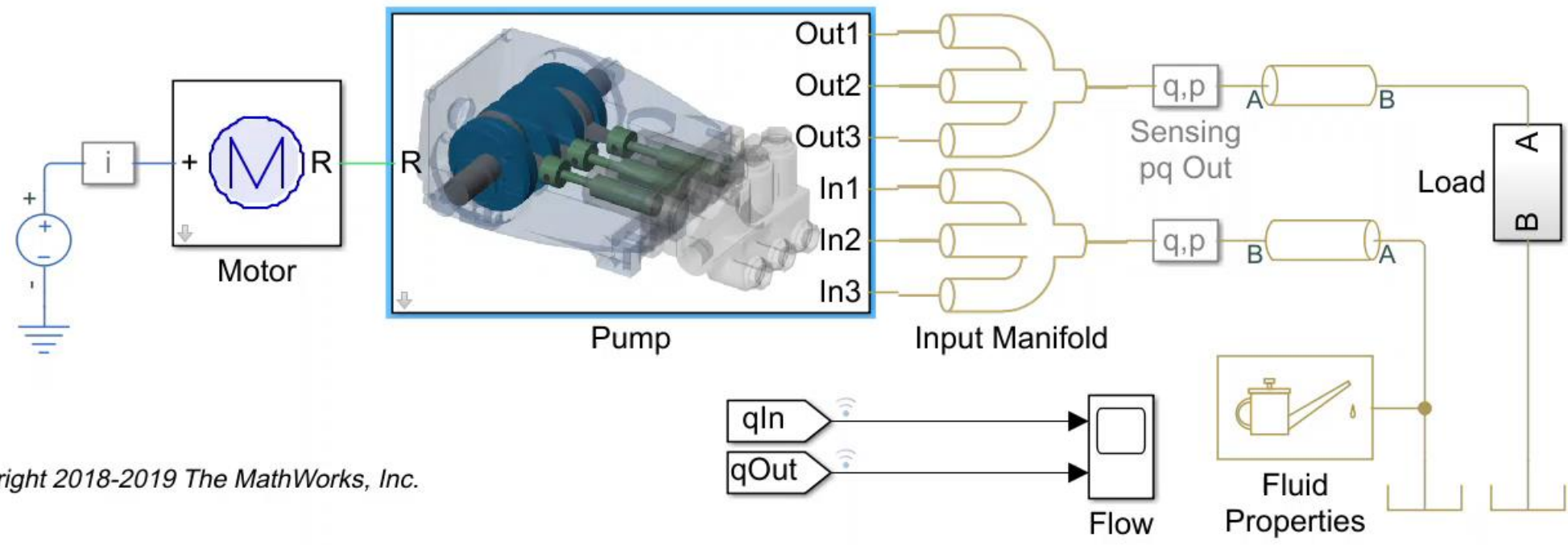
SIMULATION    DEBUG    MODELING    FORMAT    APPS    **BLOCK**

Project    New    Open    Save    Print    Library Browser    Log Signals    Add Viewer    Signal Table    Stop Time: 2    Normal    Step Back    Run    Step Forward    Stop    Data Inspector    Logic Analyzer    Simscape Results ...

PROJECT    FILE    LIBRARY    PREPARE    SIMULATE    REVIEW RESULTS

pdmRecipPump x Pump x Plunger 3 x Plunger 1 x Sensing pq In x

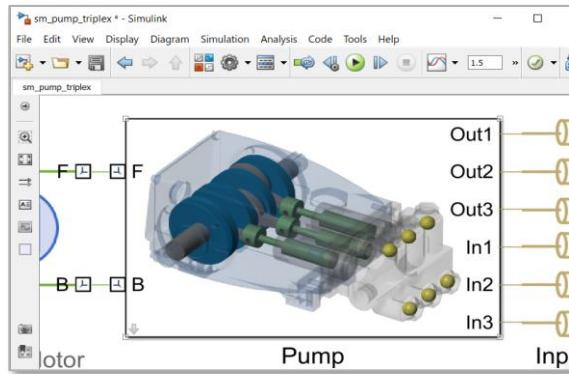
### Triplex Pump with Faults



Copyright 2018-2019 The MathWorks, Inc.



