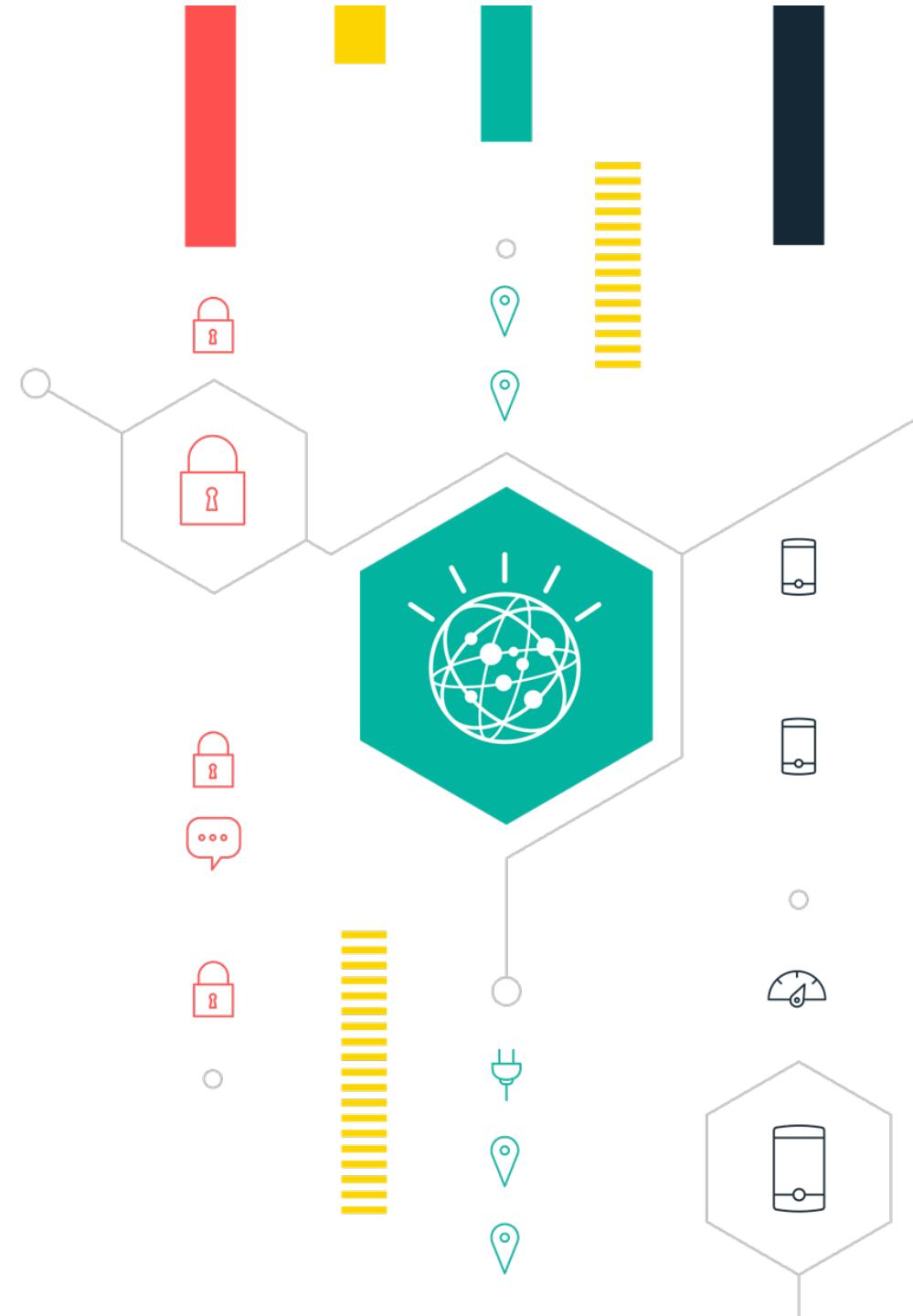


IBM Watson IoT

MAS Predict



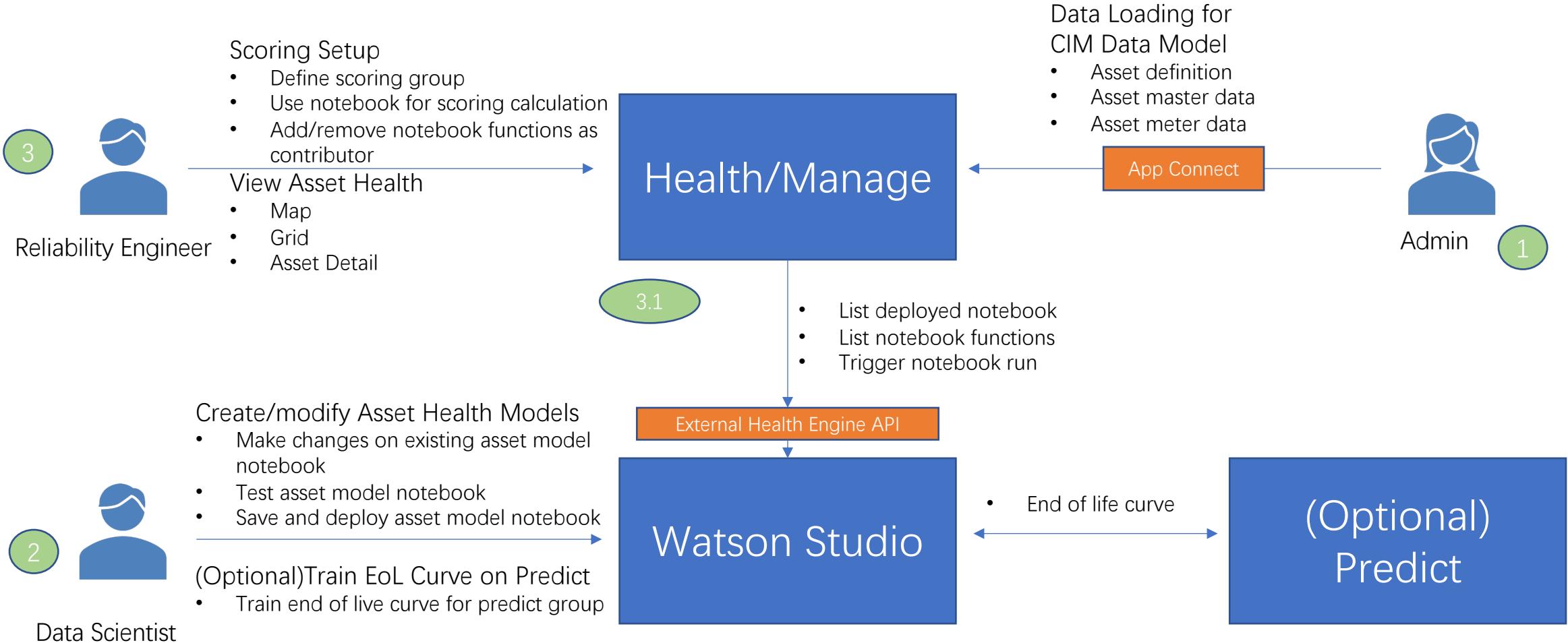
Maximo Health and Predict – Utilities – Basic Definition

■ Brief Definition

- Health and Predict – Solution designed to bring out "Health, Criticality & Risk" scores of the assets. Calculations of scores are derived by using "Manual" and "Analytical" methods. Manual methods give options to create custom definitions and OOB. Analytical methods give options to create prediction scores using various "MATHEMATICAL" models. Additionally, H&P solutions consists of asset investment optimisation. This solution is designed to work asset diagnostics.

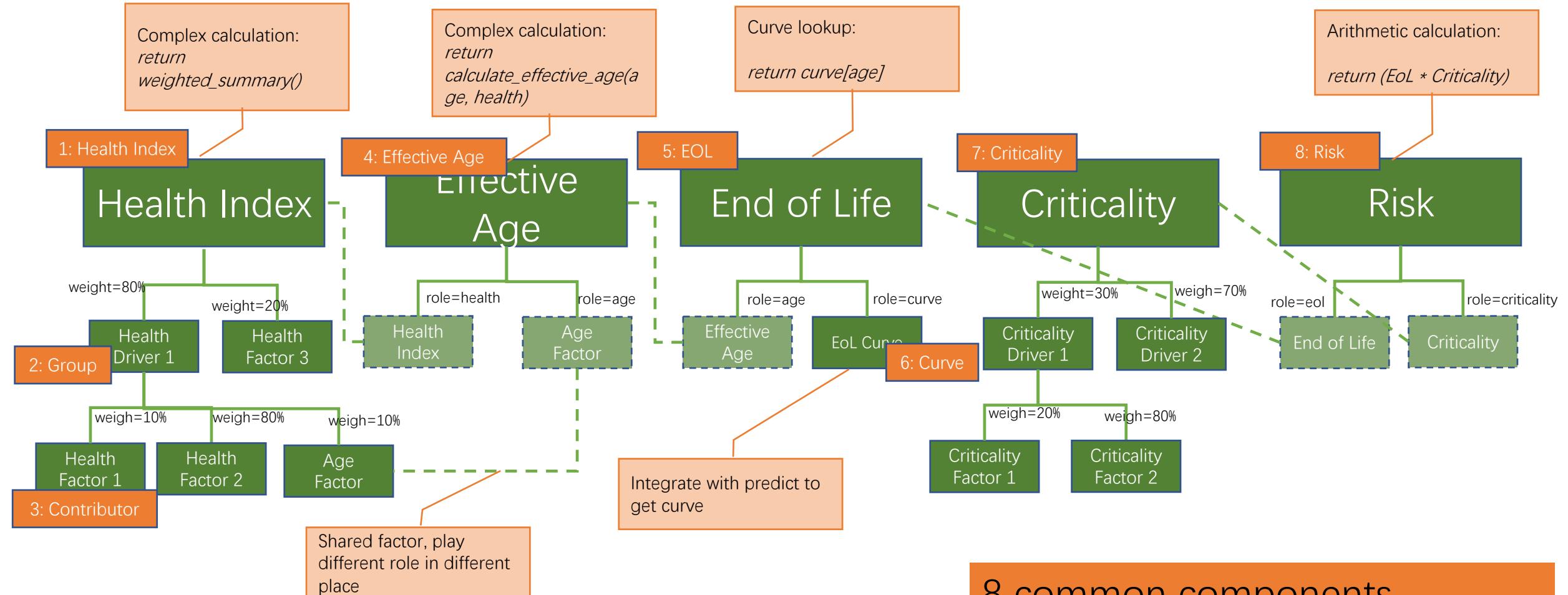
Maximo Application Suite v8.x.0 Health & Predict – Utilities

- External Engine for E&U



Maximo Application Suite v8.x Health & Predict – Utilities

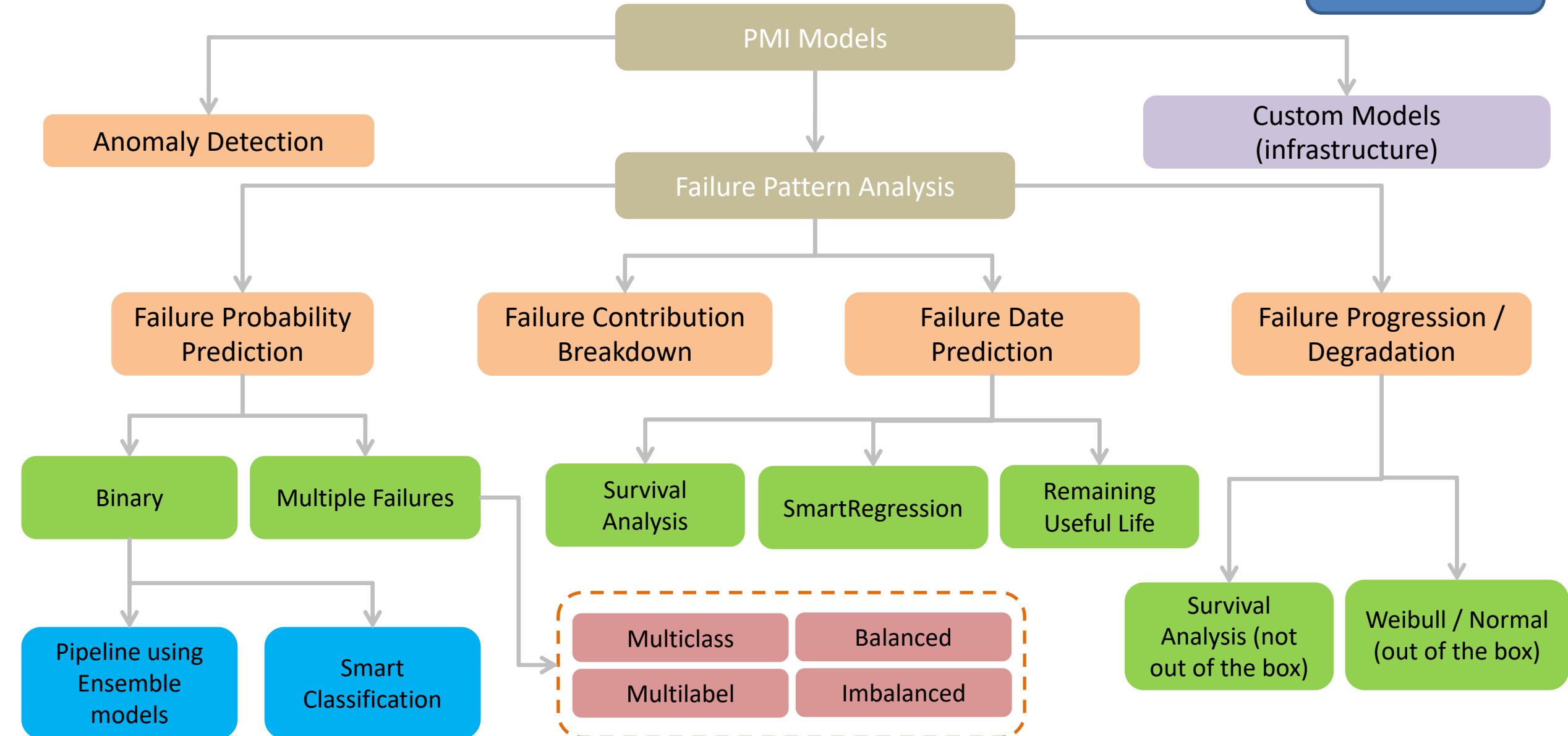
- E&U Health Calculation Methodology



8 common components identified and orchestrate them to support PCTC

Maze of Models (covers WS as well as PMI based templates / models)

DQLearn (separate module)



