

# Predictive Maintenance with MATLAB

## A data-driven approach

**Antti Löytynoja,**  
**Senior Application Engineer**

# Why perform predictive maintenance?

- Example: faulty braking system leads to windmill disaster
  - <https://youtu.be/-YJuFvjtM0s?t=39s>
- Your equipment can cost millions of dollars
- Failures can be dangerous
- Maintenance also very expensive and dangerous



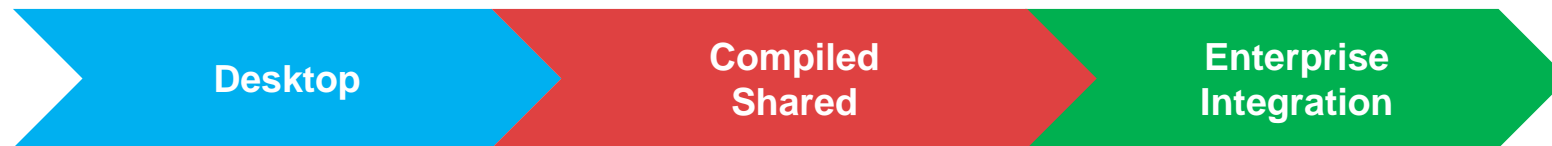
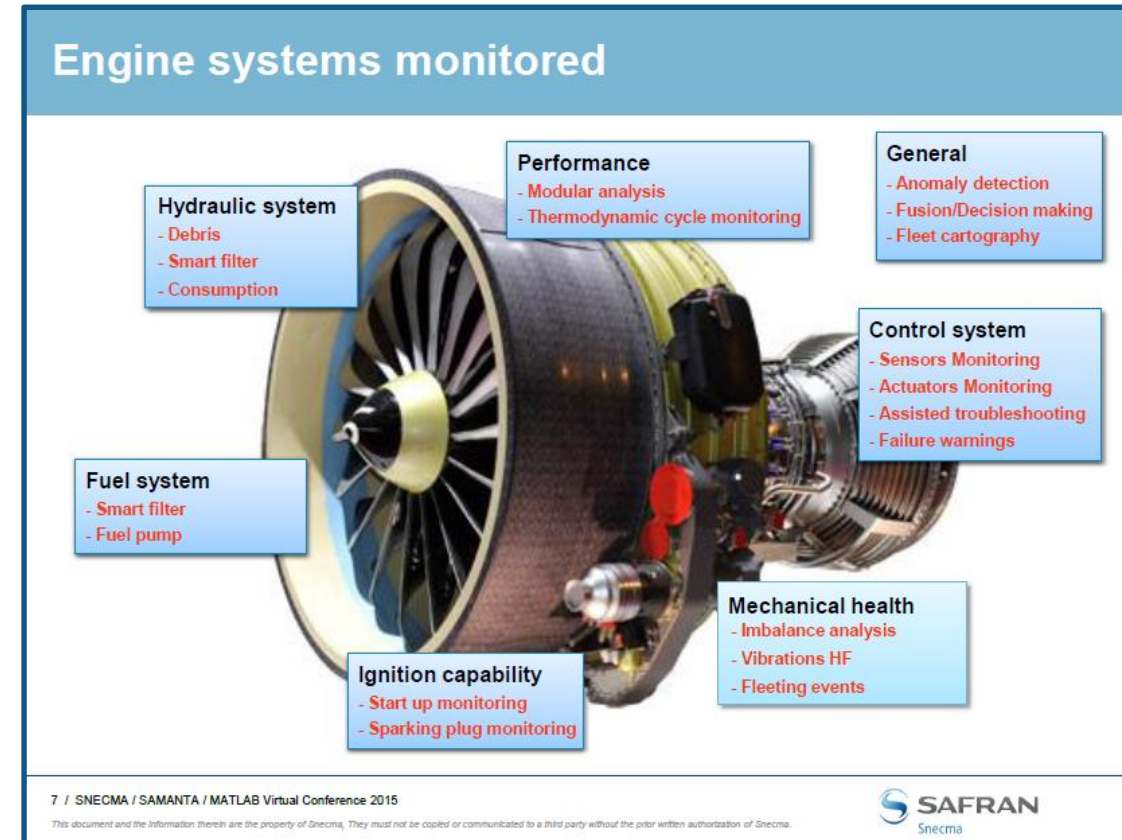
# Types of Maintenance

- Reactive – Do maintenance once there's a problem
  - Example: replace car battery when it has a problem
  - Problem: unexpected failures can be expensive and potentially dangerous
  
- Preventive – Do maintenance at a regular rate
  - Example: change car's oil every 5,000 miles
  - Problem: unnecessary maintenance can be wasteful; may not eliminate all failures
  
- Predictive – Forecast when problems will arise
  - Example: certain GM car models forecast problems with the battery, fuel pump, and starter motor
  - Problem: difficult to make accurate forecasts for complex equipment

# What Does Success Look Like?

## Safran Engine Health Monitoring Solution

- Monitor Systems
  - Detect failure indicators
  - Predict time to maintenance
  - Identify components
- Improve Aircraft Availability
  - On time departures and arrivals
  - Plan and optimize maintenance
  - Reduce engine out-of-service time
- Reduce Maintenance Costs
  - Troubleshooting assistance
  - Limit secondary damage



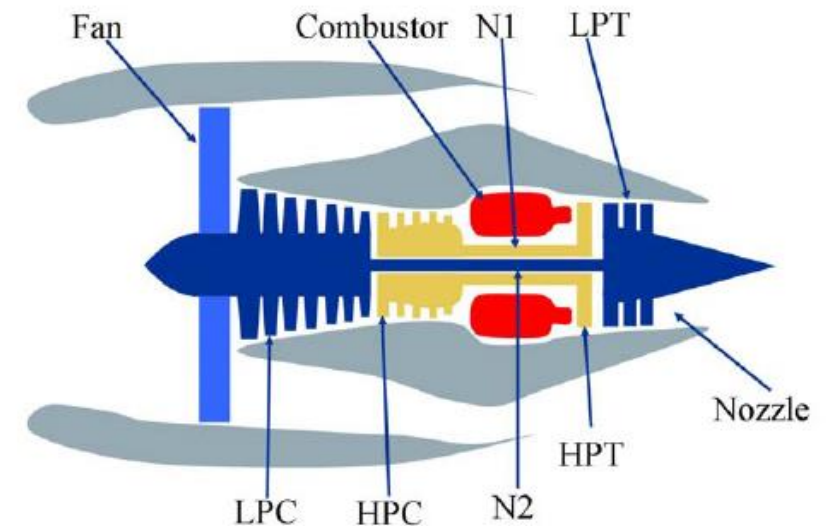
- Ad-hoc data analysis
- Analytics to predict failure
- Suite of MATLAB Analytics
- Shared with other teams
- Proof of readiness
- Real-time analytics
- Integrated with maintenance and service systems