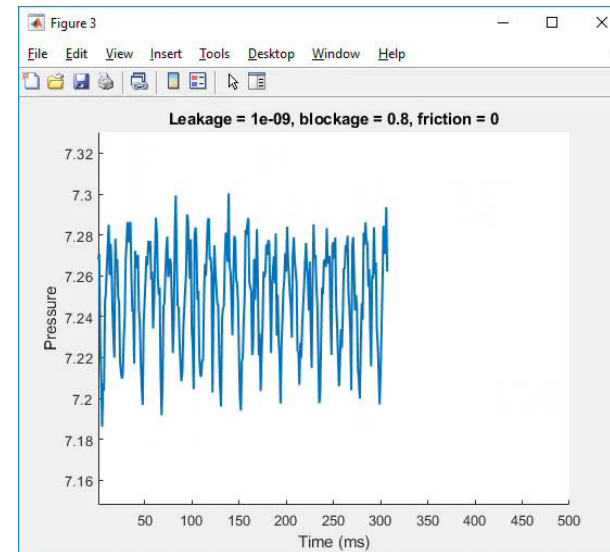
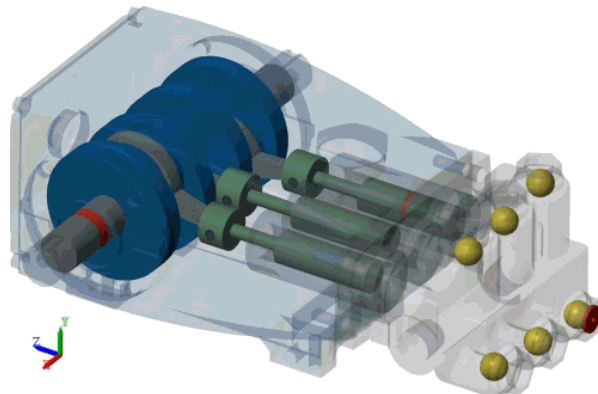
# Simulate data for Condition monitoring with a Digital Twin



**Leak Area = [1e-9 0.036]**

**Bearing Friction = [0 6e-4]**

**Blocking Fault = [0.3 0.8]**



| flow             | pressure         | faultCode |
|------------------|------------------|-----------|
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 0         |
| 1201x1 timetable | 1201x1 timetable | 100       |
| 1201x1 timetable | 1201x1 timetable | 10        |
| 1201x1 timetable | 1201x1 timetable | 10        |
| 1201x1 timetable | 1201x1 timetable | 1         |
| 1201x1 timetable | 1201x1 timetable | 11        |
| 1201x1 timetable | 1201x1 timetable | 11        |
| 1201x1 timetable | 1201x1 timetable | 10        |
| 1201x1 timetable | 1201x1 timetable | 1         |

## 2. How do you find good condition indicators?

Diagnostic Feature Designer - Feature Ranking: FeatureTable1

FEATURE DESIGNER | FEATURE RANKING | VIEW

Classification Ranking | Prognostic Ranking | Rank By: faultCode | Sort By: One-way ANOVA | Delete Scores | Export

ANALYZE | CONDITION | SORT | SCORE | EXPORT

Data Browser: Signal Trace: pressure/Data | Histogram: FeatureTable1 | Power Spectrum: pressure\_ps/Data | Feature Ranking: FeatureTable1

▼ Signals & Spectra

- pressure/Data
- pressure\_ps/Data

▼ Feature Tables

FeatureTable1

Features:

- pressure\_stats/Data\_ClearanceFactor
- pressure\_stats/Data\_CrestFactor
- pressure\_stats/Data\_ImpulseFactor
- pressure\_stats/Data\_Kurtosis
- pressure\_stats/Data\_Mean
- pressure\_stats/Data\_PeakValue
- pressure\_stats/Data\_RMS