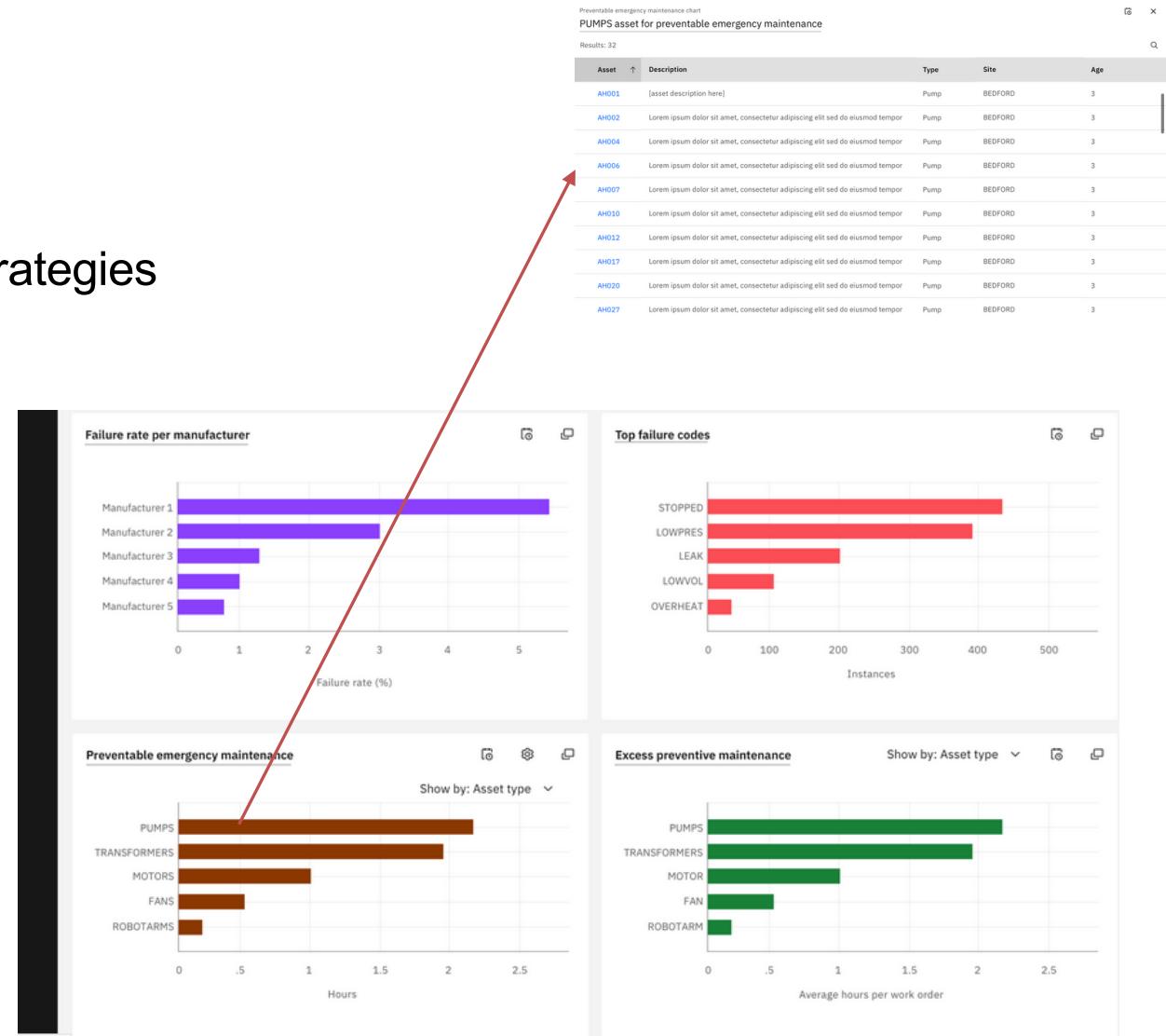
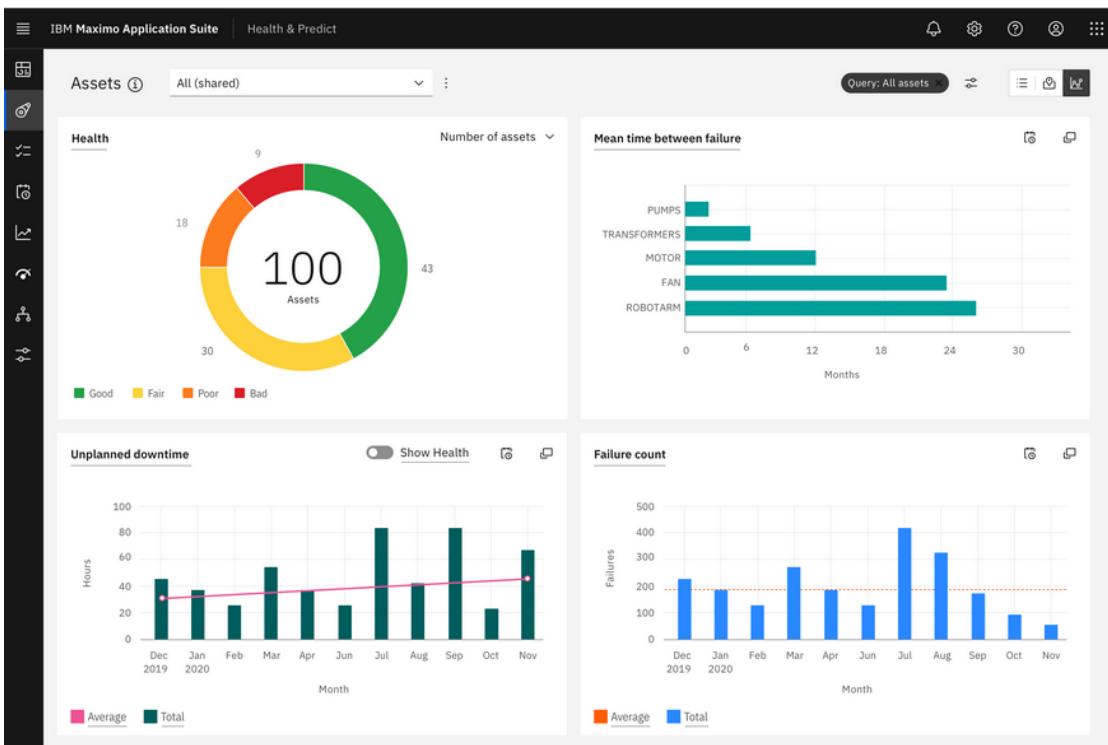
MAS Health & Predict Utilities – Multi-asset analysis

Use case

- Reliability analysis for selected assets
- Drill into charts to see individual assets
- Take action for improvements needed

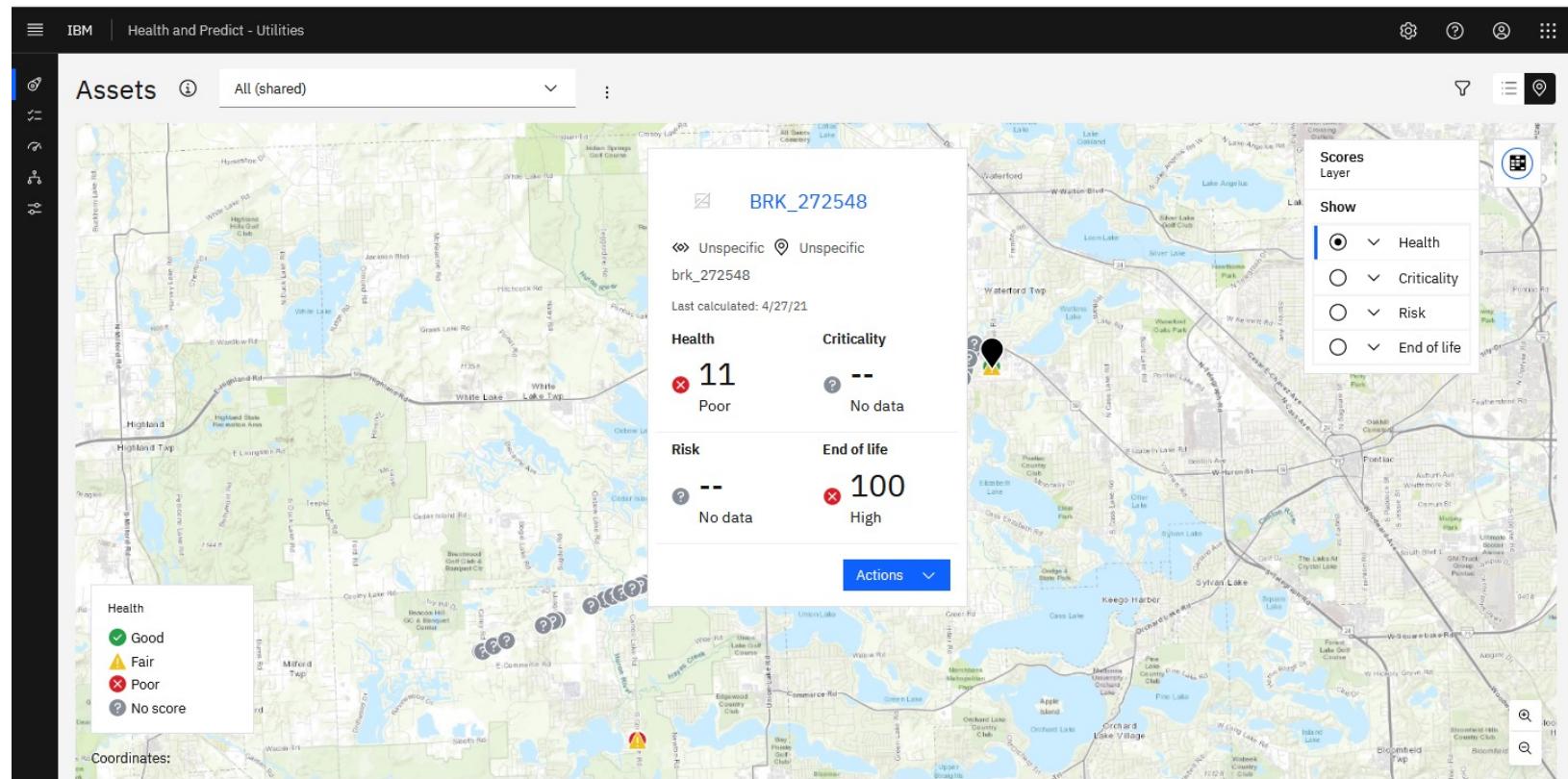
Value

- Creates valuable insights to inform and adjust strategies



MAS 8.x Map view

- Provides full screen geographical view of the performance of assets
- Allows the reliability engineer to see if there are pockets of risk in certain areas
- Includes filtering by score type
- Ability to hover or preview an asset
- Link to asset detail page for more information on asset performance
- Take actions right from the map
- Shares filters and views with the Asset list view
- Configurable to work with any map service
- Location is based on service location as defined in EAM/MAS Manage



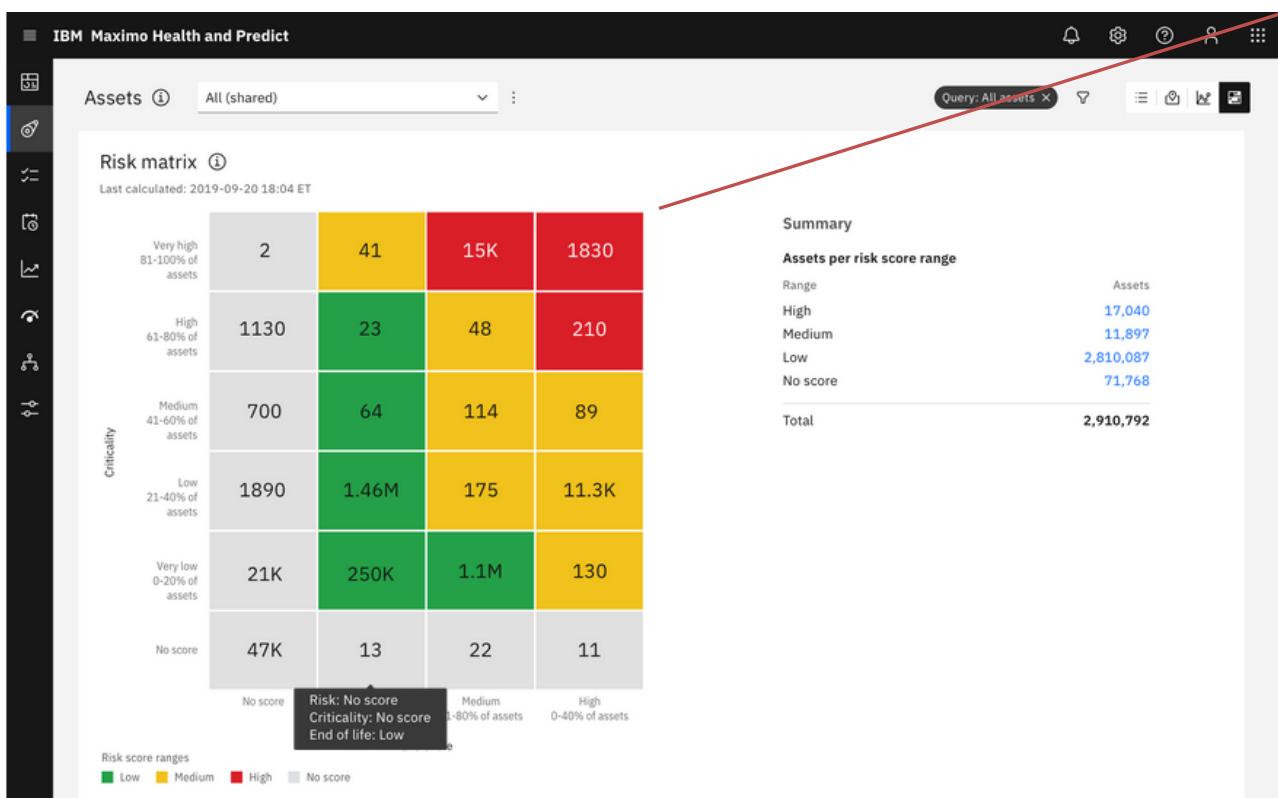
MAS Health & Predict Utilities – Risk Matrix

Use case

- Review risk levels for selected assets
- Drill into assets at certain risk levels
- Take action on individual assets or create replacement project

Value

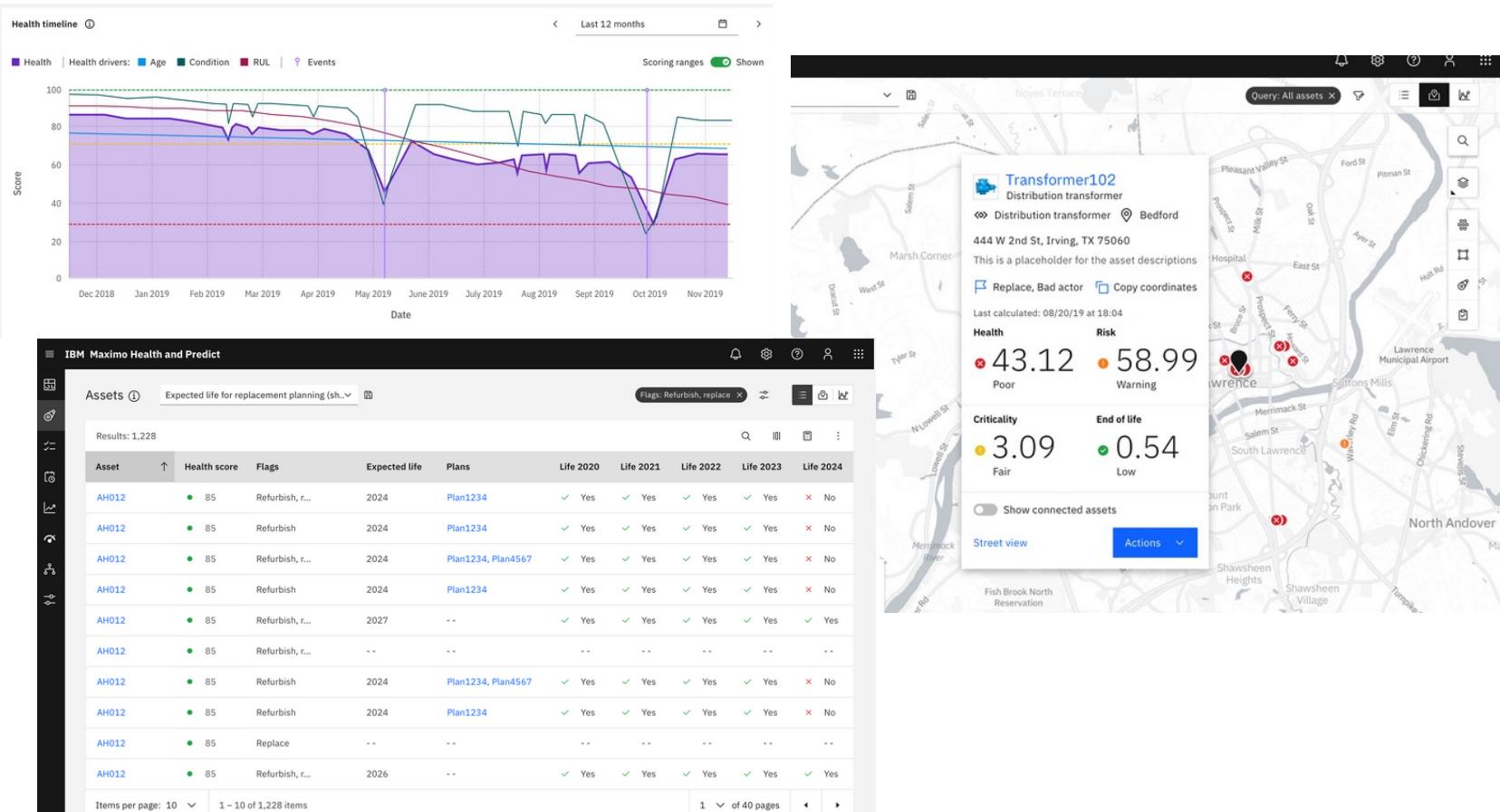
- Easily identify the assets most at risk
- Enable quick action