# Predictive Maintenance of Turbofan Engine

Sensor data from 100 engines of the same model

Predict and fix failures before they arise

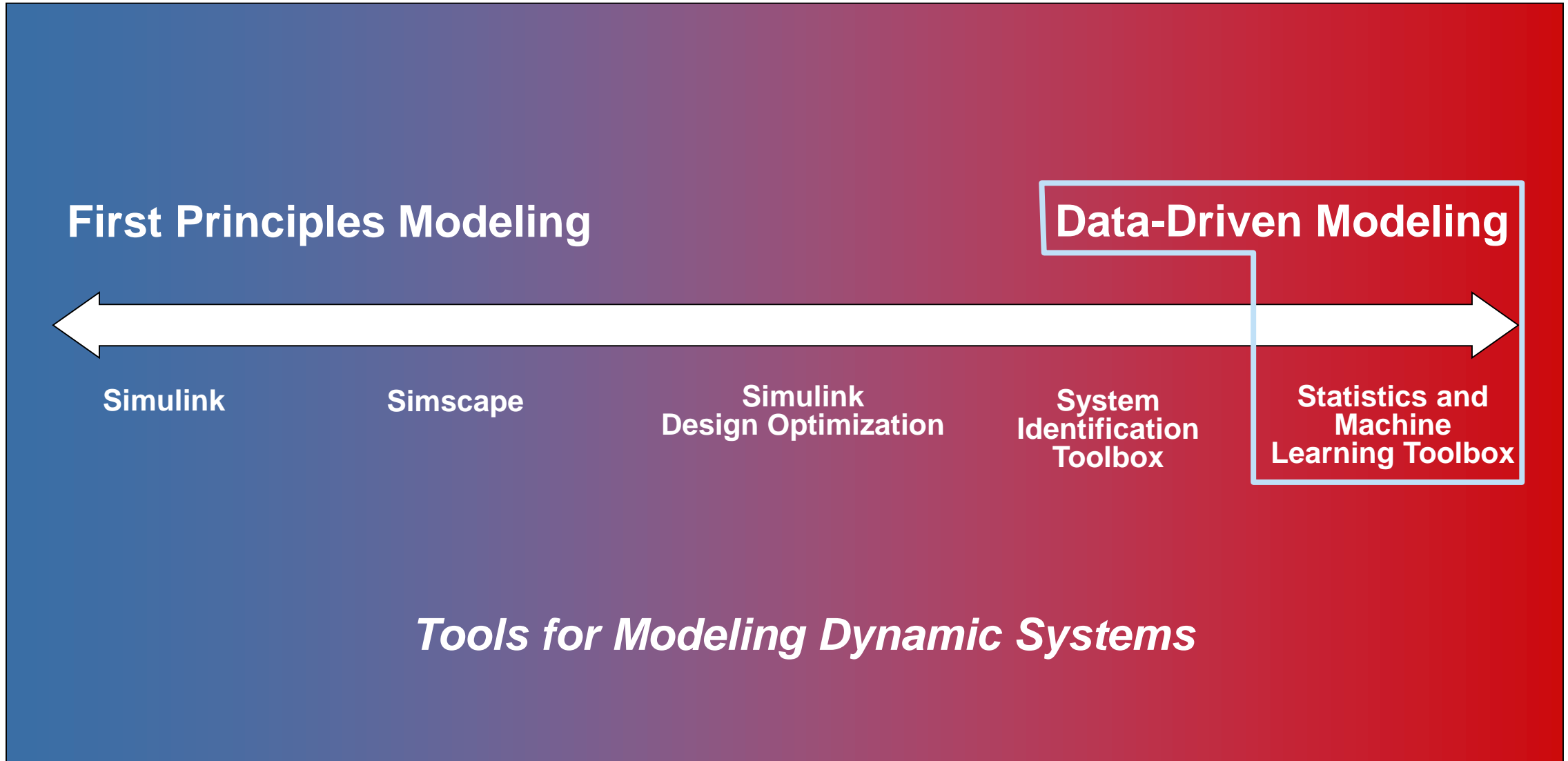
- Import and analyze historical sensor data
- Train model to predict when failures will occur
- Deploy model to run on live sensor data
- Predict failures in real time