▼ Datasets

- Ensemble1

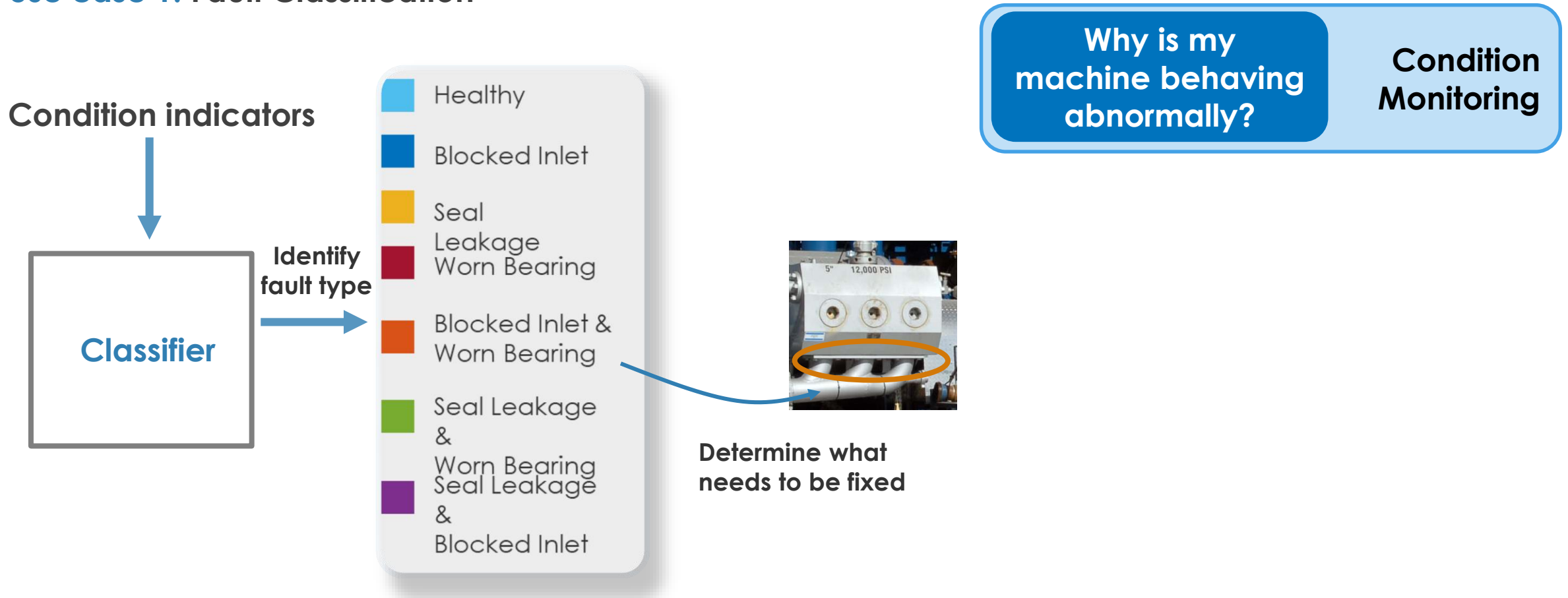
**Features Sorted by Importance**

| Feature                       | One-way ANOVA |
|-------------------------------|---------------|
| pre...ts/Data_RMS             | 124.5145      |
| pre...ts/Data_Mean            | 124.2274      |
| pre...ec/Data_PeakAmp1        | 105.5479      |
| pre...ts/Data_THD             | 63.6488       |
| pre...ts/Data_CrestFactor     | 61.9233       |
| pre...ts/Data_ImpulseFactor   | 61.8776       |
| pre...ts/Data_ClearanceFactor | 61.8542       |
| pre...ec/Data_PeakAmp2        | 47.9822       |
| pre...ec/Data_PeakFreq1       | 47.5535       |
| pre...ts/Data_SNR             | 37.5568       |
| pre...ts/Data_Skewness        | 29.7011       |
| pre...ts/Data_Std             | 26.2064       |
| pre...ts/Data_PeakValue       | 25.5898       |
| pre...ts/Data_ShapeFactor     | 22.2552       |
| pre...ts/Data_SINAD           | 20.3739       |
| pre...ec/Data_PeakFreq2       | 17.1300       |
| pre...ts/Data_Kurtosis        | 12.8701       |
| pre...ec/Data_BandPower       | 11.9010       |

Feature ranking plot for "FeatureTable1" is in focus.