Asset	Description	Type	Site	Risk score	Criticality	End of life	Health score	Effective age	Age
AH001	[asset description here]	Distribution trans...	BEDFORD	● 45	98	100	● 48	30	3
AH002	Lorem ipsum dolor...	Distribution trans...	BEDFORD	● 35	98	100	● 45	27	3
AH004	Lorem ipsum dolor...	Distribution trans...	BEDFORD	● 41	87	91	● 56	35	3
AH006	Lorem ipsum dolor...	Distribution trans...	BEDFORD	● 23	87	91	● 73	19	3
AH007	Lorem ipsum dolor...	Distribution trans...	BEDFORD	● 11	87	91	● 23	19	3
AH010	Lorem ipsum dolor...	Transmission t...	BEDFORD	● 45	98	93	● 77	19	3
AH012	Lorem ipsum dolor...	Transmission t...	BEDFORD	● 41	100	93	● 91	23	3
AH017	Lorem ipsum dolor...	Transmission t...	BEDFORD	● 23	98	93	● 23	33	3
AH020	Lorem ipsum dolor...	Transmission t...	BEDFORD	● 13	98	88	● 77	31	3
AH027	Lorem ipsum dolor...	Transmission t...	BEDFORD	● 23	98	88	● 77	33	3

Maximo Application Suite v8.x.0 Health & Predict

- **Updates / Enhancements**
 - UX
 - Health history timeline
 - Map view
 - Maintenance history
 - New Capabilities
 - Risk and Criticality Scoring
 - Replacement planning
 - Consumability
 - Scoring enhancements
 - Health & Manage shared infrastructure

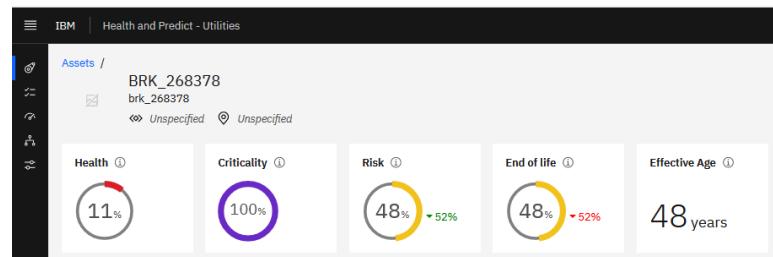


MAS 8.x Utilities Scoring Model extensions

- Provides support for BCTC scoring methodology for Transmission & Distribution Assets
- Leverages pre-defined python scoring models in Watson Studio Notebooks

Includes 2 new score types

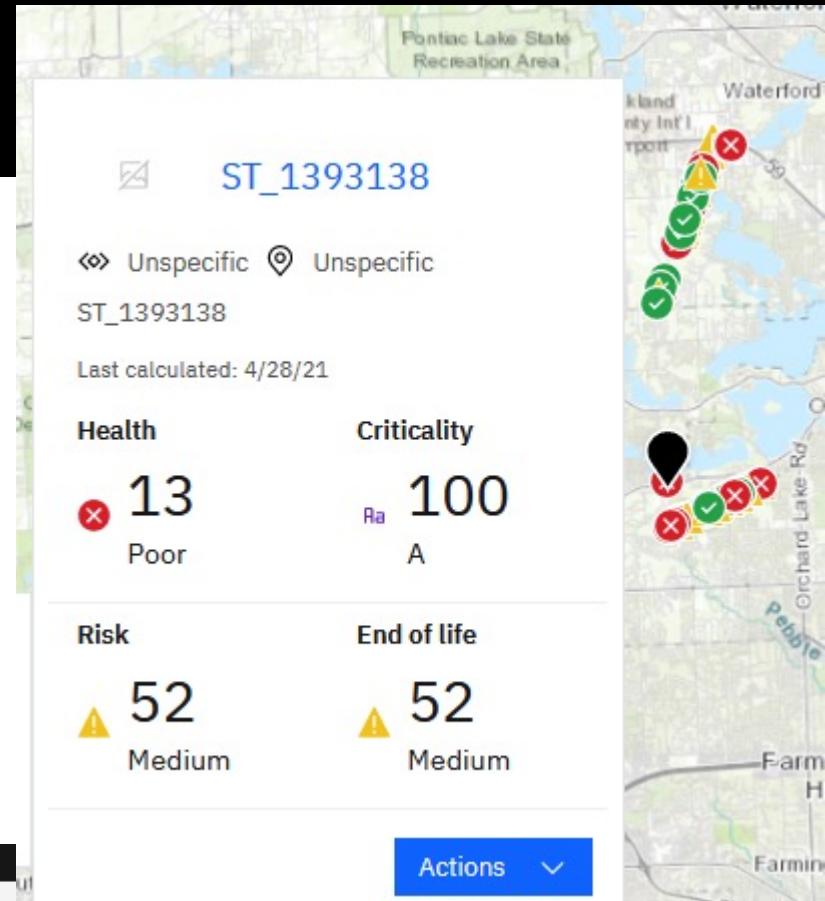
- Effective Age (based on Health)
- End of Life probability based on degradation curve



Enables ability to use one score type within another

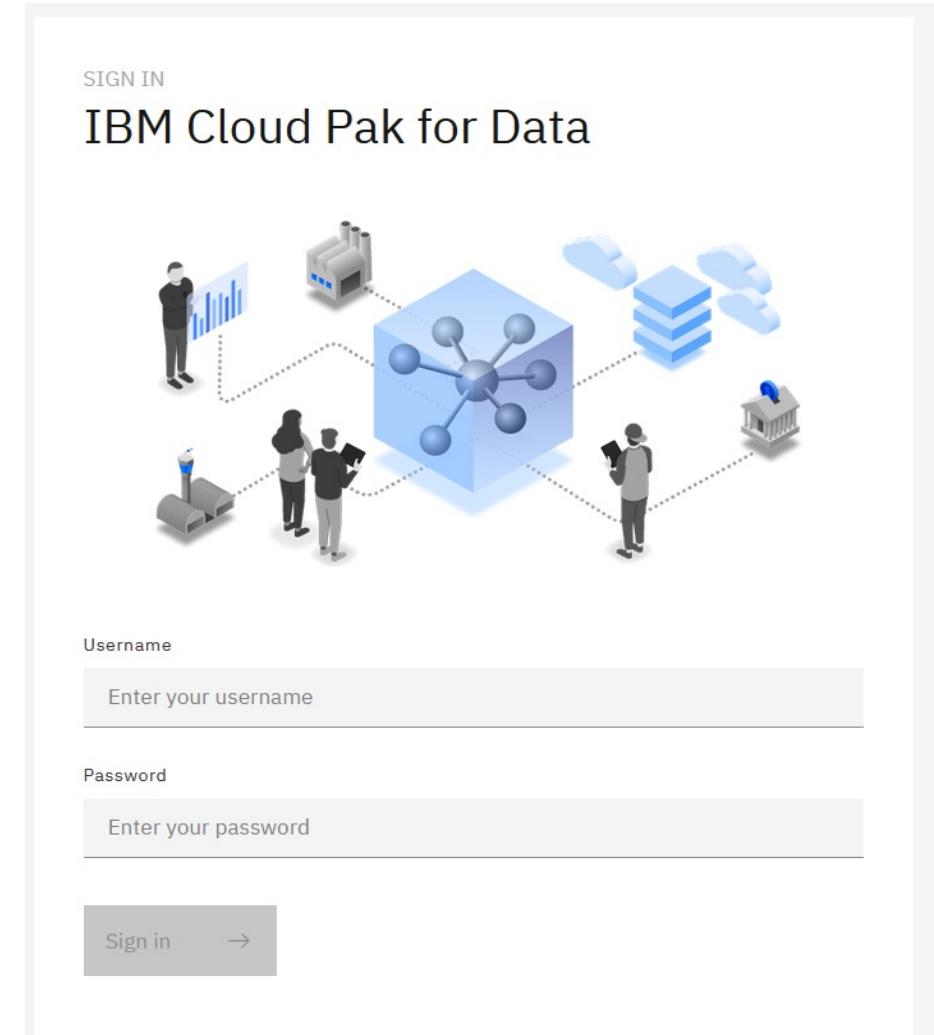
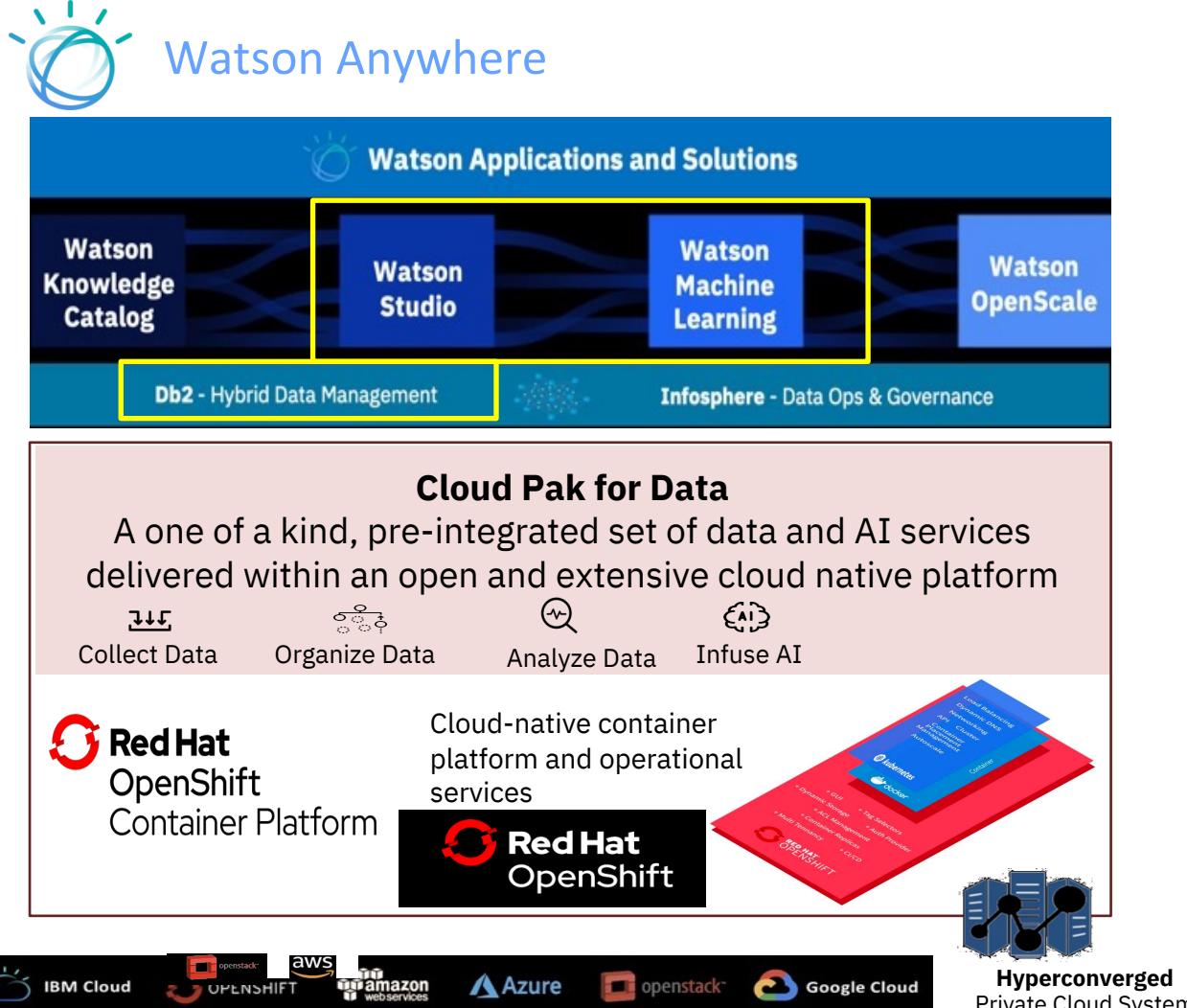
- Effective Age based on Health
- Risk based on Criticality and EOL probability

Type	Description	Active	Score dependencies
Risk	Risk of the asset	Y	Criticality, End of life
Criticality	Criticality of the asset	Y	
End of life	End of Life of the asset	Y	Effective age
Effective age	Effective Age of the asset	Y	Health
Health	Weighted health calculation	Y	



CP4D Update – Introduction

The new release uses CP4D – an integrated platform for data and AI



Key Differences to note on CP4D

- DB2 Storage
- Watson Studio
 - Similar programming model, but internal storage and project lib to access it (as opposed to ICOS)
 - Notebooks updated for portability
- Watson Machine Learning
 - WML client API V4 (the only supported version)
 - Deployment Spaces
- New way of obtaining credentials

```
import os
execution_env = os.environ.get('AX_EXT_DEPLOYMENT_TYPE', 'wscloud') #'icp alternatively cloud.
store_artifacts = False
bucket_name_for_pickled_models = 'maximo-predict-pickled-models' # Keep in mind that neither
#atters in the name, or only those allowed (hyphen or dash is allowed by ICOS)
local_data_folder = 'project_data/data_asset'
deployment_space = 'MAS-Testing-Deployment-Space'
```

The screenshot shows the IBM Cloud Pak for Data web interface. The top navigation bar includes links for 'Most Visited', 'Getting Started', and 'Welcome to watson...'. The main menu on the left has sections like 'Home', 'Projects', 'Connections', 'My instances', 'Collect', 'Organize', and 'Analyze'. The 'Analyze' section is currently active, with its sub-section 'Analytics deployments' highlighted with a yellow box. On the right, there's a sidebar for 'Environments' and a 'Filter navigation' bar. The central area displays a 'Data' tab with 'Files' selected, showing a list of files including CSVs and a ZIP file related to 'trainbrake' data. A blue arrow points from the 'Deployment Spaces' bullet point in the list above to the 'Analytics deployments' section in the interface.

What do I need to know to get things going?

- Credentials:
 - Need to use the UI to get the right values for APM_ID, APM_API_BASEURL, APM_API_KEY

In [105]:

```
%capture  
# @hidden_cell  
%env APM_ID=*****  
%env APM_API_BASEURL=*****  
%env APM_API_KEY=*****
```

In []:

```
!pip install --trusted-host ${TRUST_PREDICT} -U ${APM_API_BASEURL}/ibm/pmi/service/rest/ds/${APM_ID}/${APM_API_KEY}/lib/download?filename=pmlib-1.0.0.tar.gz
```

Downloading the notebooks:

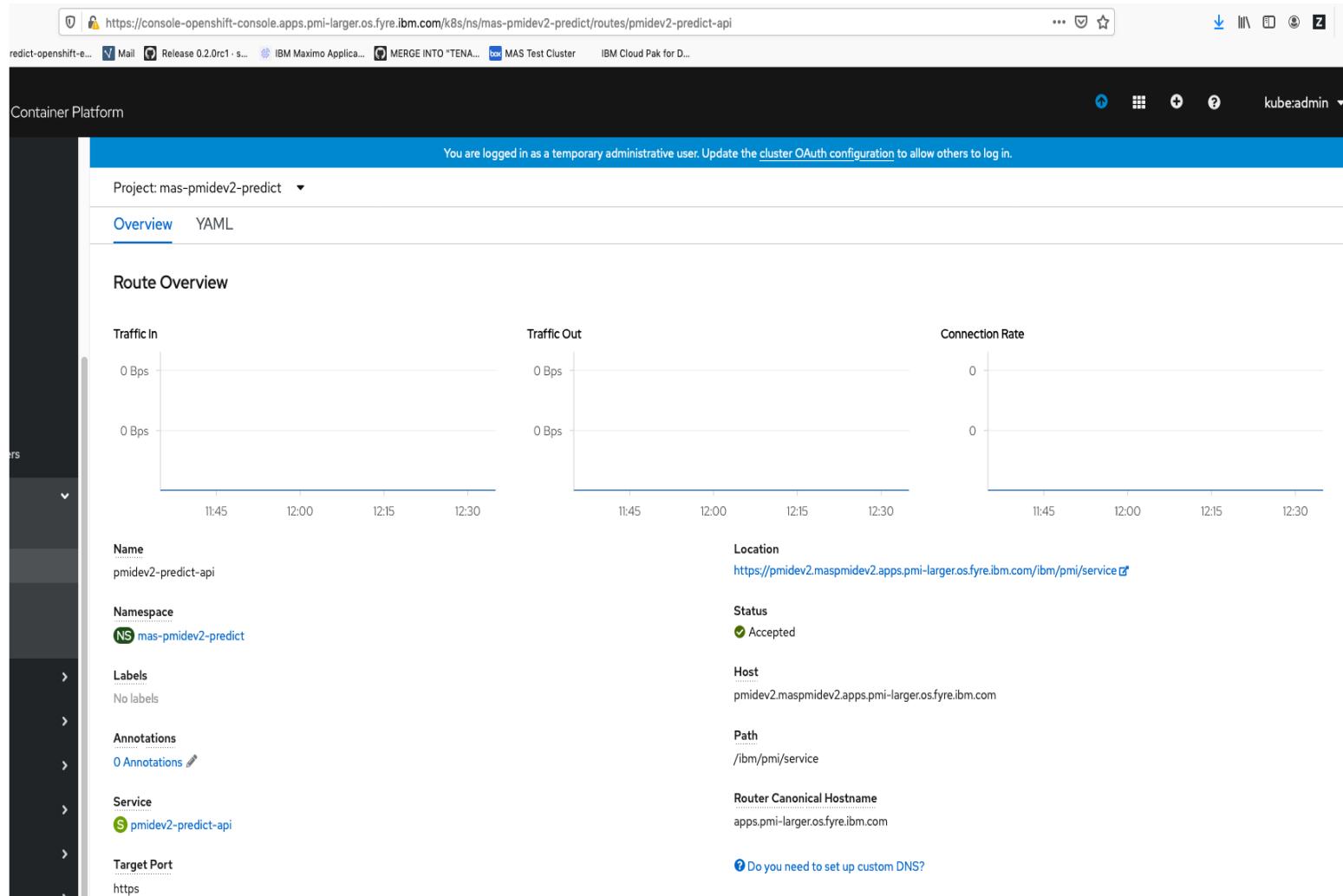
https://Maximo_Predict_PROD_URL/ibm/pmi/service/rest/ds/APM_ID/APM_API_KEY/file/download?ver=1.0