Data provided by NASA PCoE

<http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/>

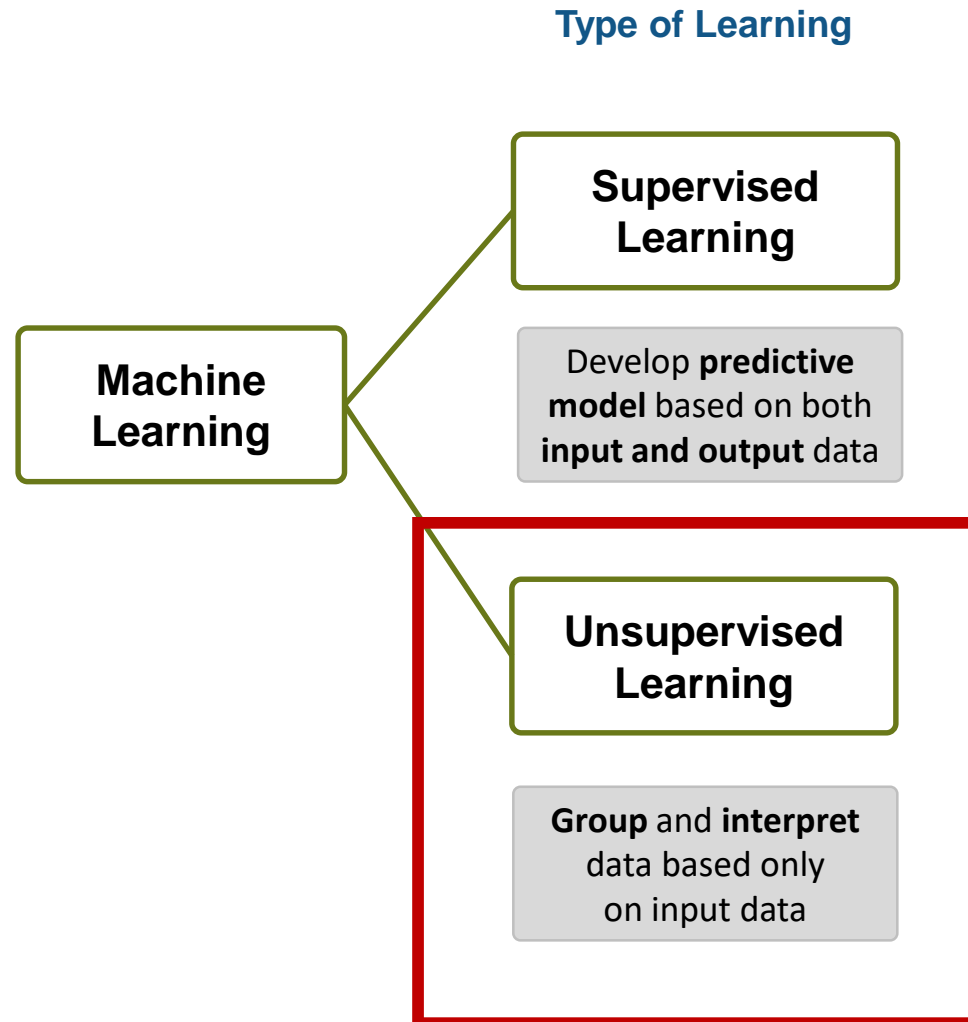
# Modeling Approaches



# Challenges

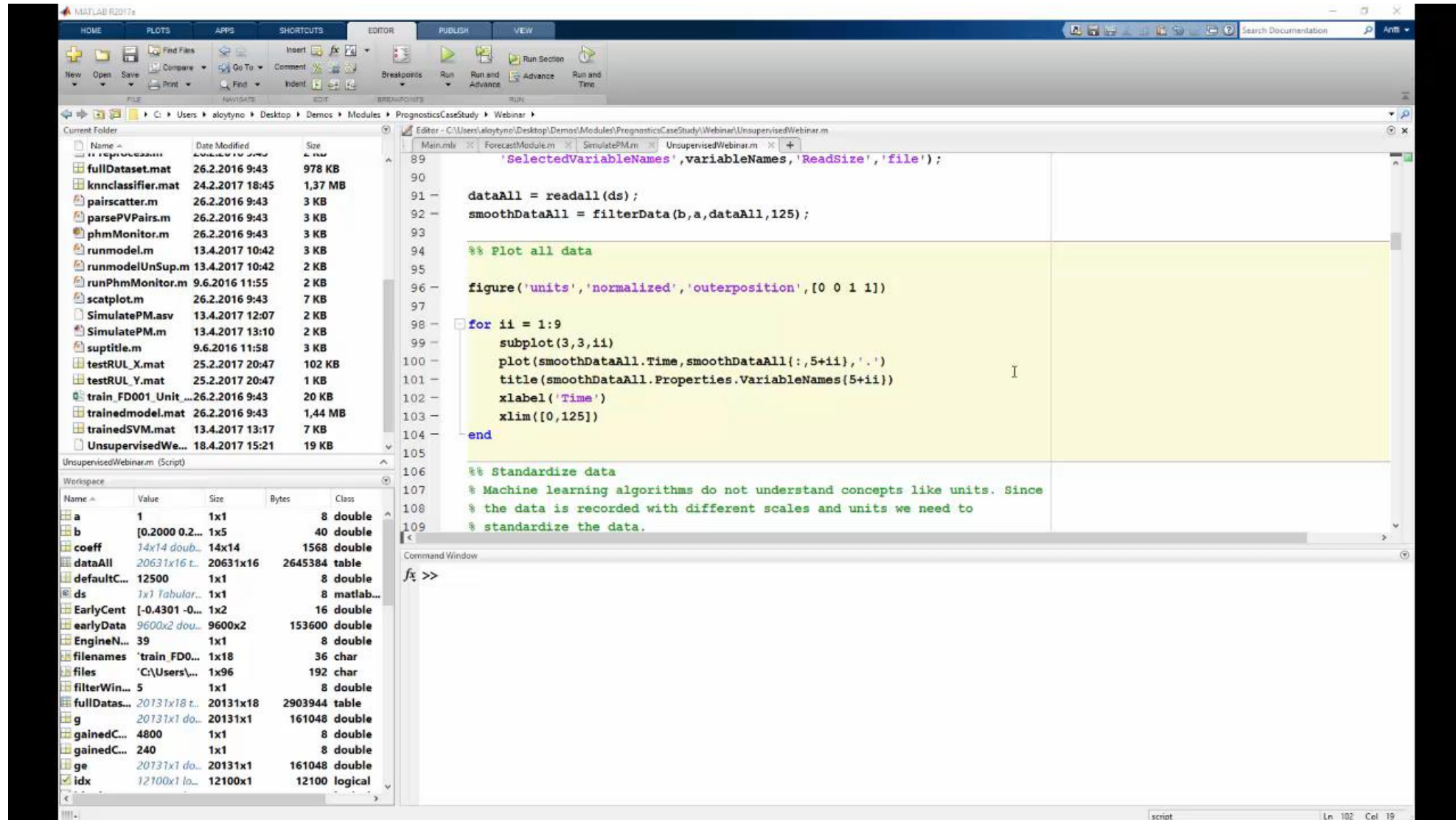
1. Data – Do you have enough/correct data?  
A failure might be a rare occurrence – how do you develop an algorithm if you don't know what a failure looks like
2. How do you find the best possible algorithm?
3. How do you deploy your algorithm into production?

# Overview – Machine Learning





# Using Unsupervised Machine Learning to Detect Deterioration of an Engine



The image shows the MATLAB R2017a environment with a script titled 'UnsupervisedWebinar.m' open in the Editor. The script is located at 'C:\Users\aloytyno\Desktop\Demos\Modules\PrognosticsCaseStudy\Webinar\UnsupervisedWebinar.m'. The script performs the following operations:

- Line 89: `'SelectedVariableNames', variableNames, 'ReadSize', 'file');`
- Line 90: `dataAll = readall(ds);`
- Line 92: `smoothDataAll = filterData(b,a,dataAll,125);`
- Line 94: `%% Plot all data`
- Line 96: `figure('units','normalized','outerposition',[0 0 1 1])`
- Line 98: `for ii = 1:9`
- Line 99: `subplot(3,3,ii)`
- Line 100: `plot(smoothDataAll.Time,smoothDataAll(:,5+ii),'.')`
- Line 101: `title(smoothDataAll.Properties.VariableNames{5+ii})`
- Line 102: `xlabel('Time')`
- Line 103: `xlim([0,125])`
- Line 104: `end`
- Line 106: `%% Standardize data`
- Line 107: `%% Machine learning algorithms do not understand concepts like units. Since`
- Line 108: `%% the data is recorded with different scales and units we need to`
- Line 109: `%% standardize the data.`