### 3. Which Machine Learning model should I use?

#### Use case 1: Fault Classification



# 3. Which Machine Learning model should I use?

Classification Learner - Confusion Matrix

CLASSIFICATION LEARNER VIEW

FILE FEATURES OPTIONS MODEL TYPE TRAINING PLOTS EXPORT

Data Browser

▼ History

- Last change: Kernel Naive Bayes 9/9 features
- 2.8 ☆ SVM Accuracy: 92.8% Last change: Linear SVM 9/9 features
- 2.9 ☆ SVM Accuracy: 94.1% Last change: Quadratic SVM 9/9 features
- 2.10 ☆ SVM Accuracy: 95.3%** Last change: Cubic SVM 9/9 features
- 2.11 ☆ SVM Accuracy: 92.3% Last change: Fine Gaussian SVM 9/9 features
- 2.12 ☆ SVM Accuracy: 92.0% Last change: Medium Gaussian SVM 9/9 features
- 2.13 ☆ SVM Accuracy: 87.0% Last change: Coarse Gaussian SVM 9/9 features
- 2.14 ☆ KNN Accuracy: 91.0% Last change: Fine KNN 9/9 features
- 2.15 ☆ KNN Accuracy: 90.9% Last change: Medium KNN 9/9 features

▼ Current Model

**Model 2.10: Trained**

**Results**

- Accuracy 95.3%
- Total misclassification cost 125
- Prediction speed ~45000 obs/sec
- Training time 68.069 sec

**Model Type**

- Preset: Cubic SVM
- Kernel function: Cubic
- Kernel scale: Automatic
- Box constraint level: 1

**Model 2.10**

| True class \ Predicted class | 0   | 1   | 2   | 3   | 4   | 5   | True Positive Rate | False Negative Rate |
|------------------------------|-----|-----|-----|-----|-----|-----|--------------------|---------------------|
| 0                            | 94% | 6%  |     |     |     |     | 94%                | 6%                  |
| 1                            | 5%  | 92% | 2%  |     |     |     | 92%                | 8%                  |
| 2                            |     | 4%  | 96% |     |     |     | 96%                | 4%                  |
| 3                            |     |     |     | 98% | 2%  |     | 98%                | 2%                  |
| 4                            |     |     |     | 2%  | 97% | 2%  | 97%                | 3%                  |
| 5                            |     |     |     |     | 5%  | 95% | 95%                | 5%                  |

Plot

- Number of observations
- True Positive Rates
- False Negative Rates
- Positive Predictive Values
- False Discovery Rates

[What is the confusion matrix?](#)

Data set: featureDataTraining Observations: 2652 Size: 107 kB Predictors: 9 Response: FaultCode Response Classes: 6 Validation: 5-fold Cross-Validation

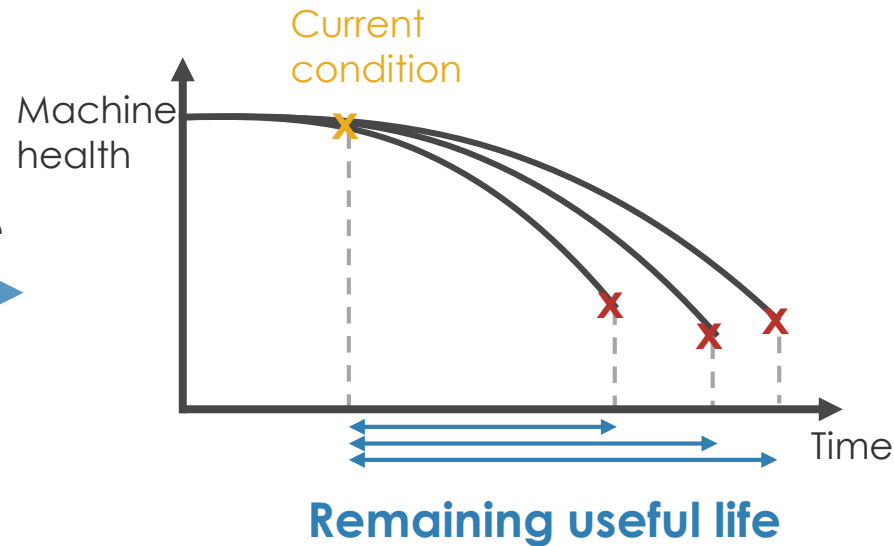
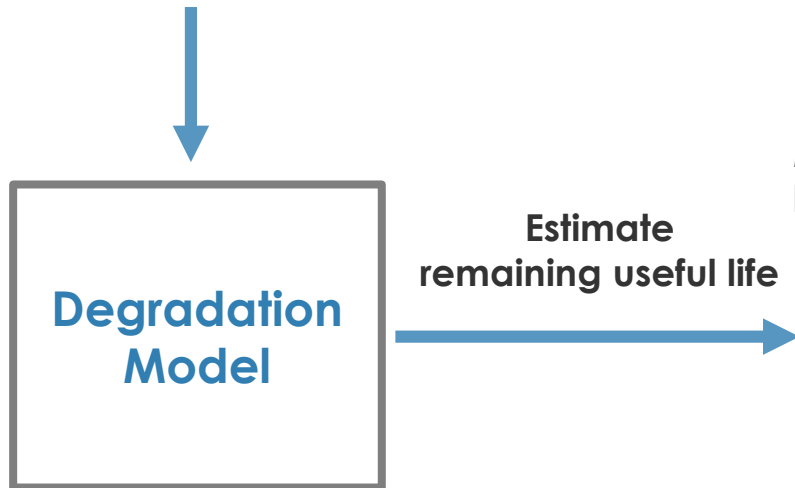
# Remaining Useful Life Estimation