Knowledge center: [https://www-](https://www-03.preprod.ibm.com/support/knowledgecenter/SSLPYA_current/com.ibm.pre.doc/admin/t_using_notebooks.html)

[03.preprod.ibm.com/support/knowledgecenter/SSLPYA_current/com.ibm.pre.doc/admin/t_using_notebooks.html](https://www-03.preprod.ibm.com/support/knowledgecenter/SSLPYA_current/com.ibm.pre.doc/admin/t_using_notebooks.html)

What do I need to know to get things going - To get the MAXIMO_PREDICT_PROD URL:

- login in openshift cluster
Networking-->Routes
- Login to Openshift Cluster as kubeadmin
- From the left Menu, click on Projects
- Search and select mas-<your_instance>-predict
- From the Left Menu, click on routes
- Copy the 2nd URL from the list
`https://.<your_instance>.<mas_domain>/ibm/pmi/service`



The screenshot shows the Container Platform interface for a route named "pmidev2-predict-api". The top navigation bar includes links for "redact-openshift-e...", "Mail", "Release 0.2.0rc1 · s...", "IBM Maximo Applica...", "MERGE INTO 'TENA...', MAS Test Cluster", and "IBM Cloud Pak for D...". The user is logged in as "kube:admin". The main content area displays "Route Overview" with three graphs: Traffic In (0 Bps), Traffic Out (0 Bps), and Connection Rate (0). Below the graphs, detailed route information is listed:

Name	pmidev2-predict-api
Namespace	mas-pmidev2-predict
Labels	No labels
Annotations	0 Annotations
Service	pmidev2-predict-api
Target Port	https
Location	https://pmidev2.maspdmidev2.apps.pmi-larger.os.fyre.ibm.com/ibm/pmi/service
Status	Accepted
Host	pmidev2.maspdmidev2.apps.pmi-larger.os.fyre.ibm.com
Path	/ibm/pmi/service
Router Canonical Hostname	apps.pmi-larger.os.fyre.ibm.com

A note at the bottom right says "Do you need to set up custom DNS?".

Place to get the credentials

The screenshot shows the IBM Maximo Application Suite Predict interface. On the left, there is a navigation sidebar with the following items:

- Assets
- Work queues
- Scoring
- Predict grouping
- Application administration

The "Application administration" item is highlighted with a blue border. The main area of the screen displays a table with the following columns: Description, Group ID, Query, and Object. The table contains five rows of data:

Description	Group ID	Query	Object
	1018	FDT1	ASSET
	1023	pump_query	ASSET
	1017	FD1	ASSET
	1021	IVT3456-stayaway	ASSET
jhweeks	1022	IVT3456-stayaway	ASSET

At the bottom of the table, it says "of 5 items". On the right side of the table, there are navigation controls: a dropdown menu set to "1", a page indicator "1 of 1 pages", and arrows for navigating between pages.

Place to get the credentials – System Properties

System Properties jenn03

Select Action Global Properties Filter > 1 - 8 of 1002

<input type="checkbox"/>	Property Name	Description	Current Value	
<input type="checkbox"/>	PMBBISTRACKED	Setting global variable for BB to track visitors.	1	
<input type="checkbox"/>	RESThelp.url	Work Center REST Documentation URL	https://ibm.box.com/s/uep6etdgz2lg6sa5ej8d5e3b2cr418fs	
<input type="checkbox"/>	WAS.AdminPassword	Websphere Admin Security Password.		
<input type="checkbox"/>	WAS.AdminUserName	Websphere Admin Security Username.		
<input type="checkbox"/>	WAS.LTAURL	URL for launching the Log and Trace Analyzer	/ibm/action/launch?pageID=com.ibm.ac.lta.web.ui.LogAnalyzer&showNavArea=false	
<input type="checkbox"/>	alertmessages	Show messages in title bar	false	
<input type="checkbox"/>	apptBookMgrWindowSize	System property to that sets the window size	7	
<input type="checkbox"/>	bim.import.dir	Root directory for BIM working directory	c:/bim/import	
<input type="checkbox"/>	Select Records			

New Row

Instance Properties Filter > 1 - 2 of 2

Property Name	Description	Value	
mxe.com.port	Com port		
mxe.rmi.port	Port used by RMI for communication; if left at 0, RMI will use a random available port on the system	0	

Place to get the credentials – APM_ID

The screenshot shows the IBM Maximo Application Platform interface for managing system properties. The top navigation bar includes links for All Files, predict-openshift-e..., Mail, Release 0.2.0rc1, IBM Maximo Application, MERGE INTO "TEN... MAS Test Cluster, and IBM Cloud Pak for D... The user is logged in as jenn03.

System Properties

Global Properties

Property Name	Description	Current Value
mxe.PMIId	Instance identifier to the PMI server	06b2f320

Instance Properties

Property Name	Description	Value
mxe.com.port	Com port	
mxe.rmi.port	Port used by RMI for communication; if left at 0, RMI will use a... available port on the system	0

Place to get the credentials – APM_BASE_URL

The screenshot shows the IBM API Connect interface for managing endpoints. The top navigation bar includes a home icon, a menu icon, and the title "End Points". Below the navigation is a toolbar with "Query", "Find End Point", "Select Action", and various icons for creating, deleting, and editing endpoints.

The main area displays the "End Point" details for "PREDICTAPI". The endpoint is identified as "Predict rest end point" with "Handler: HTTP" and "Consumed By: INTEGRATION".

A table titled "Properties for End Point PREDICTAPI" lists the following properties:

Property	Value	Encrypted Value	Allow Override?
CONNECTTIMEOUT			<input type="checkbox"/>
COOKIES			<input type="checkbox"/>
HEADERS	content-type=plain/text		<input checked="" type="checkbox"/>
HTTPEXIT	com.ibm.tivoli.maximo.assethealth.pre		<input type="checkbox"/>
HTTPMETHOD			<input checked="" type="checkbox"/>
PASSWORD			<input type="checkbox"/>
READTIMEOUT			<input type="checkbox"/>
URL	https://ivt-predict-api.mas-ivt-predict.s		<input checked="" type="checkbox"/>
USERNAME			<input type="checkbox"/>

A blue "Test" button is located at the bottom right of the table.

Place to get the credentials: APM_KEY

IBM Maximo Asset Management Administration

Integration Message Templates API Keys IoT Connector

API keys enable users to access and interact with this Maximo system by using REST API calls.

Search Results: 9 User Add API key >

CISCO001 API key: xxxxxxxxxxxxxx Copy key	CISCO002 API key: xxxxxxxxxxxxxx Copy key	CISCO003 API key: xxxxxxxxxxxxxx Copy key	DATAFLOW API key: xxxxxxxxxxxxxx Copy key	JENN00 API key: xxxxxxxxxxxxxx Copy key	JENN03 API key: a1234b1234c1234d1 Copy key
MAXADMIN API key: xxxxxxxxxxxxxx Copy key	PMIDEV API key: xxxxxxxxxxxxxx Copy key	SYSADMIN API key: xxxxxxxxxxxxxx Copy key			

UI flows and insights

RE Dashboard

IBM Maximo Predict

No results

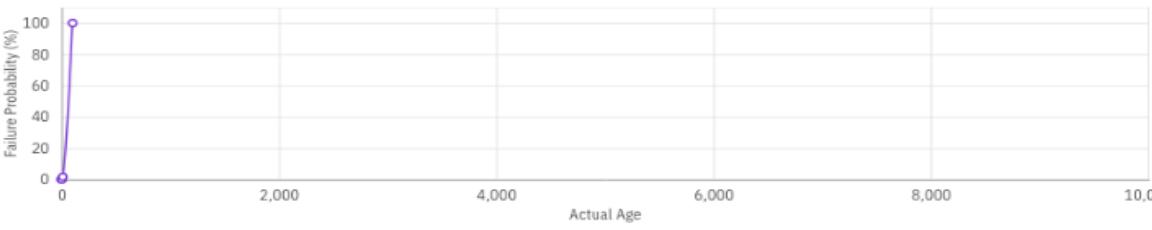
Unspecified
Installation date
01/19/2015

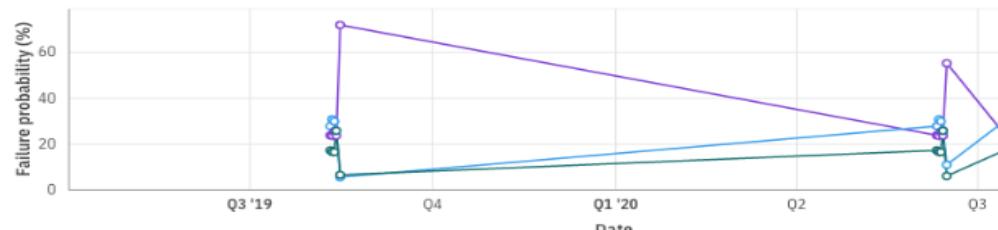
Predictions

Anomaly detection ⓘ
Last calculated: 07/15/20 at 20:00
Threshold -0.0
0.1 ⓘ
Last anomaly detected
-0.0 14 Jul 2020

Failure probability ⓘ
Last calculated: 07/15/20 at 20:00
PUMPS/LOWPRES
55 %
Time window 5 days Group average 55%

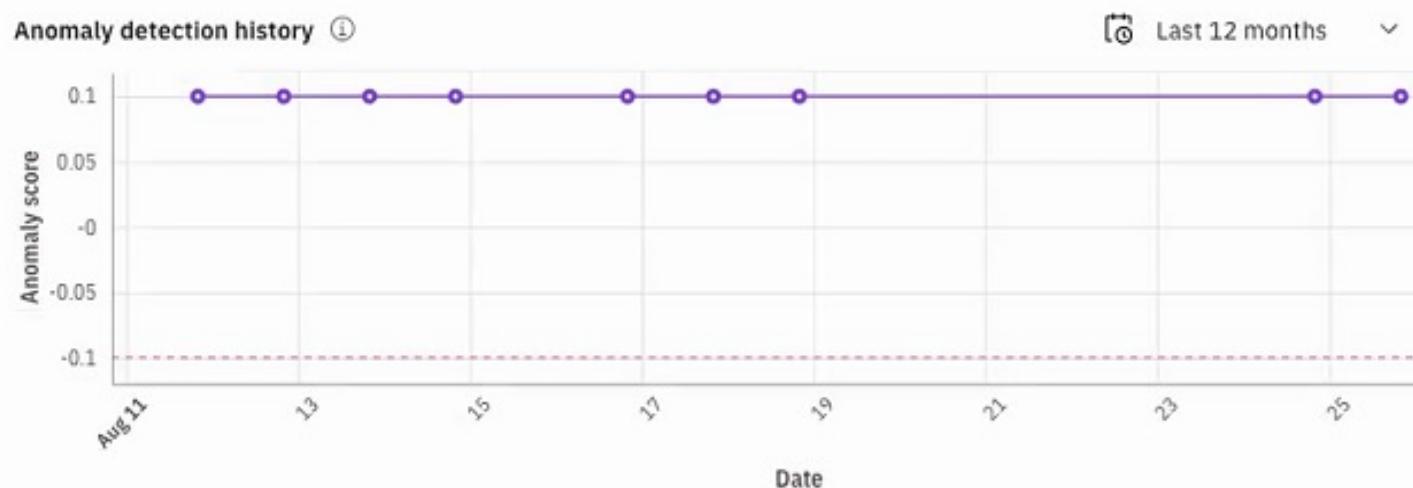
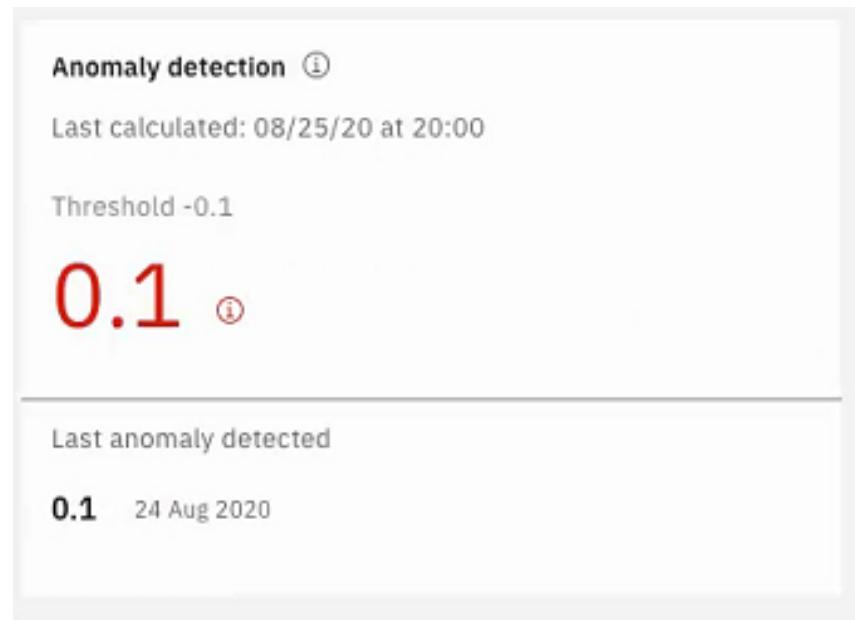
Estimated Time to Failure ⓘ
Last calculated: 07/15/20 at 20:00
PUMPS/LEAK
130 ± 54 days
Date of training data
09 Apr 2015 - 15 Jul 2020

Asset life curve ⓘ
last calculated: 08/20/19 at 18:04


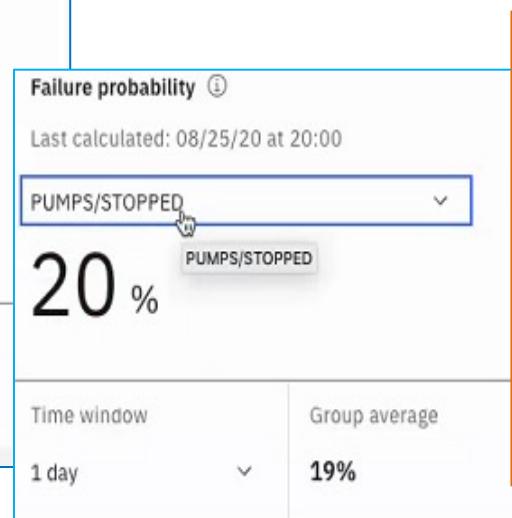
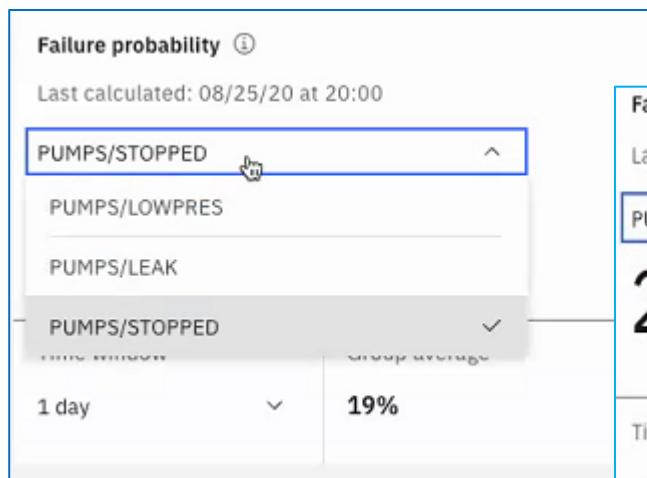
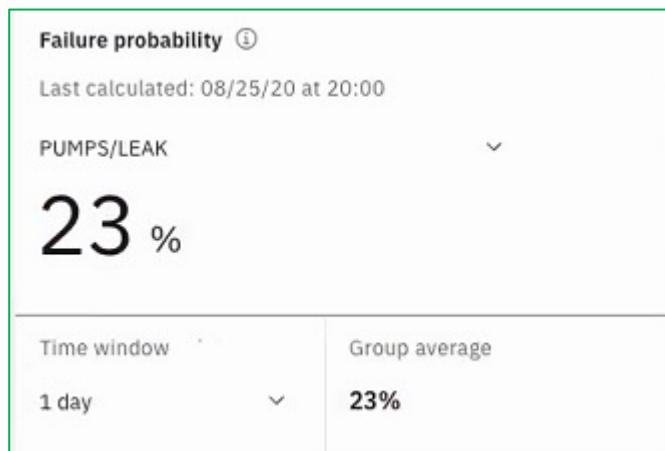
Failure probability trend ⓘ
5 days Last 12 months
PUMPS/LOWPRES PUMPS/LEAK PUMPS/STOPPED