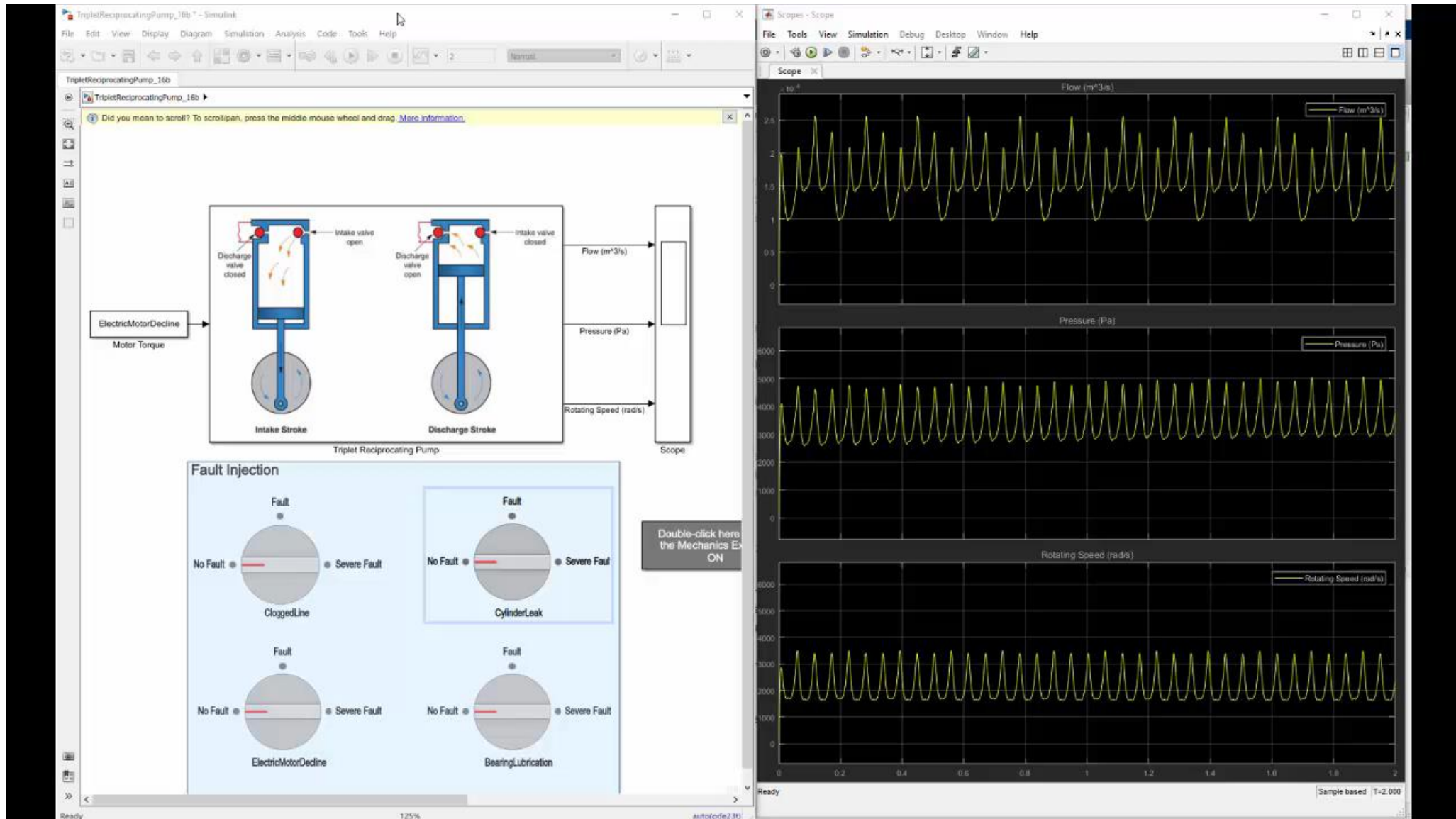
The Command Window at the bottom shows the prompt `>>`. The Workspace window on the left displays the following variables:

Name	Value	Size	Bytes	Class
a	1	1x1	8	double
b	[0.2000 0.2...	1x5	40	double
coeff	14x14 doub...	14x14	1568	double
dataAll	2063 1x16 t...	2063x16	2645384	table
defaultC...	12500	1x1	8	double
ds	1x1 Tabular...	1x1	8	matlab...
EarlyCent	[-0.4301 -0...	1x2	16	double
earlyData	9600x2 dou...	9600x2	153600	double
EngineN...	39	1x1	8	double
filenames	'train FD0...	1x18	36	char
files	'C:\Users\...	1x96	192	char
filterWin...	5	1x1	8	double
fullData...	2013 1x18 t...	2013x18	2903944	table
g	2013 1x1 do...	2013x1	161048	double
gainedC...	4800	1x1	8	double
gainedC...	240	1x1	8	double
ge	2013 1x1 do...	2013x1	161048	double
idx	12100x1 lo...	12100x1	12100	logical

# Generating Datasets for Model Training Through Simulation

- If you don't have real data available, consider generating data through simulation
- Model your system in Simulink, introduce errors (e.g. clogged hydraulics line), log the output of the simulation
- Use the generated dataset to develop a model to predict e.g. remaining useful lifetime

# Generating Datasets for Model Training Through Simulation



# Predictive Maintenance of Turbofan Engine

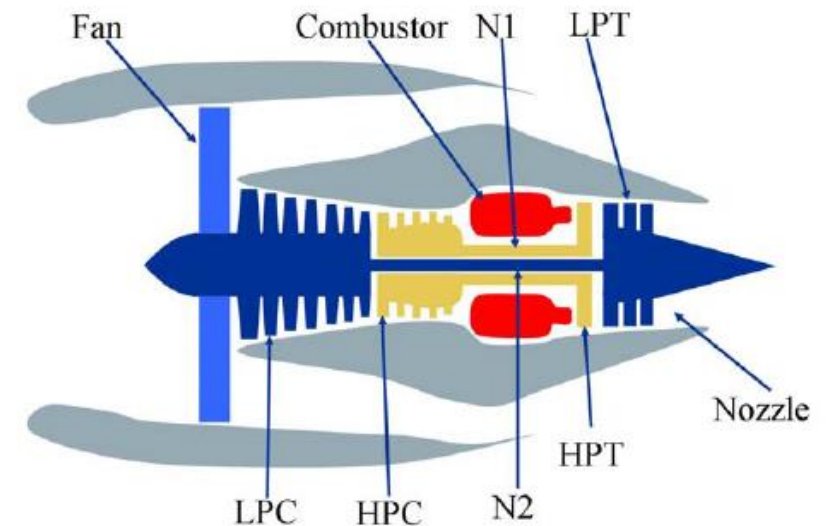
Sensor data from 100 engines of the same model