## Use case 2: Prognostics

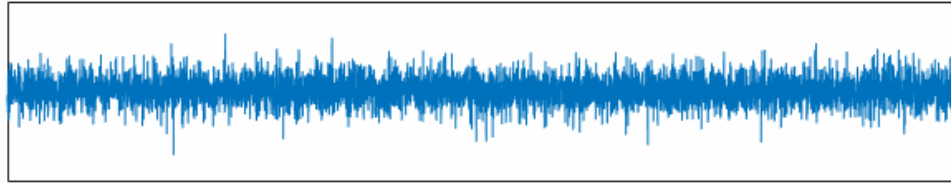
**How much longer can I operate my machine ?**

**Remaining Useful Life Estimation**

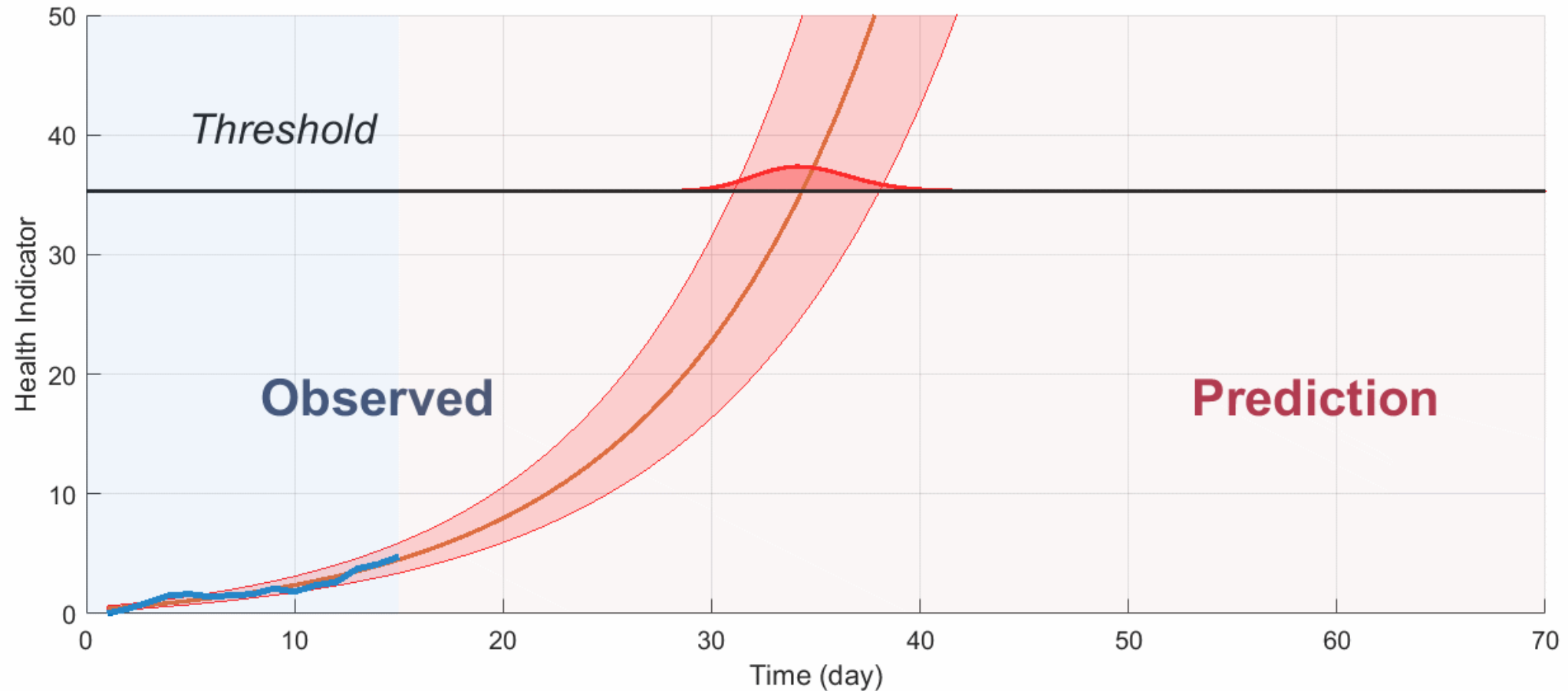