All Files | Powered by predict-openshift-e... Mail Release 0.2.0rc1 · s... https://tenant01.health.maspimidev1.apps.pmi-larger.os.fyre.ibm.com/maximo/oslc/graphite/reengineer/index.html#/assetDetail?linkedDS=overviewasset&ids=2680

Anomaly Detection



Failure Probability Prediction



Predicted Failure Date / Estimated Time to Failure

Estimated time to failure (i)

Last calculated: 08/25/20 at 20:00

PUMPS/STOPPED ▼

85 ± 35 days

Date of training data
09 Apr 2015 - 02 Aug 2017

Estimated time to failure (i)

Last calculated: 08/25/20 at 20:00

PUMPS/STOPPED ▼

PUMPS/LEAK

PUMPS/STOPPED ▼

Date of training data
09 Apr 2015 - 02 Aug 2017

Failure Contribution Breakdown

Factors that contribute to failures ⓘ

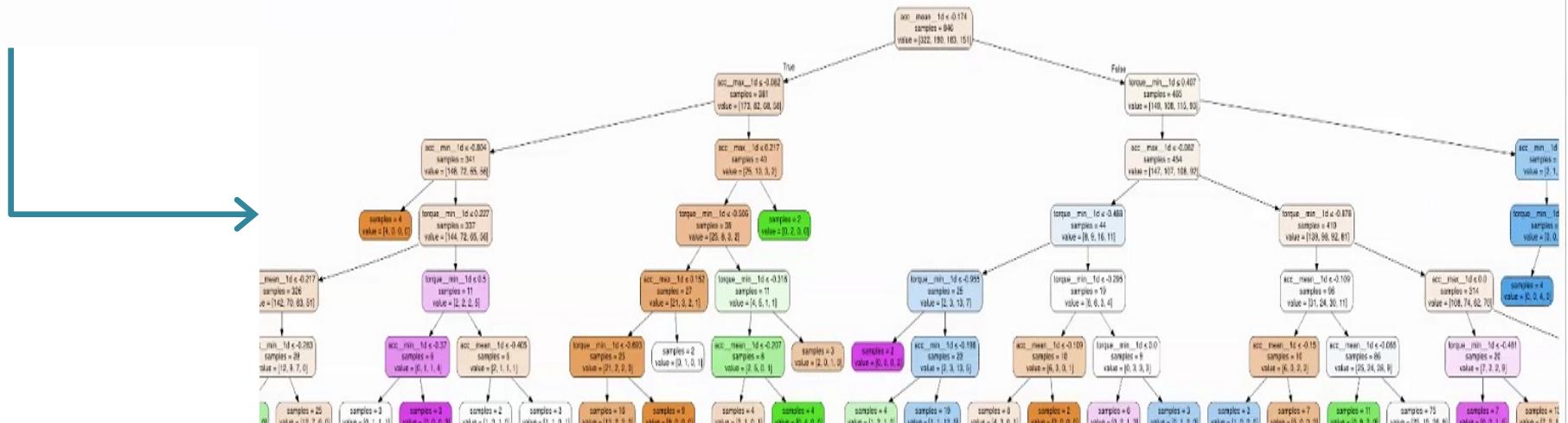
1 day ▾

Factors	Importance	Importance score
torque_min_1d	● High	0.41156
acc_mean_1d	● High	0.31813
acc_max_1d	● Moderate	0.16503
acc_min_1d	● Moderate	0.10528

Full failure analysis tree ⓘ

You can analyze the results of asset factor evaluations to determine which ones are contributing to failures. The evaluation nodes indicate the attributes and the split values they are compared with, the number of training rows that are evaluated, and the number of true and false results.

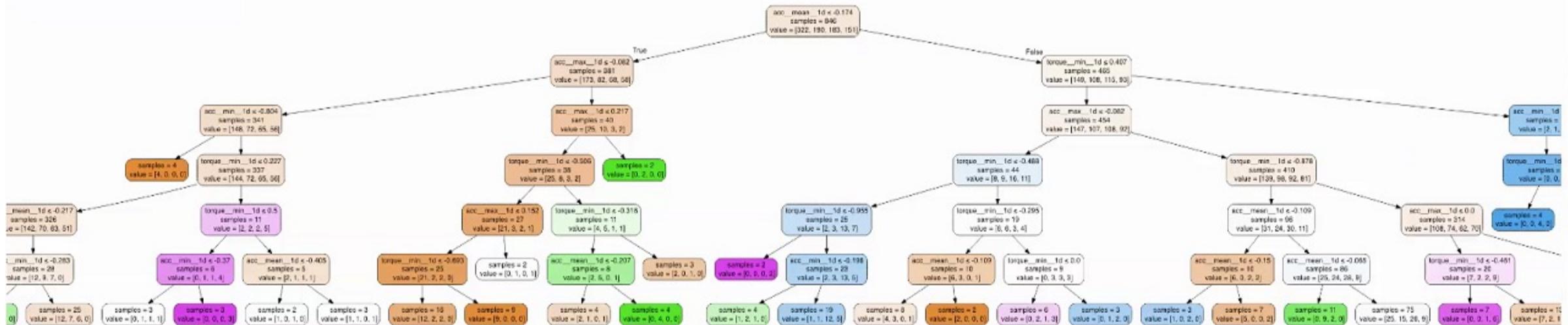
View full analysis tree



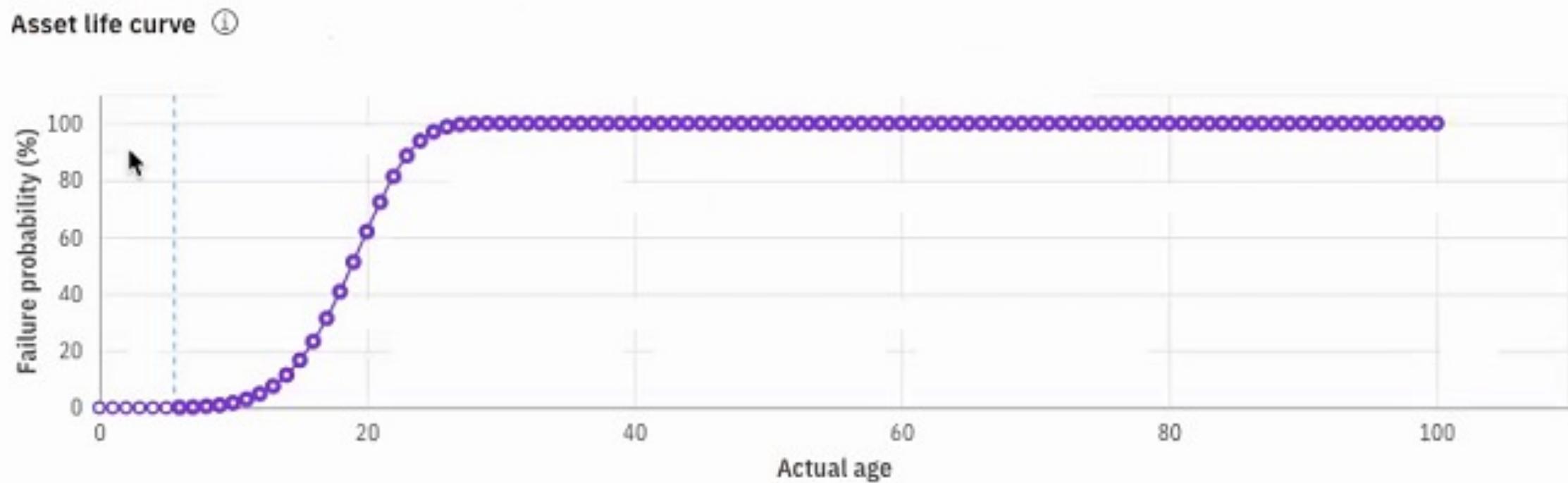
Failure Analysis Tree

Full failure analysis tree

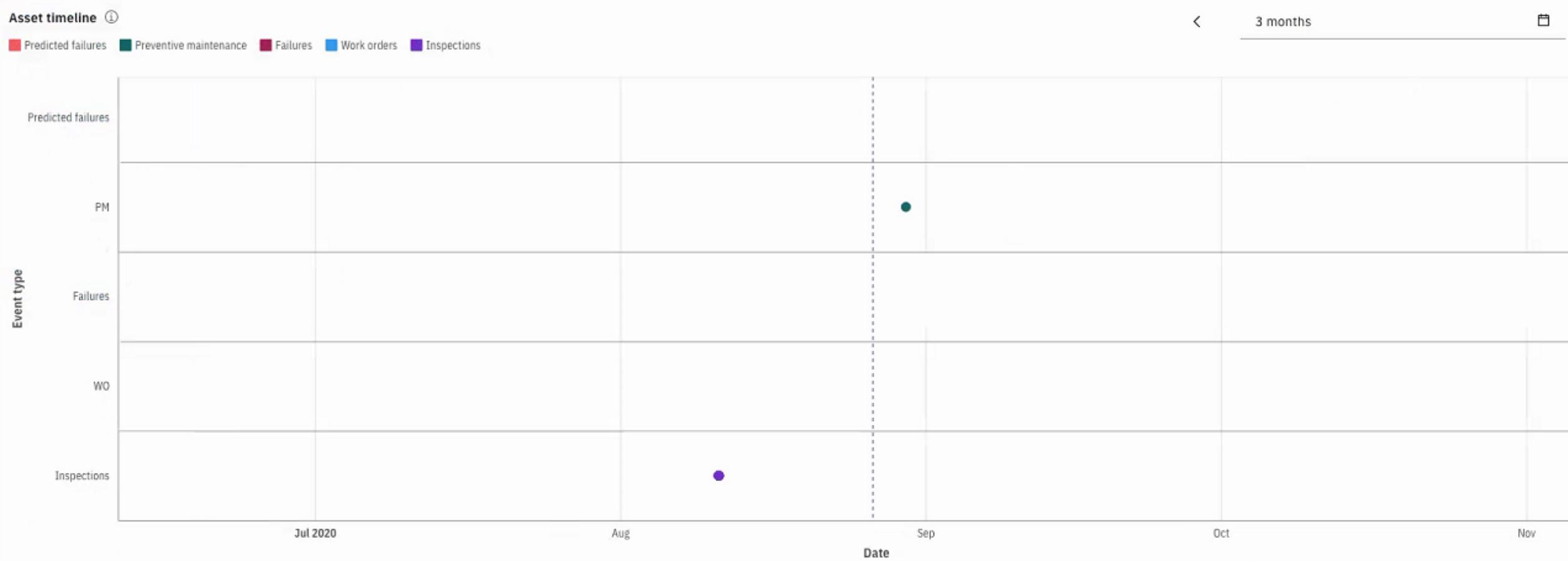
You can analyze the results of asset factor evaluations to determine which ones are contributing to failures. The evaluation nodes indicate the attributes and the split values they are compared with, the number of training rows that are evaluated, and the number of true and false results.



Asset Life Curve



Asset Timeline



Work Queues

Work queues						Refresh queues	
Queue	Queu...	↑	Rema...	Go to	Notes		
High Probability of Failure	1		2	Open first asset	These assets have a High Probability of Failure.		
Failure before PM	1		0	Open first asset	These assets have Failure before PM.		
Low health	1		17	Open first asset	These asset records have health scores that are in the lowest possible health scoring range. Review current asset data, such as t...		
Missing expected life	2		716	Open first asset	These asset records do not have values for expected life. The expected life is typically provided by the asset's manufacturer. To ...		
Missing criticality score	2		696	Open first asset	These asset records do not have criticality scores, or the scores were never successfully calculated. A criticality score is require...		
Missing installation date	2		255	Open first asset	These asset records do not have installation dates. The installation date is required to determine an asset's age. To specify an in...		
Missing replacement cost	2		479	Open first asset	These asset records do not have values for replacement cost. If a replacement cost is not specified, the maintenance-to-replace...		

Work Queue – assets prone to high failure probability

Work queues /

High Probability of Failure ⓘ

Results: 2

Number	Type	Location	Health	↑ Criticality	Installation Date	Expected Life	Replacement Cost	Failure Probability	Failure Date
IVT-3456	ⓘ FLEET		ⓘ 51	3	01/19/2015	10	2,500.00	1.00	09/03/2020 3:48 ...
IVT3456	ⓘ FLEET		ⓘ 52	4	08/10/2017	5	5,000.00	1.00	09/03/2020 3:33 ...