## Scenario 2: Have failure data

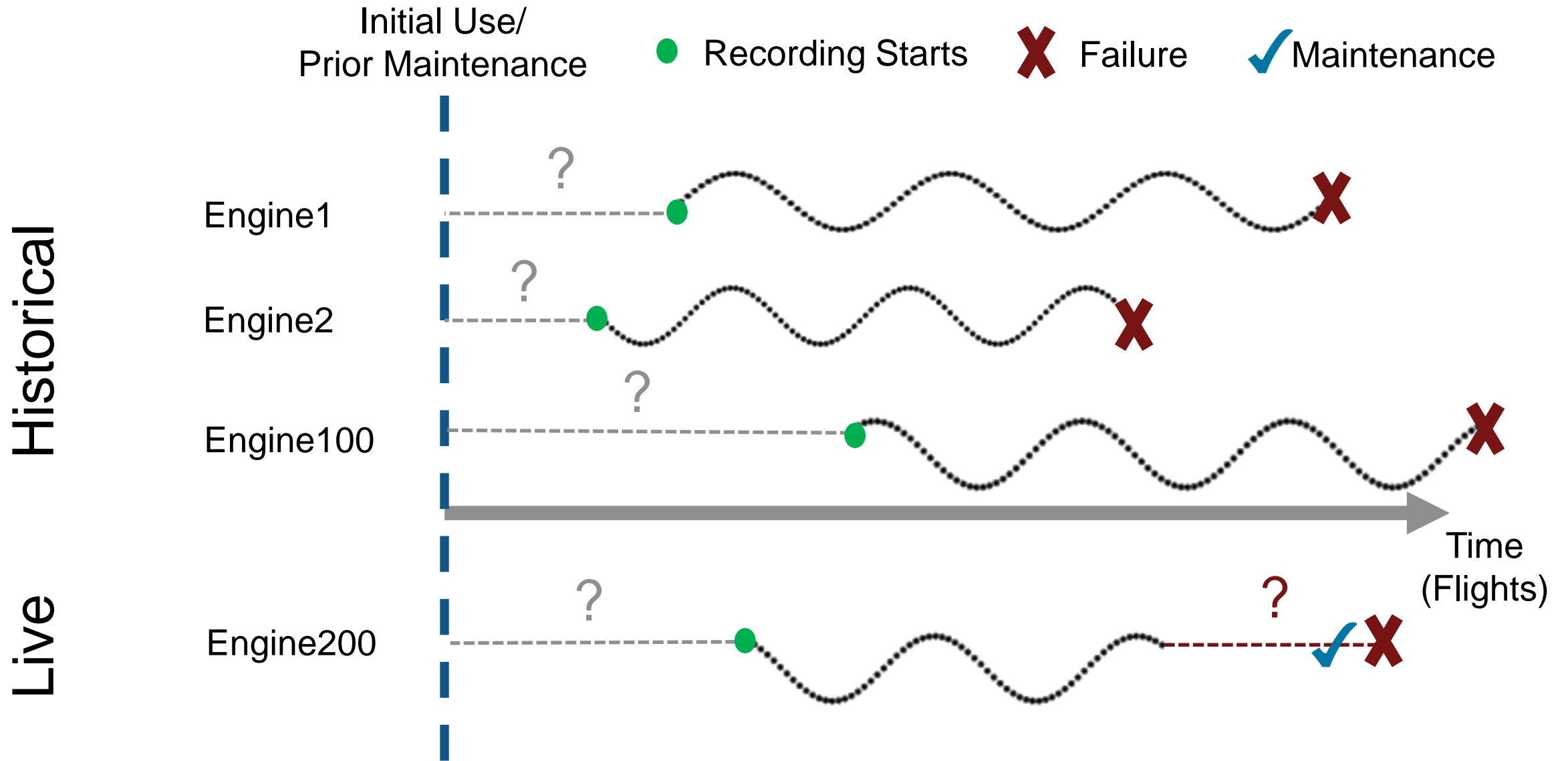
- Performing scheduled maintenance
- Failures still occurring (maybe by design)
- Search records for when failures occurred and gather data preceding the failure events
- Can we predict how long until failures will occur?

Data provided by NASA PCoE

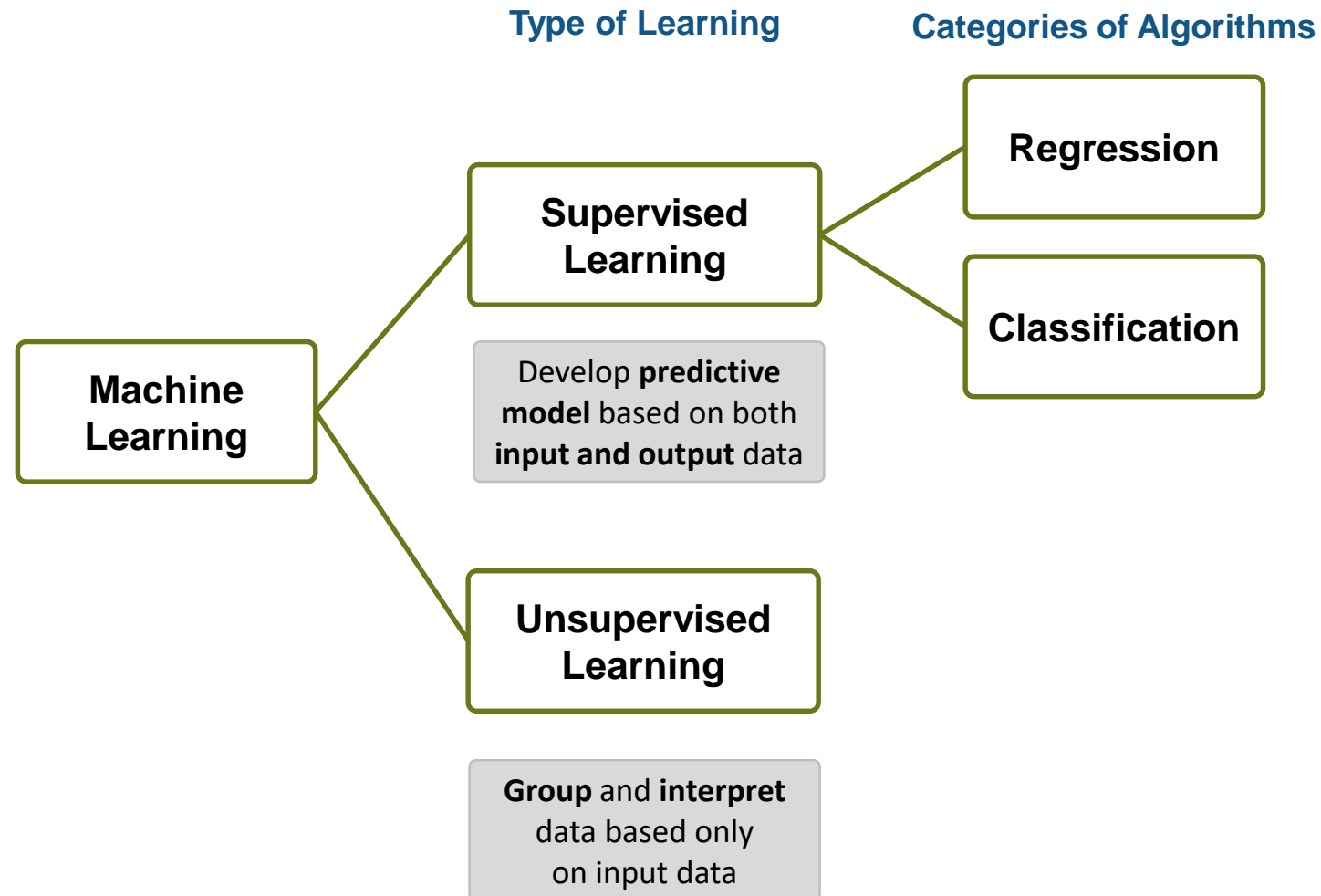
<http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/>