Condition indicators



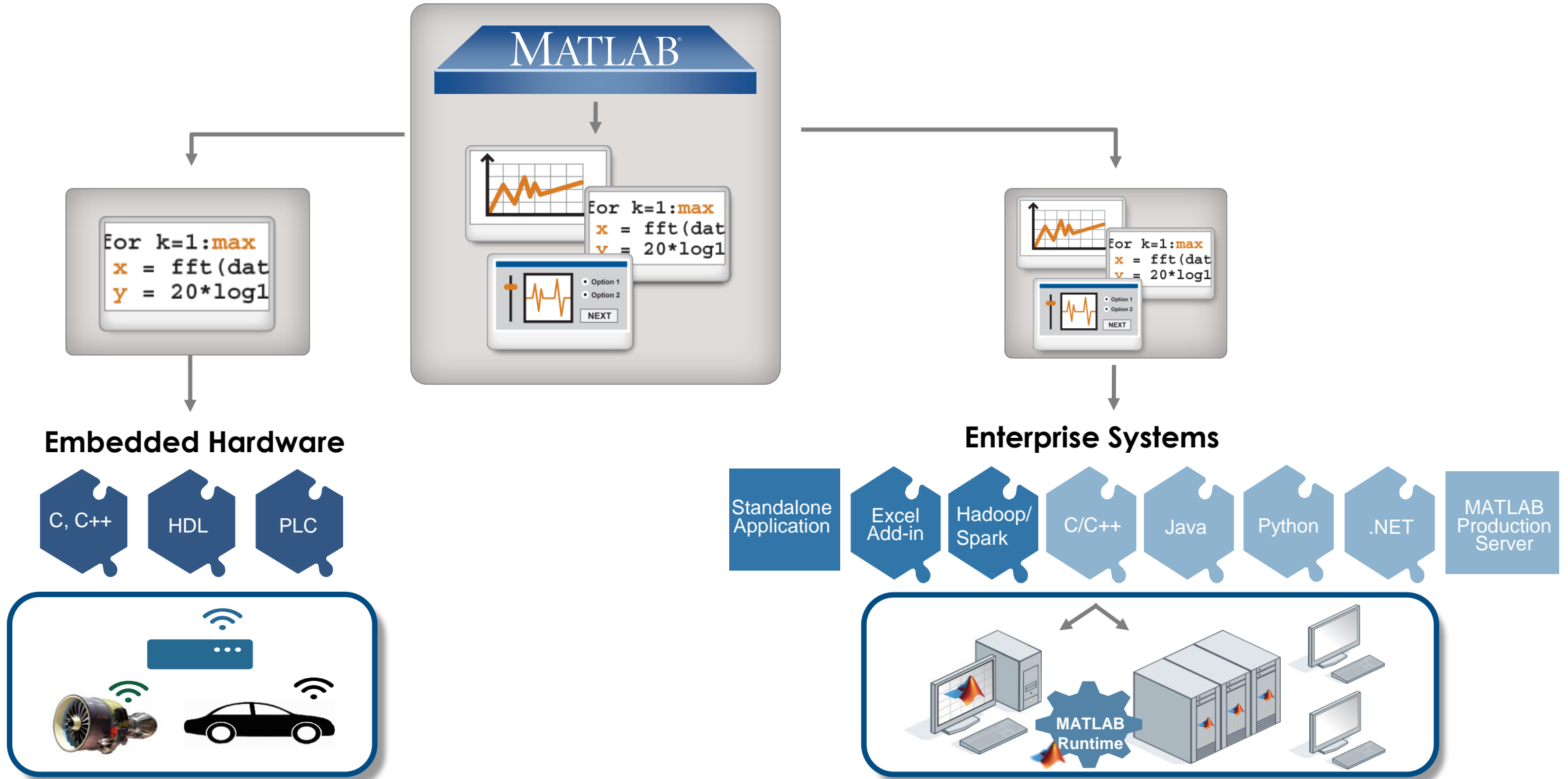
# Remaining Useful Life Estimation



RUL: 459 hours  
(95%CI: 374-558 hours)