Items per page: 10 ⓘ 1–2 of 2 items 1 ⓘ 1 of 1 pages ⏪ ⏩

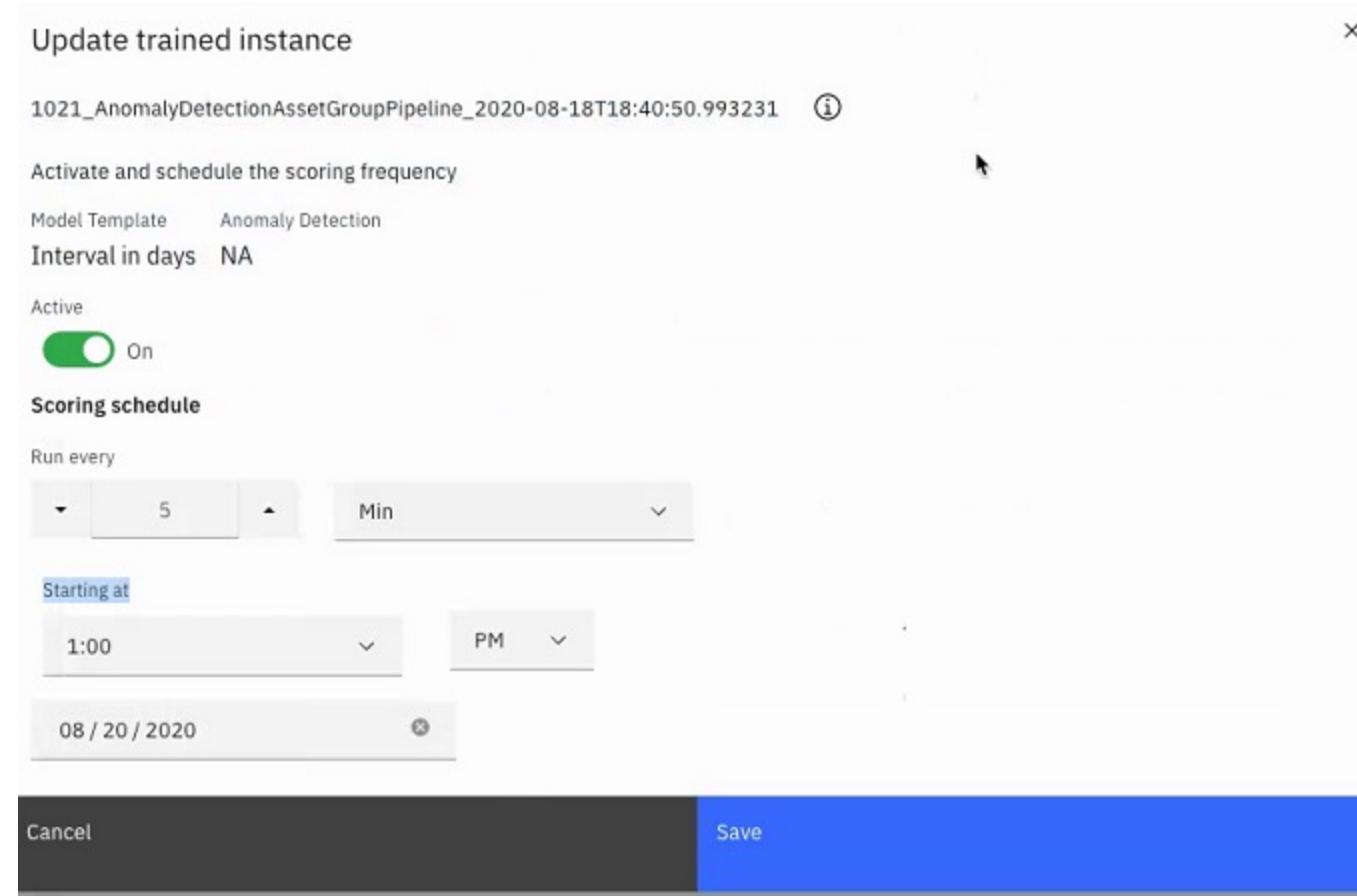
Trained Models

Predict grouping /

test_rvt_fix

Trained instances registered for this group						
Results: 7						
Model template	Trained instance	Trained instance ID	Schedule	Metrics	Active	
Anomaly Detection	1021_AnomalyDetectionAssetGroup	93CCC2CC-EA0D-4ABE-9E32-0F43AB01...	Every 5 Min starting a...		true	
Predicted Failure Date	1021_TimeToFailureAssetGroupPipe	833A8188-5979-4EDF-830F-462E48F3...	Every 5 min starting ...		true	
Predicted Failure Date	1021_TimeToFailureAssetGroupPipe	EA8DA985-7238-4741-92E5-7E035EBF...	Every 5 min starting ...		true	
Predicted Failure Date	1021_TimeToFailureAssetGroupPipe	F01CF6AD-BC6A-4F58-AFB2-2E9D130A...	Every 5 min starting ...		true	
Failure Probability	1021_FailurePredictionAssetGroupP	25D48BD4-3497-4C7A-BE69-355859F4...	Every 5 min starting ...		true	
Failure Probability	1021_FailurePredictionAssetGroupP	863B1F40-031A-4373-8C33-C8838FDA...	Every 5 min starting ...		true	
Asset Life Curve	1021_DegradationCurveAssetGroup	163A5749-361A-43F5-9558-C3F526BD...	Every 5 min starting ...		true	

Trained model scheduling



Predict Grouping

Predict grouping ⓘ					Create group
Group Name	Description	Group ID	Query	Object	
NASA_FDT1		1018	FDT1	ASSET	
test_pump		1023	pump_query	ASSET	
NASA_FD1		1017	FD1	ASSET	
test_rvt_fix		1021	IVT3456-stayaway	ASSET	
jhweeks	jhweeks	1022	IVT3456-stayaway	ASSET	
Items per page:	10 ▾	1-5 of 5 items		1 ▾	1 of 1 pages

Workflow

MAS – Predict: The workflow

- IoT platform (using the data lake) receives and stores the device data (sensor measurements)
- Maximo provides the asset maintenance / failure data
- The PM Library enables special pipelines wrapping around the SROM pipelines to take care of the underlying integration with IoT platform and Maximo

Group Name	Description	Group ID	Query	Object
NASA_F01		1018	F01	ASSET
sec_pmp		1023	pure_sury	ASSET
NASA_F01		1017	F01	ASSET
test_f01		1021	IVT1456-021000	ASSET
PN005	PN005	1022	IVT1456-021000	ASSET
1027		1027	boiler123	ASSET
1028		1028	F01	ASSET

Create Asset group

Device ID	Device Type	Class ID	Date Added
23 results			
device011	new_device_type	Device	Mar 7, 2019 10:13 AM
nbsample01	NotebookSample	Device	Mar 6, 2019 8:42 PM
net01	NewEntityType	Device	Mar 6, 2019 9:16 PM
net201	NewEntityType2	Device	Mar 6, 2019 9:21 PM
net301	NewEntityType3	Device	Mar 6, 2019 9:33 PM
pmifp-asset-10133-device	failurepredictiondevicetype2	Device	Mar 11, 2019 10:18 PM
pmifp-asset-10157-device	failurepredictiondevicetype2	Device	Mar 11, 2019 10:18 PM
pmifp-asset-10913-device	failurepredictiondevicetype2	Device	Mar 11, 2019 10:18 PM
pmifp-asset-11048-device	failurepredictiondevicetype2	Device	Mar 11, 2019 10:18 PM

Register devices on
IoT Platform

The screenshot shows the 'Configure Integration' step of the 'Configure the IoT Service (Step 2 of 5)' process. It displays fields for 'Password', 'User Name', and 'URL' to be filled in for connecting Watson IoT Platform. A note at the bottom states: 'You can choose how to bring data into Maximo Asset Health Insights for analysis by selecting the appropriate method from the dropdown menu. You can also enter the URL of your Watson IoT Platform instance in the URL field. If you connect with Watson IoT Platform, do not modify the HTTP method field for either Watson IoT Platform or Maximo Asset Management.'

Configure asset-device
mapping

MAS Predict

Easy Integration

1 Real Time data from low cost add-on sensors



2 Real time filtered data from automation systems



Accelix™



3 Historical sensor and weather data

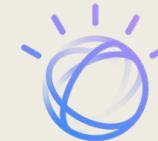


4 Other IoT Platforms

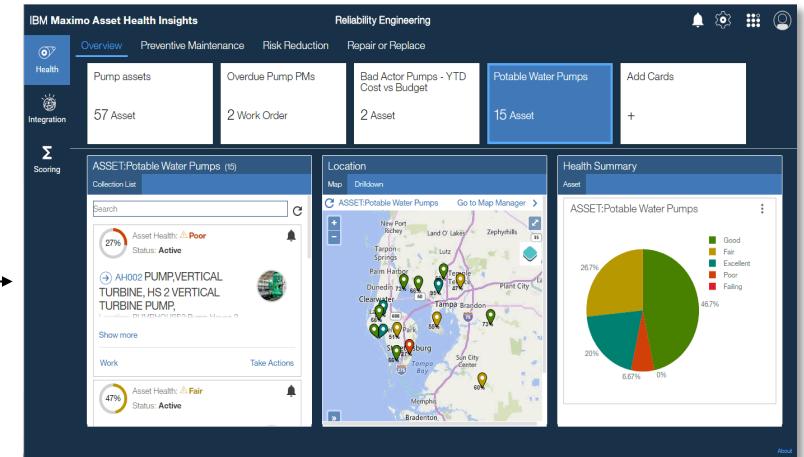


IBM
Watson
IoT Platform

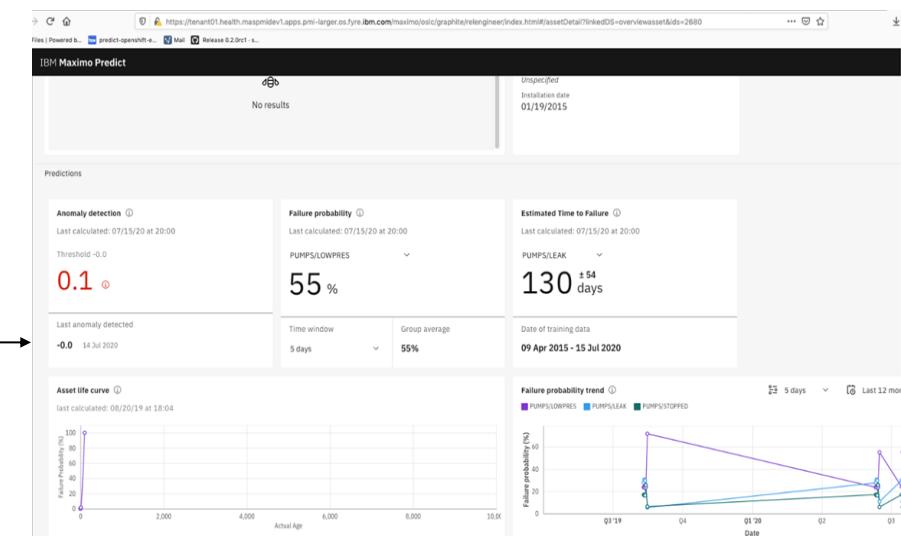
5 Watson Studio & ML



IBM Maximo - Health



IBM Maximo – Predict



Model / Template Specifics

MAS Predict

Train & Register Models using PMI Pipeline on Watson Studio



Connect to Maximo – Predict



Install Maximo Predict SDK



Configure the pipeline



Train the model



Register the model

```
In [1]: %%capture  
# @hidden_cell  
%env APM_ID=08e5ad71  
%env APM_API_BASEURL=https://  
%env APM_API_KEY=*****
```

```
In [2]: # @hidden_cell  
!pip install -U pip==18.1  
!pip install -U git+https://
```

```
In [3]: from pmlib.failure_prediction import FailurePredictionAssetGroupPipeline  
  
group = FailurePredictionAssetGroupPipeline(  
    asset_group_id='1005',  
    model_pipeline={  
        "features": ["SampleFailureSensor:sensor_1", "SampleFailureSensor:  
        "features_for_training": [":faildate"],  
        "predictions": ["failure_probability_15d", "rca_path_15d"],  
        "aggregation_methods": ["mean", "max", "min", "median", "std"],  
        "prediction_window_size": "15d",  
    })
```

```
In [4]: df = group.execute()
```

```
In [5]: group.register(df=df)
```

MAS Predict: Anomaly Detection



Model Description

- Detect anomalies in data
- Leveraged with sparse failure event data



Inputs Required

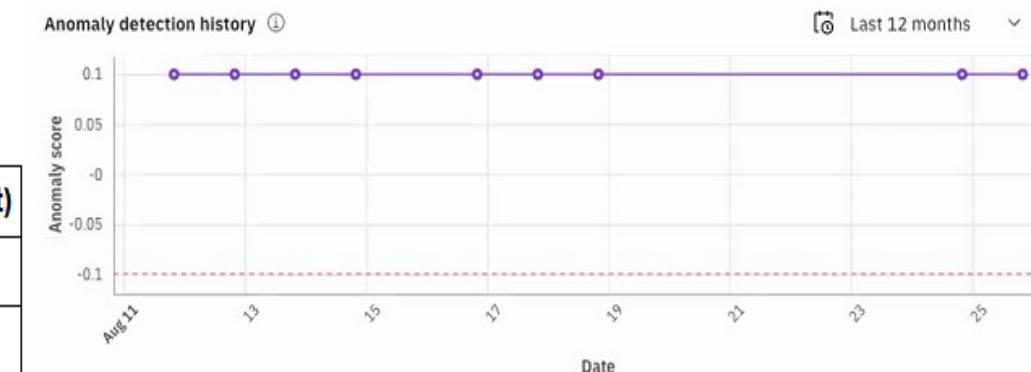
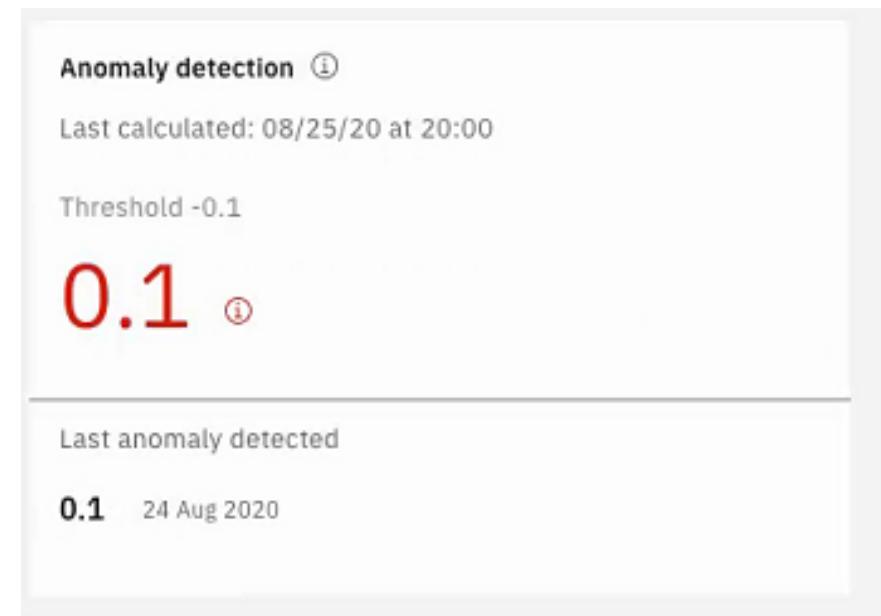
- Timestamped IoT sensor data
- Timestamped failure data
- Validation Data for anomaly scoring

Id (string, optional)	Variable 1 (float64)	Variable 2 (float64)	Variable 3 (float64)Variable n (float64)
P-0983	38	73.54	22.56	39.392
A-3748	98	80.44	-26.384	7.493

Id (string, optional)	Variable 1 (float64)	Variable 2 (float64)	Variable 3 (float64)	Variable n (float64)	Label (int)
P-0983	38	73.54	22.56	39.392	1
A-3748	98	80.44	-26.384	7.493	0

Questions Answered

- What are the assets or entities that show anomalous behavior? (batch level / record level / feature level)



Maximo – Predict Failure Probability Prediction



Model Description

- Predicts imminent failures
- Forecasts Failure Window with probability



Inputs Required

- Timestamped IoT sensor data
- Other process variable as a discrete time variable
- Timestamped failure data



Questions Answered

- What are the assets that have the propensity to fail in the next 'n' days?
What is the probability of failure?

Available in three flavors

Failure Probability Prediction in a given temporal window

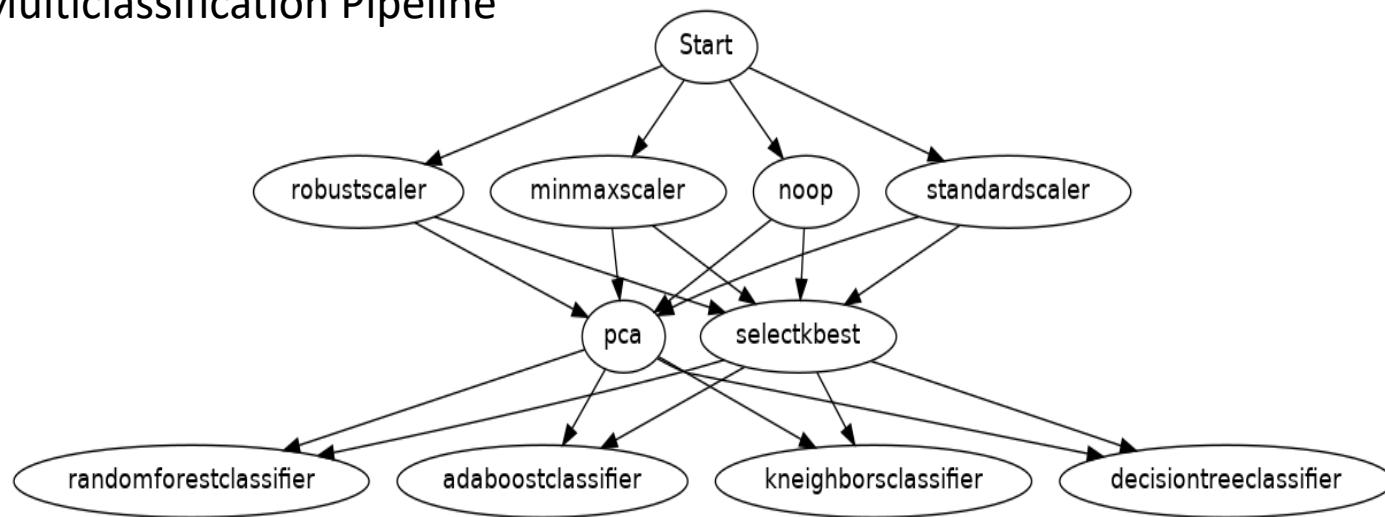
Time to event (where event is failure, or any other adverse condition). Can be used as a measure of RUL (Remaining Useful Life).