# How Data was Recorded



# Overview – Machine Learning



# Challenges

1. Data – Do you have enough/correct data?

A failure might be a rare occurrence – how do you develop an algorithm if you don't know what a failure looks like



2. How do you find the best possible algorithm?

There are dozens of modeling techniques

3. How do you deploy your algorithm into production?

# Identifying the Best Classifier Using Classification Learner App

The screenshot displays the MATLAB R2017a environment. The left pane shows a file explorer with a folder named 'ClassificationWebinar.m' selected. The main editor window contains the following MATLAB code:

```

63 %% Visualize all sensor data in categories
64 figure('units','normalized','outerposition',[0 0 1 1])
65
66 for ii = 1:9
67     h(ii) = subplot(3,3,ii);
68     scatter(h(ii),fullDataset.Time,fullDataset(:,2+ii),[],fullDataset.TTF, 'filled');
69     title(h(ii),fullDataset.Properties.VariableNames(2+ii))
70     xlabel(h(ii),'Time')
71     set(h(ii), 'CLim', [1 length(catThreshold)+1])
72 end
73
74 %% Rapid preparation for Machine Learning by leveraging App generated codes
75 %% Rapidly iterate through different models within App, e.g. k-nearest
76 %% neighbor, bagged decision tree, etc, and choose the best model for the
77 %% dataset
78
79 %% use the sensorData variable in the Classification Learner App
80 classificationLearner
81
82 %% k-nearest neighbors
83 %% In our case, k-nearest neighbor outperforms other models based on
    
```

The workspace pane at the bottom left shows the following variables:

Name	Value	Size	Bytes	Class
ans	1x1 Pool	1x1		8 paralle...
catThres...	[50 125 200]	1x3	24	double
fullDatas...	20231x17 t...	20231x17	2615137	table
h	1x9 Axes	1x9	0	matlab...
ii	9	1x1	8	double
ordered...	1x4 cell	1x4	490	cell
sensorD...	20231x15 t...	20231x15	2290977	table



# Challenges

1. Data – Do you have enough/correct data?

A failure might be a rare occurrence – how do you develop an algorithm if you don't know what a failure looks like