# 4. How do I operationalize the model?



# Enable your workforce to develop AI capabilities

## BMW Uses Machine Learning to Detect Oversteering

### Challenge

Develop automated software for detecting oversteering

### Solution

Use interactive apps to quickly explore different machine learning models, and identify the most accurate one.  
Automatically generate C code for the model.

### Results

- Oversteering identified with greater than 98% accuracy
- Multiple machine learning classifiers trained automatically
- Model deployed to an ECU for real-time, in-vehicle testing



A BMW M4 oversteering on a test track.

*“Working in MATLAB, we developed a supervised machine learning model as a proof of concept. **Despite having little previous experience with machine learning, in just three weeks we completed a working ECU prototype capable of detecting oversteering with over 98% accuracy.**”*

*- Tobias Freudling, BMW Group*



# MathWorks can help you get started **TODAY**

- [Examples](#)
- [Documentation](#)
- [Consulting](#)
- [Tech Talk Series](#)

**Documentation** All More ▾ Search Help

CONTENTS

**Predictive Maintenance Toolbox**

Design and test condition monitoring and predictive maintenance algorithms

Predictive Maintenance Toolbox™ lets you label data and estimate the remaining useful life (RUL) of a machine. The toolbox provides functions and an interactive app for ranking features using data-based and model-based methods, such as bearings and gearboxes by extracting features using spectral, and time-series analysis. You can monitor time-series data using frequency and time-frequency methods. To estimate RUL, you can use survival, similarity, and trend-based models.

You can analyze and label sensor data imported from distributed file systems. You can also label simulated Simulink® models. The toolbox includes reference examples for batteries, and other machines that can be reused for maintenance and condition monitoring algorithms.

**Getting Started**  
Learn the basics of Predictive Maintenance Toolbox

**Manage System Data**  
Import measured data, generate simulated data, organize data

**Preprocess Data**  
Clean and transform data to prepare it for extracting features

**Identify Condition Indicators**  
Explore data at the command line or in the app to identify features

**Detect and Predict Faults**  
Train decision models for condition monitoring and fault detection

**Deploy Predictive Maintenance Algorithms**  
Implement and deploy condition-monitoring and predictive maintenance algorithms

**Documentation** All More ▾ Search Help

CONTENTS

**Detect and Diagnose Faults**

**Fault Diagnosis of Centrifugal Pumps Using Steady State Experiments**

Use a model-based approach for detection and diagnosis of different types of faults in a pumping system.

[Open Live Script](#)

**Fault Diagnosis of Centrifugal Pumps Using Residual Analysis**

Use a model parity-equations-based approach for detection and diagnosis of faults in a pumping system.

[Open Live Script](#)

**Multi-Class Fault Detection Using Simulated Data**

Use a Simulink model to generate faulty and healthy data, and use the data to develop a multi-class classifier to detect different faults.

[Open Live Script](#)

**Analyze and Select Features for Pump Diagnostics**

Use the Diagnostic Feature Designer app to analyze and select features to diagnose faults in a triplex reciprocating pump.

[Open Live Script](#)

**Fault Detection Using an Extended Kalman Filter**

Use an extended Kalman filter for online estimation of the friction of a simple DC motor. Significant changes in the estimated friction are detected as faults.

[Open Script](#)

**Fault Detection Using Data Based Models**

Use a data-based modeling approach for fault detection.

[Open Script](#)

## Summary – enablers for Predictive Maintenance

1. Digital Twins can be used to generate missing failure data
  - Most companies designing machines already have simulation models
2. High-level language and apps speed-up development
  - Non-programmers and non-ML specialists can contribute
3. Specialized tools for various domains, e.g. Predictive Maintenance
4. Flexible deployment options
  - Embedded, edge or enterprise