Adaptation of Survival Analysis models

Remaining Useful Life – Like how much juice is left before the asset needs to be overhauled / decommissioned / retired

Failure Prediction Pipeline

Multiclassification Pipeline



Input Data

- Timestamped sensor data for the assets, or process variable as a discrete time variable data.
- Timestamped failure data for the assets, or the process

asset id	Timestamp (date with time)	Variable 1 (float64)	Variable 2 (float64)	Variable n (float64)
P-2250	2012-11-14 17:45:21	7	2	3
A-1Y68	2012-08-05 18:32:41	8	2	7

asset id	Timestamp (date with time)	Failure Id (integer)
P-2250	2012-11-14 17:45:21	1
A-1Y68	2012-08-05 18:32:41	0

Metrics supported: ROC AUC, Precision, Recall, F1. Others can be computed easily using scikit learn calls

Smart Classification based Pipeline (applies for binary classification)

```
estimator_set = [('bernoullinb', BernoulliNB()),\
('multinomialnb', MultinomialNB()),\
('decisiontreeclassifier', DecisionTreeClassifier()),\
('extratreesclassifier', ExtraTreesClassifier()),\
('randomforestclassifier', RandomForestClassifier()),\
('gradientboostingclassifier', GradientBoostingClassifier()),\
('kneighborsclassifier', KNeighborsClassifier()),\
('linearsvc', LinearSVC()),\
('logisticregression', LogisticRegression()),\
('xgbclassifier', XGBClassifier()),\
('sgdclassifier', SGDClassifier(loss='log')),\
('svc', SVC(probability=True)),\
('perceptron', Perceptron()),\
('mlpclassifier', MLPClassifier()),\
('passiveaggressiveclassifier', PassiveAggressiveClassifier()),\
('adaboostclassifier', AdaBoostClassifier()),\
('gaussiannb', GaussianNB()),\
('lineardiscriminantanalysis', LinearDiscriminantAnalysis()),\
('quadraticdiscriminantanalysis', QuadraticDiscriminantAnalysis()),\
('gaussianprocessclassifier', GaussianProcessClassifier()),\
('ridgeclassifier', RidgeClassifier()),\
('baggingclassifier', BaggingClassifier()),\
('nusvc', NuSVC())]
```

Binary and Multiclass Failure Probability Prediction

- Binary Classification: Classify into one of the two types of failures (classes)
 - All failures are treated as 1, and “no failure” class is represented by 0
 - SmartClassification or Legacy pipeline
- Multiclass: Classify into one of the multiple / many types of failures (classes) such as Blockage / Leak / Crack etc.
 - Multiclass with proportional probabilities – Out of the box scenario
 - Special pipeline that can be extended with the supporting classifiers from scikit-learn
 - **Challenge:** Classes may be (severely) imbalanced, which will need specific data preparation
 - Provide proportional probabilities of failure involving all types of failure represented by the dataset
- To build failure probability prediction there are two mutually exclusive choices available:
 - Multiclass classification model for multilabel / multi-type / multiclass failures, - OR –
 - Smart Classification model, which supports only binary type failure classification
- **Make sure not to use both techniques for the same set of assets. UI display may not be consistent if both models are deployed for the same assets.**

Maximo Predict : Predicted Failure Date



Model Description

- Predicts when next failures will occur
- Determine if an asset is well-maintained
- Adjust Maintenance Schedule



Inputs Required

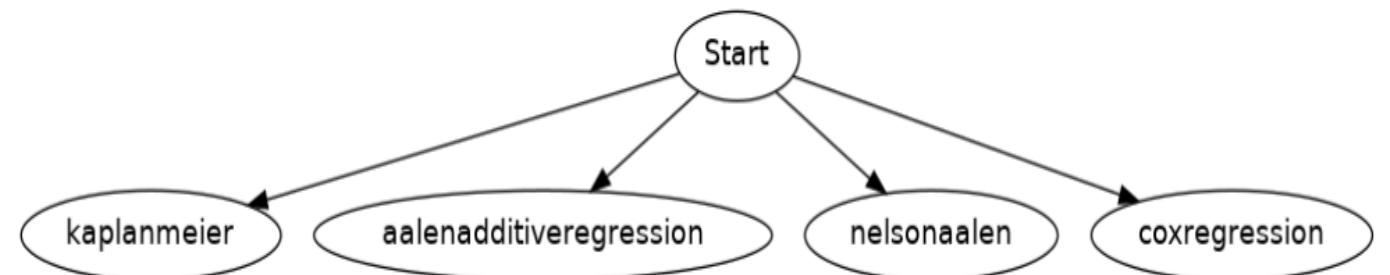
- Timestamped IoT sensor data
- Other process variable as a discrete time variable
- Timestamped failure data



Questions Answered

- How many days until the system fails?

```
estimator_set = [
    ("linearregression", LinearRegression()),
    ("ridge", Ridge()),
    ("lasso", Lasso()),
    ("elasticnet", ElasticNet()),
    ("lassolars", LassoLars()),
    ("orthogonalmatchingpursuit", OrthogonalMatchingPursuit()),
    ("bayesianridge", BayesianRidge()),
    ("sgdregressor", SGDRegressor()),
    ("passiveaggressiveregressor", PassiveAggressiveRegressor()),
    ("kneighborsregressor", KNeighborsRegressor()),
    ("decisiontreeregessor", DecisionTreeRegressor()),
    ("mlpregressor", MLPRegressor()),
    ("gradientboostingregressor", GradientBoostingRegressor()),
    ("adaboostregressor", AdaBoostRegressor()),
    ("baggingregressor", BaggingRegressor()),
    ("randomforestreregessor", RandomForestRegressor()),
    ("extratreesregressor", ExtraTreesRegressor()),
    ("plsregression", PLSRegression()),
    ("gaussianprocessregressor", GaussianProcessRegressor()),
    ("isotonicregression", IsotonicRegression()),
    ("kernelridge", KernelRidge()),
    ("theilsenregressor", TheilSenRegressor()),
    ("ransacregressor", RANSACRegressor()),
    ("huberregressor", HuberRegressor()),
    ("partitionregressor", PartitionRegressor()),
    ("xgbregressor", XGBRegressor()),
    ("gaussianmixtureregressor", GaussianMixtureRegressor())
]
```



Predicted Failure Date (aka Time to Event)

- Two approaches are available
 - Survival Analysis (SA) with proportional hazards model (COX Regression)
 - Smart Regression with many regressors from OLS to more complex ML based techniques

Proportional Hazards Model:

- For datasets structured as trials / experiments / observations
- Requires large number of subjects



asset_id (string)	days run	outcome (failed = 1, didn't fail = 0)	variable 1	variable 2	variable n
P-0983	257	0	73.54	8	39.392
A-3748	312	1	80.44	17	7.493

For a timeseries data Smart Regression approach would be more suitable



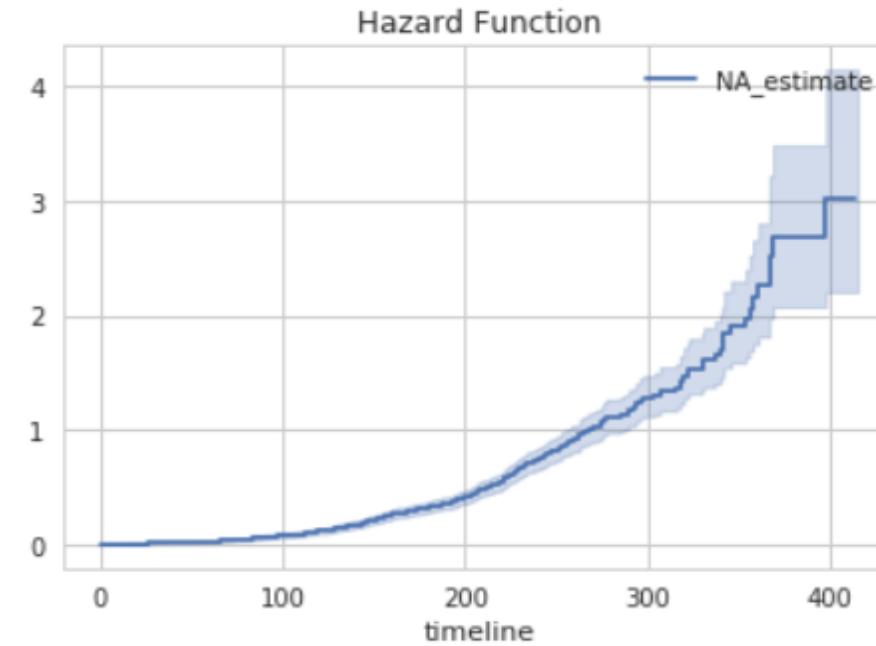
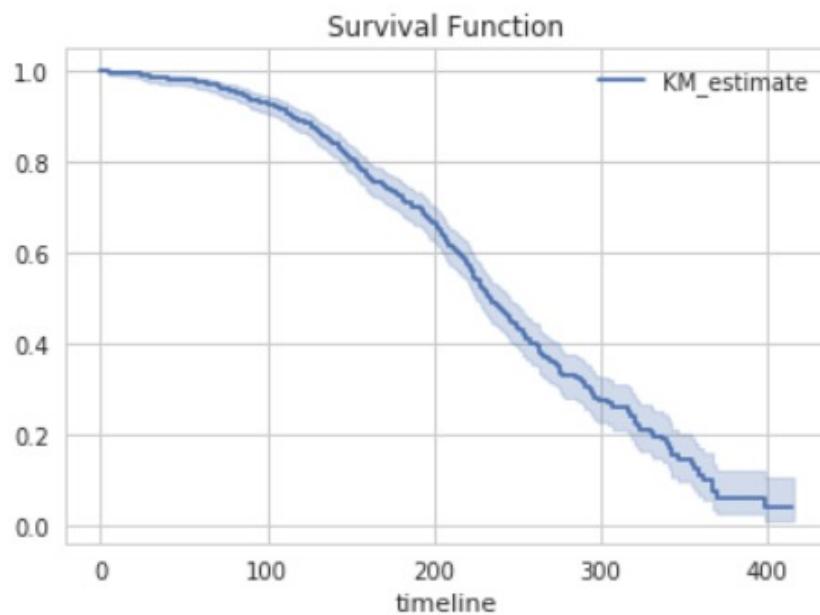
Sensor Data

asset_id (string)	timestamp (UTC recommended)	variable 1 (float)	variable 2 (float)	variable 3 (float)	variable n (float)
P-0983	2006-08-27 19:56:41	73.54	22.56	38.54	39.392
A-3748	2006-08-28 11:32:17	80.44	-26.384	43.965	7.493

Failure Data

asset_id (string)	timestamp (UTC recommended)	failure_id (integer)
P-0983	2006-08-27 19:56:41	1
A-3748	2006-08-28 11:32:17	1

Failure Prediction Date Pipeline and Output Functions



To build failure date prediction there are two mutually exclusive choices available:

- Smart Regression based approach for either binary class or multiclass failure types, - OR -
- Survival Analysis model which supports only binary-class

Make sure not to use both techniques for the same set of assets. UI display may not be consistent if both model are deployed for the same asset(s).

Maximo – Predict Train Models: Failure Contribution Breakdown



Model Description

- Root cause analysis of bad outcome



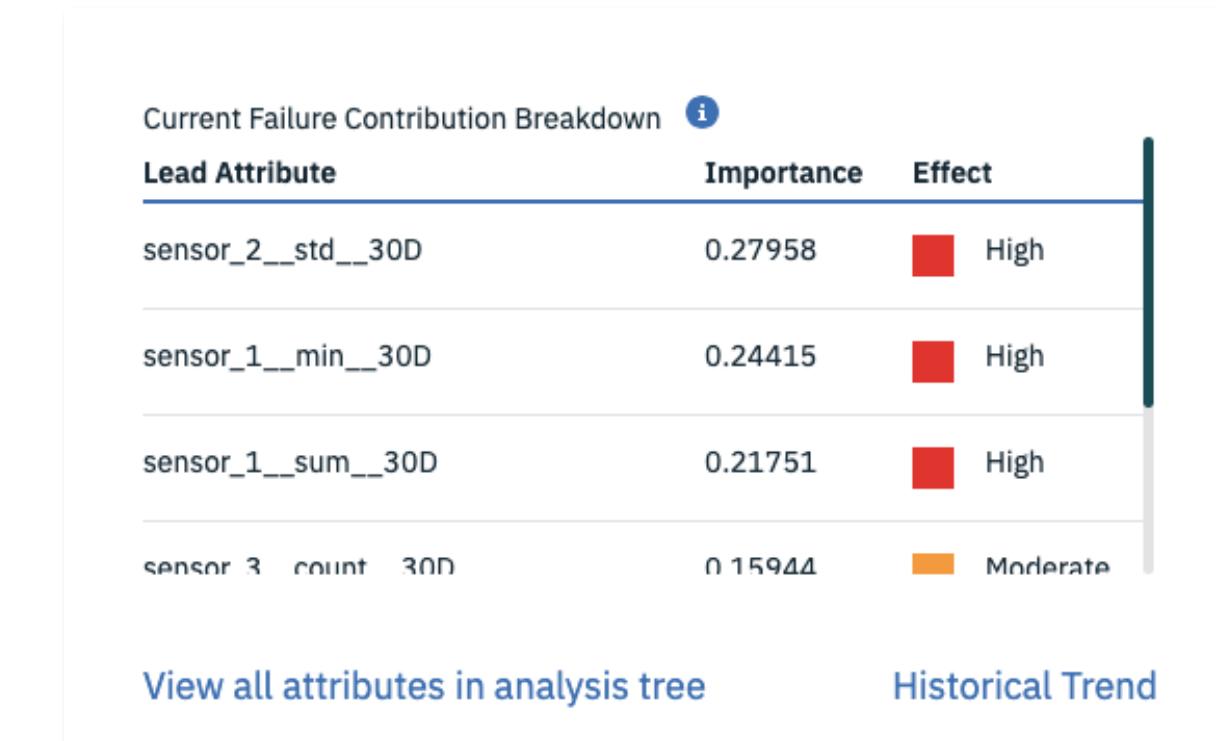
Inputs Required

- Timestamped IoT sensor data
- Other process variable as a discrete time variable
- Timestamped failure data



Questions Answered

- What are the top features that contribute to failure?



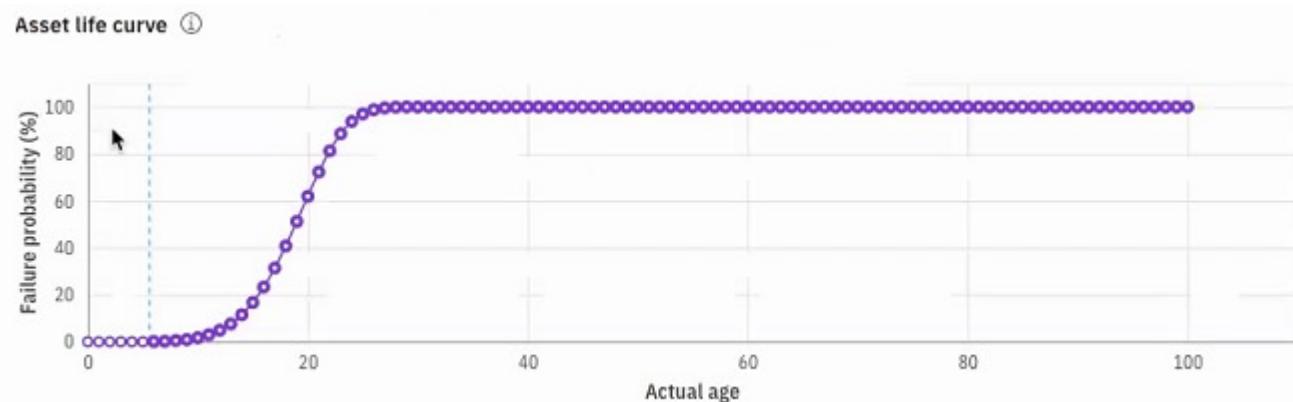
Failure Probability Curve

- Statistically, to evaluate mean life of assets the sample mean or the average age method is acceptable if a big population has end-of-life information. But asset such as generators, transformers, reactors, cables, etc. have a relatively long life up to and even beyond 40 years and generally there are very limited end-of-life failure data. This algorithm is designed to address this use case (to estimate mean life with limited end-of-lift failure data). In fact the proposed algorithm works best when less than 20% of the assets has end-of-life failure data.
- The Failure Probability Curve model uses statistics distribution to assess the failure probability vs. year, and this model has two methods:
 - **Normal Distribution:** a small percentage of assets fail during the early life cycle, a few last beyond the average expected life span, but the majority fails within their mean life.
 - **Weibull Distribution:**
$$\{f(x;\lambda,k)=k\lambda(x\lambda)^{k-1}e^{-(x/\lambda)} kx \geq 0, 0x < 0,$$
where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter.