2. How do you find the best possible algorithm?

There are dozens of modeling techniques

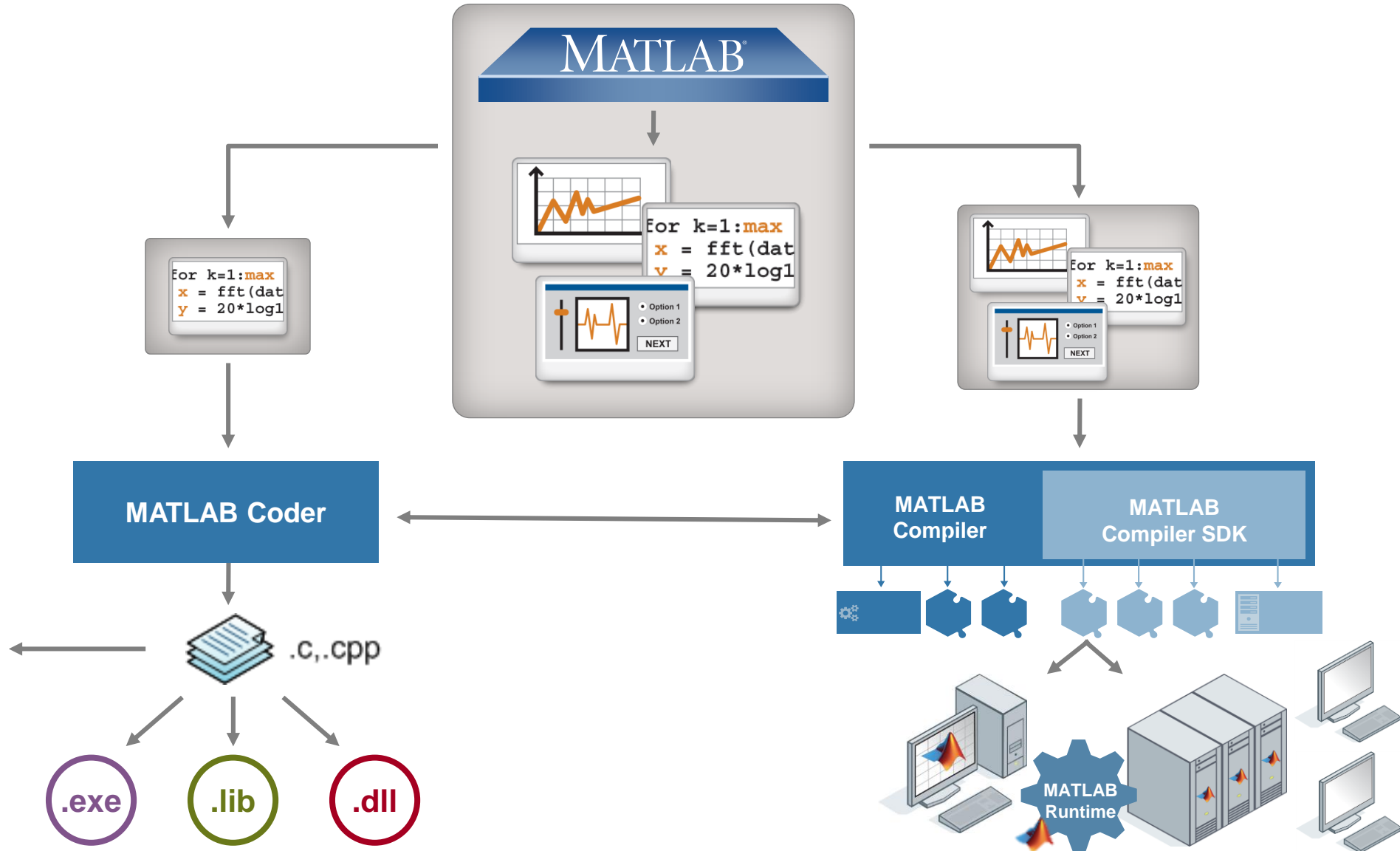


3. How do you deploy your algorithm into production?

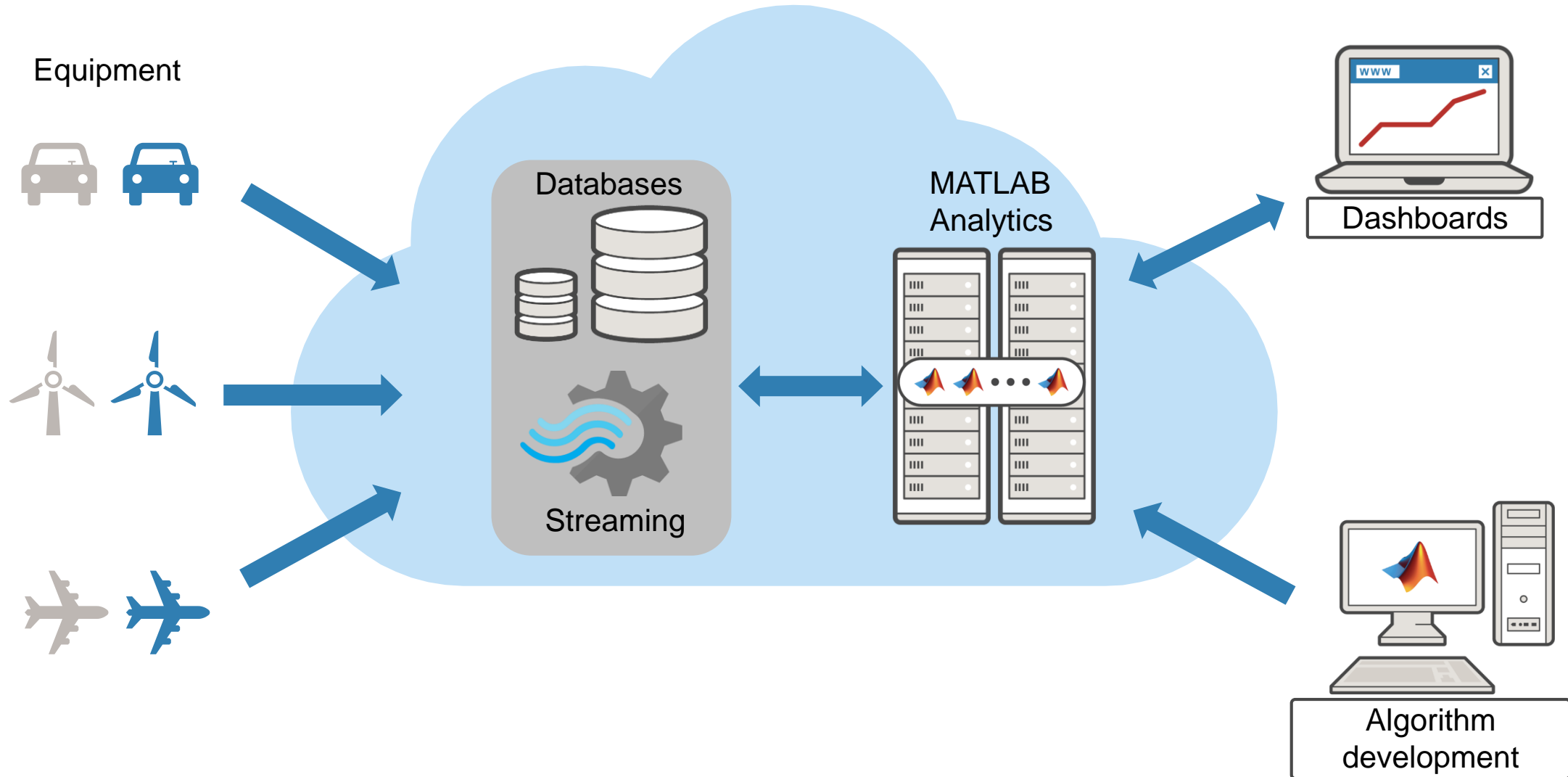
Manually translating MATLAB into other languages can be error prone, and building a production-quality back-end from scratch is expensive

# Integrate Analytics with Your Enterprise Systems

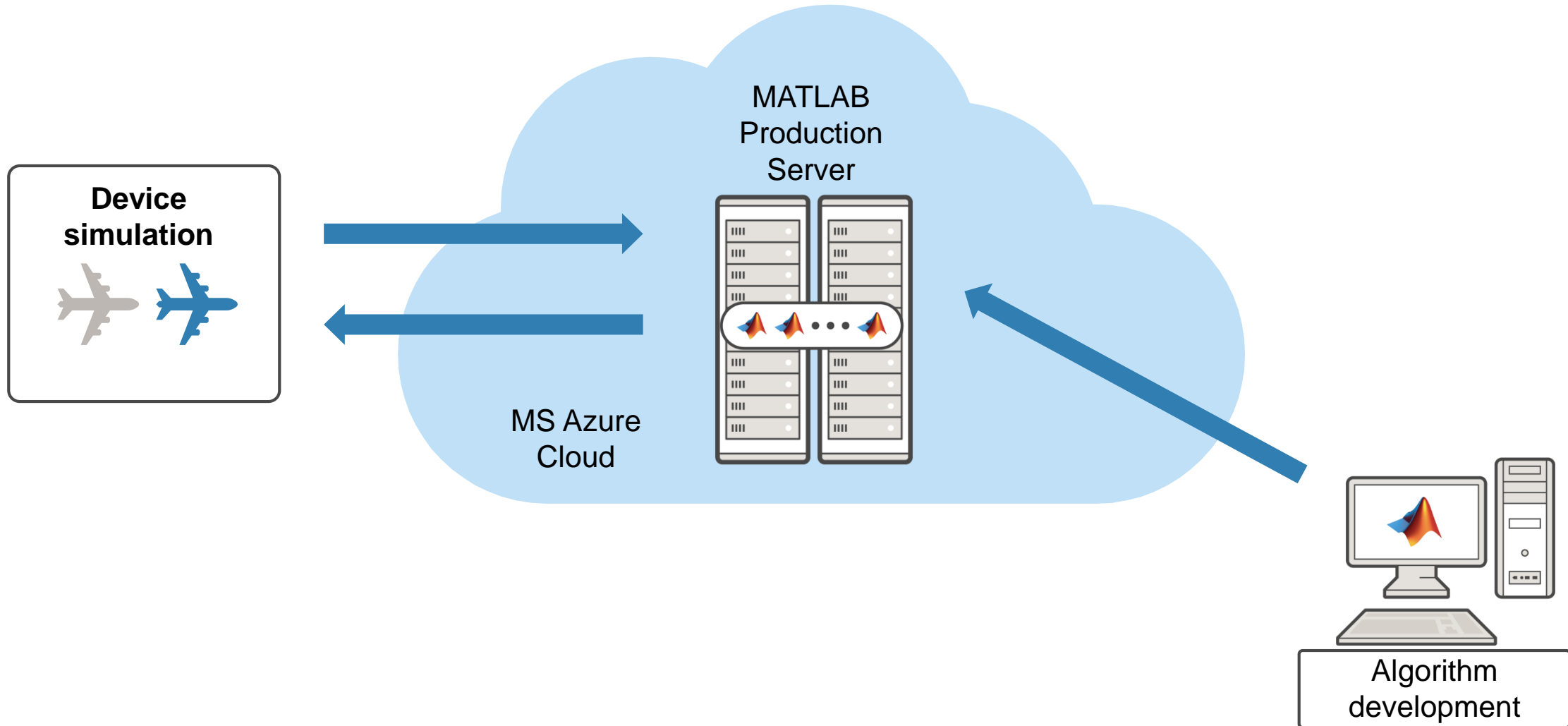
*MATLAB Compiler and MATLAB Coder*



# Building a MATLAB-based service



# Demo: Predictive Maintenance Analytics in the Cloud



# Performing RUL Classification in the Cloud

The image shows the MATLAB R2017a environment. The main window displays a script named `estimateRUL.m` with the following code:

```

1 function RULestimation = estimateRUL(datain)
2     %#function ClassificationKNN
3
4     datain = jsondecode(datain);
5
6     model = load('knnclassifier.mat');
7
8     numdata = struct2table(datain);
9
10    RULestimation = model.KNNClassifier5c.predict(numdata);
11
12    RULestimation = jsonencode(RULestimation);

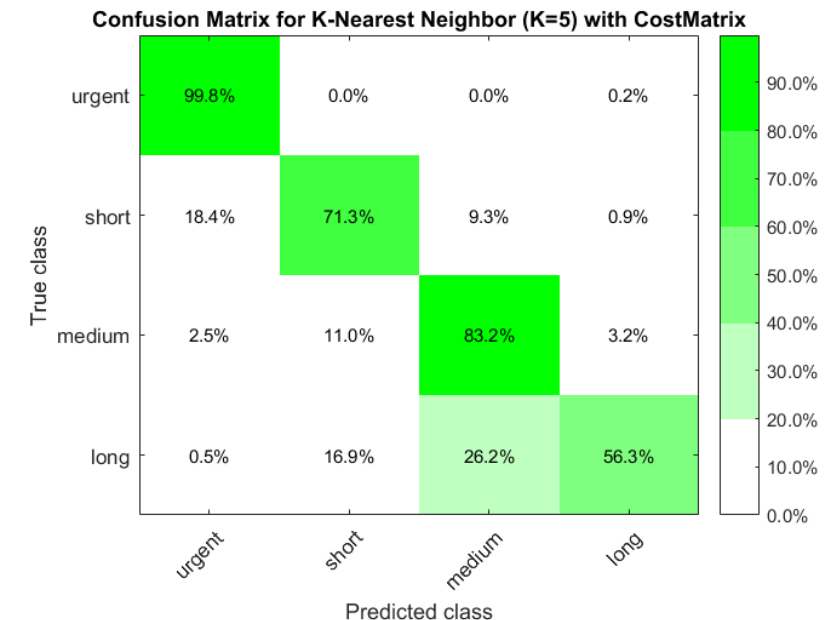
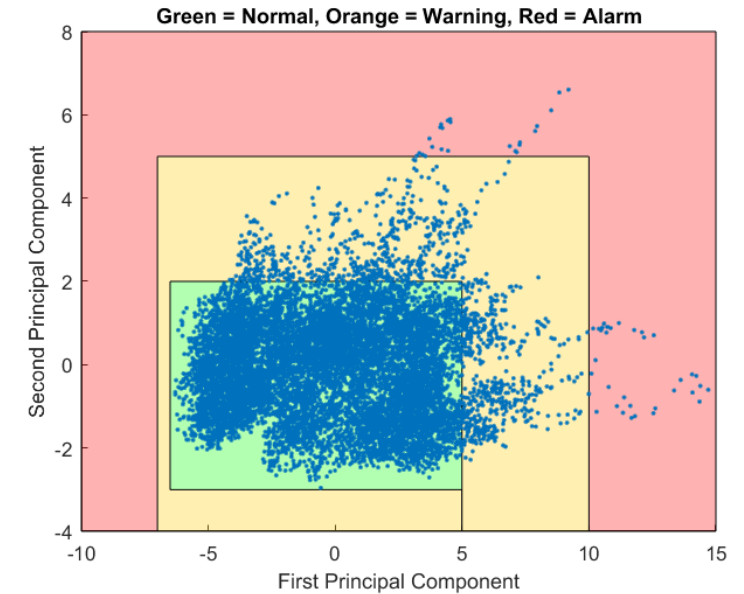
```

The workspace window shows the following variables:

Name	Value	Size	Bytes	Class
ans	1x1 Pool	1x1		8 paralle...
catThres...	[50 125 200]	1x3		24 double 2
data	1x1 struct	1x1		1138 struct
datain	'[["LPCOut...	1x389		778 char
estclass	1x1 cell	1x1		124 cell
fullDatas...	20231x17 t...	20231x17	2615137	table
h	1x9 Axes	1x9		0 matlab...
i	200	1x1		8 double 2
ii	9	1x1		8 double 9
numFra...	200	1x1		8 double 2
options	1x1 webopt...	1x1		255 webop...
ordered...	1x4 cell	1x4		490 cell
response	1x1 struct	1x1		860 struct
sensorD...	20231x15 t...	20231x15	2290977	table
trained...	1x1 struct	1x1		2619935 struct
Xtest	2023x14 ta...	2023x14	230728	table
Ytest	2023x1 cat...	2023x1		2515 catego...

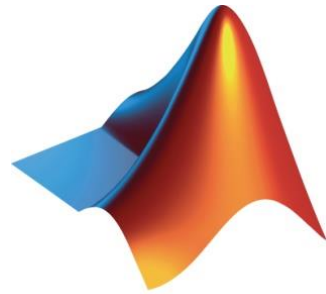
# Key Takeaways

- Frequent maintenance and unexpected failures are a large cost in many industries
- MATLAB enables engineers and data scientists to quickly create, test and implement predictive maintenance programs
- Predictive maintenance
  - Saves money for equipment operators
  - Increases reliability and safety of equipment
  - Creates opportunities for new services that equipment manufacturers can provide



# Predictive Maintenance with MATLAB

- Stop by at the "Data Analytics" -demo station to learn more
- Full webinar at: <http://se.mathworks.com/videos/predictive-maintenance-with-matlab-a-prognostics-case-study-118661.html>



# MathWorks®

*Accelerating the pace of engineering and science*

© 2016 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.