Inputs Required

- Asset meta data
- Installation Date
- Decommission Date



Custom RUL Model

Three Notebooks:

1. Custom0-RUL-LSTM-WML: Download NASA data, build LSTM model and deploy it into WML.
2. Custom1-RUL-LoadData: Load same data into Maximo and IoT.
3. Custom2-RUL-PMI-Scoring: Train and register custom model in PMI for scoring.

Dataset: NASA C-MAPSS data set (Turbofan Engine Degradation Simulation Data Set)

```
total 5620
-rw-r----- 1 dsxuser dsxuser      429 Apr 16 18:34 RUL_FD001.txt
-rw-r----- 1 dsxuser dsxuser 2228855 Apr 16 18:34 test_FD001.txt
-rw-r----- 1 dsxuser dsxuser 3515356 Apr 16 18:34 train_FD001.txt
```

	engine_no	cycle	os1	os2	os3	sm1	sm2	sm3	sm4	sm5	...	sm12	sm13	sm14	sm15	sm16	sm17	sm18	sm19	sm20	sm21
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	522.86	2388.08	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	522.19	2388.04	8133.80	8.4294	0.03	393	2388	100.0	38.90	23.4044

DQ Learn

- New set of capabilities for data quality analysis, exploratory data analysis
 - Check for deficiencies in the data
 - Check for distributions
 - Check for timeseries patterns
- The following capabilities are available
 - Transactional data with minimal UI
 - Transactional data with regular UI
 - Transactional Pipeline
 - Timeseries data Regular UI (Univariate)
 - Timeseries Pipeline
 - Use case specific DQLearn templates (for AD, FPA etc.)
- WML Deployment capable
- Usage Pattern
 - Use individually for data quality assessment
 - Use with your own pipeline
 - Use the integrated PMI pipeline
 - Deploy on WML as you need

A Tale of Two Pipelines

■ WS Pipeline

- WS*** notebooks
- SROM Pipeline
- Shows the full SROM classifiers
- Takes CSV files as input. No out of the box way to read the data in IoT data lake or Maximo
- Useful for quick experiments, feature creation and establishing a baseline
- The models are deployable only on WML

■ PMI Pipeline

- PMI*** Notebooks
- Wraps around SROM pipeline
- Does not show the classifiers (though in some cases one can override the classifiers)
- Takes the data in the system (data lake, Maximo)
- **Does all the integration internally**
- **Provides the production ready model to deploy on AS**

```
AnomalyDetectionAssetGroupPipeline(  
asset_group_id='ID of an asset group',  
model_pipeline={  
    'features': ['DeviceTypeOne:temperature', 'DeviceTypeTwo:Humidity']  
    'features_for_training': [':failedate'],  
    'predictions': ['anomaly_score', 'anomaly_threshold'],  
    'pre_failure_window_size': 20,  
    'pre_failure_failure_size': 10,  
    'srom_training_options': {  
        'executype': 'spark_node_random_search'  
    }  
})
```

```
TimeToFailureAssetGroupPipeline(  
asset_group_id='ID of an asset group',  
model_pipeline={  
    'features': ['DeviceTypeOne:temperature', 'DeviceTypeTwo:Humidity']  
    'features_for_training': [':installdate', ':failedate'],  
    'predictions': ['predicted_time_to_failure'],  
    'srom_training_options': {  
        'executype': 'single_node_complete_search'  
    }  
})
```

Additional Examples of PMI Pipeline definition

```
FailurePredictionAssetGroupPipeline(  
    asset_group_id='ID of an asset group',  
    model_pipeline={  
        'features': ['DeviceTypeOne:temperature', 'DeviceTypeTwo:Humidity'],  
        'features_for_training': [':faildate'],  
        'predictions': ['failure_probability', 'rca_path'],  
        'prediction_window_size': '5d',  
        'aggregation_methods': ['mean', 'max', 'min', 'median', 'std', 'sum'],  
        'srom_training_options': {  
            'executype': 'single_node_complete_search'  
        }  
        'override_training_stages': my_stage  
    })
```

```
feature_transformation_set = [('skiptransformation', NoOp())]  
scaler_set = [('skipscaling', NoOp()), ('standardscaler', StandardScaler()), ('minmaxscaler', MinMaxScaler())]  
feature_preprocessing_set = [('skipfeaturepreprocessing', NoOp()), ('pca', PCA()), ('selectkbest', SelectKBest())]  
estimator_feature_generator = [('skipmodelfeaturegeneration', NoOp())]  
estimator_set = [('logisticregression', LogisticRegression())]  
my_stage = [feature_transformation_set,  
           scaler_set,  
           feature_preprocessing_set,  
           estimator_feature_generator,  
           estimator_set]
```

Asset data in Maximo

IBM Maximo Application Suite | Predict

NASA_FD1

Group details

Group name: NASA_FD1
Description: Unspecified
Group ID: 1017

Trained instances registered for this group

Results: 0

Model template	Trained instance	Trained instance ID	Schedule	Metrics	Active
No data					

Scored assets

Results: 100

Asset	Description	Site	Type	Failure Class
FD1_ENG_1	FD1_ENG_1	BEDFORD		
FD1_ENG_10	FD1_ENG_10	BEDFORD		

IBM Maximo Application Suite | Predict

Scored assets

Results: 100

Asset	Description	Site	Type	Failure Class
FD1_ENG_1	FD1_ENG_1	BEDFORD		
FD1_ENG_10	FD1_ENG_10	BEDFORD		
FD1_ENG_100	FD1_ENG_100	BEDFORD		
FD1_ENG_11	FD1_ENG_11	BEDFORD		
FD1_ENG_12	FD1_ENG_12	BEDFORD		
FD1_ENG_13	FD1_ENG_13	BEDFORD		
FD1_ENG_14	FD1_ENG_14	BEDFORD		

Predict grouping /

Group details

Group name
VEHICLEINTGKDEDescription
Vehicle groupGroup ID
1014

Trained instances registered for this group

Results: 5

+

Model template	Trained instance	Trained instance ID	Schedule	Metrics	Active
Anomaly Detection	1014_AnomalyDetectionAsset	08BC31C1-F252-49EC-9427-833...	Every 5 min starti...		true
Predicted Failure Date	1014_TimeToFailureAssetGrou	E609CC14-DF75-4907-BB93-4B7...	Every 5 min starti...		true
Failure Probability	1014_FailurePredictionAssetG	E4AC55F3-203E-4F71-B112-477...	Every 5 min starti...		true
Failure Probability	1014_FailurePredictionAssetG	EDBF62C4-827C-4049-9A63-43F...	Every 5 min starti...		true
Failure Probability Curve	1014_DegradationCurveAssetC	C862019E-34D6-4DC9-8AA6-21A...	Every 5 min starti...		true

Items per page: 10 ▾ 1–5 of 5 items

1 ▾ 1 of 1 pages

Scored assets

Results: 51

Asset	Type	Site	Installation Date	Age in Years	Total Cost
KDE-1597		BEDFORD	09/22/1975	44.9	0.00

Maximo Objects to read failures

- In Maximo, Failure codes are part of the categorization mechanism of Work Orders and Tickets. They are usually entered when closing the WO/Ticket or when the reason of the failure is known.
Failure codes represent the hierarchical structure describing the cause of the failure and suggesting the potential solution.
- Failure Code can be read from the following hierarchy. Note this may be customized differently across the Maximo instances
- Class
 - Problem
 - Cause
 - Remedy

[PUMPS/STOPPED]

[PUMPS/STOPPED/MOTRFAIL]

[PUMPS/STOPPED/MOTRFAIL/RESET]

Failure Details

Failure Class: PUMPS > Pump Failures

Failed Date: 2/9/17 9:06 AM

Failure Codes	↑	↓	← 1 - 3 of 3 →	⬇	✖
Type	PROBLEM	Failure Code	BROKEN		
CAUSE	PUMPJAM	Description	PUMP JAM		
REMEDY	REPLACE		REPLACE		

Failure Codes:

[PUMPS/BROKEN]

[PUMPS/BROKEN/PUMPJAM]

[PUMPS/BROKEN/PUMPJAM/REPLACE]

Maximo Failure Codes – A screenshot

The screenshot shows the Maximo Failure Reporting interface. At the top, the URL is https://sb190128a.stage.maximo.com/maximo/ui/maximo.jsp?event=gotoapp&value=wotrack&uisessionid=132&_tt=. The title bar includes Release 1.1.9.post1 ..., Your Learning, Slack | Shobha Mal... and various browser icons. The main menu bar has tabs: List View, Work Order, Plans, Assignments, Related Records, Actuals, Safety Plan, Log, Data Sheet, Failure Reporting (which is selected), Specifications, Service Address, and Map. On the left, there's a sidebar with a checkmark icon, a plus sign, a location pin, a refresh arrow, a back arrow, a forward arrow, a document icon, a printer icon, and a trash bin icon. The main content area shows a work order APM8123781 with Site: BEDFORD and Status: WAPPR. Below this is a 'Failure Details' section with a tree view showing Failure Class: PUMPS expanded to Pump Failures, Failed Date: 1/19/17 11:29 PM, and Remarks and Remark Date fields. At the bottom, a table titled 'Failure Codes' lists three entries: PROBLEM STOPPED stopped, CAUSE MOTRFAIL MOTOR FAIL, and REMEDY RESET. The 'Failure Code' and 'Description' columns have edit and delete icons. The 'Type' column has edit and delete icons. The 'REMEDY' row has edit and delete icons.

Failure Details

Failure Class: PUMPS > Pump Failures

Failed Date: 1/19/17 11:29 PM

Failure Codes

Type	Failure Code	Description		
PROBLEM	STOPPED	stopped	[edit]	[delete]
CAUSE	MOTRFAIL	MOTOR FAIL	[edit]	[delete]
REMEDY	RESET		[edit]	[delete]

CLASS=PUMPS, PROBLEM=STOPPED,CAUSE=MOTRFAIL,
REMEDY=RESET

WML Deployment

Using Deployment Spaces

- On CP4D the supported platform is WML Client API V4
- Uses Deployment Spaces
- Notebooks will support both SaaS as well as CP4D with minor Config Changes
- CP4D deployment requires Deployment Spaces

```
import os
execution_env = os.environ.get('AX_EXT_DEPLOYMENT_TYPE', 'wscloud') #'icp alternatively cloud.
store_artifacts = False
bucket_name_for_pickled_models = 'maximo-predict-pickled-models' # Keep in mind that neither
#atters in the name, or only those allowed (hyphen or dash is allowed by ICOS)
local_data_folder = 'project_data/data_asset'
deployment_space = 'MAS-Testing-Deployment-Space'
```

```
ad_pipeline_endpoint = None
if execution_env == 'icp':
    from srom.cloud.wml.cloud_scoring_v4 import WMLScorer
    wml_scorer_trained_pipeline = WMLScorer()
    wml_scorer_trained_pipeline.connect(wml_credentials=wml_credentials, deployment_space_name=deployment_space)
    wml_scorer_trained_pipeline.add_local_package(srom_zip_file, version='0.1')
    wml_scorer_trained_pipeline.add_local_package(dq_learn_zip_file, version='0.1')
    ad_pipeline_endpoint = wml_scorer_trained_pipeline.deploy(model=pipeline, name='ad_pipeline_srom_121rc3', randomize_name=True)
    print(ad_pipeline_endpoint)
else:
    from srom.cloud.wml.cloud_scoring import WMLScorer
    wml_scorer_trained_pipeline = WMLScorer()
    wml_scorer_trained_pipeline.connect(wml_credentials)
```

Troubleshooting Tips

Minor Error message when install pmlib

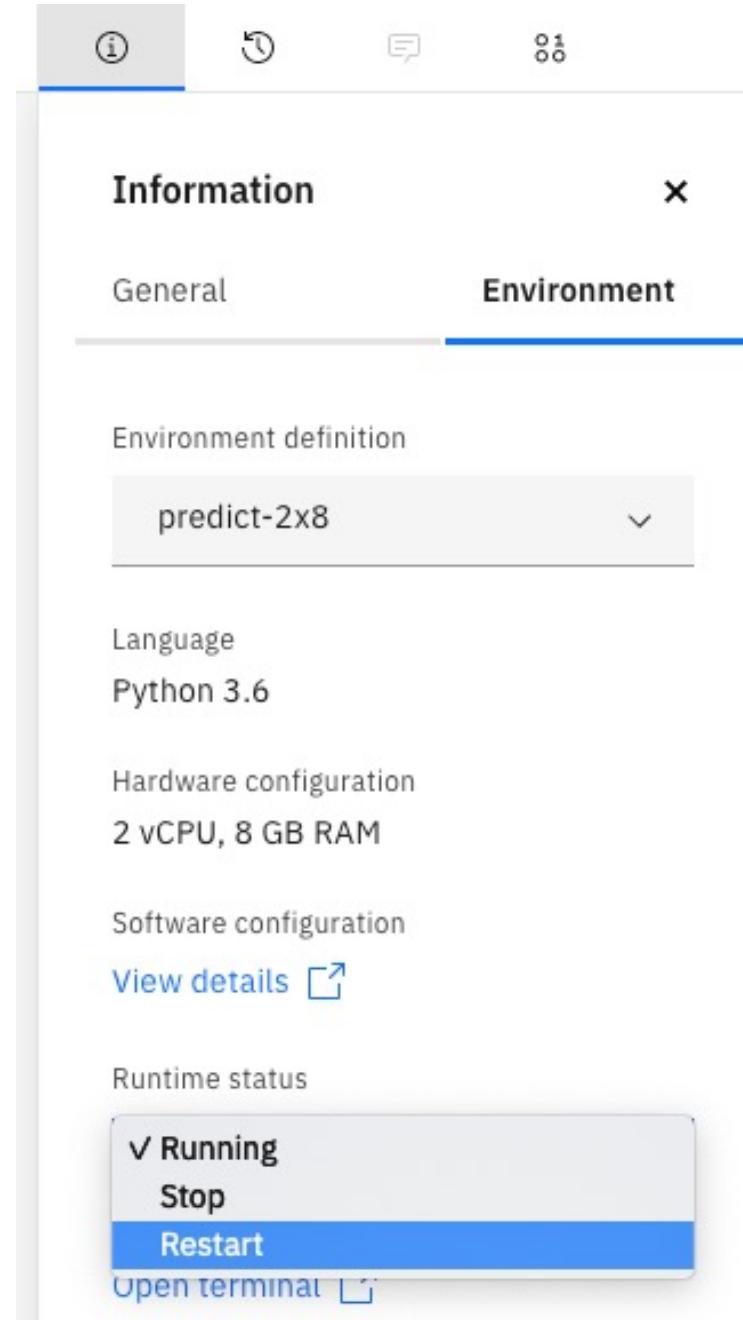
Minor error message can be ignored.

```
!pip install --trusted-host ${TRUST_PREDICT} -U ${APM_API_BASEURL}/ibm/pmi/service/rest/ds/${APM_ID}/  
${APM_API_KEY}/lib/download?filename=pmlib-1.0.0.tar.gz
```

```
....  
Successfully built pmlib iotfunctions  
ERROR: iotfunctions 2.0.3 has requirement ibm-db==3.0.1, but you'll have ibm-db 3.0.2 which is incompatible.  
ERROR: iotfunctions 2.0.3 has requirement ibm-db-sa==0.3.3, but you'll have ibm-db-sa 0.3.4 which is incompatible.  
ERROR: iotfunctions 2.0.3 has requirement scikit-learn==0.20.3, but you'll have scikit-learn 0.20.4 which is incompatible.  
ERROR: iotfunctions 2.0.3 has requirement sqlalchemy==1.3.10, but you'll have sqlalchemy 1.2.18 which is incompatible.  
Installing collected packages: iotfunctions, pmlib  
Successfully installed iotfunctions-2.0.3 pmlib-1.0.0
```

CP4D runtime issue

- If you see the error "pmlib not found" in the notebook, you need to runtime Environments (see screenshot). Then restart the notebook Kernel.



Minute and Second to be ignored when set scheduler in Python

```
group.enable(enabled=True, schedule={"starting_at": "2020-08-28 12:55:15", "every": "1day"})
```

Update trained instance X

1021_TimeToFailureAssetGroupPipeline_2020-08-18T19:04:57.579928 (i)

Activate and schedule the scoring frequency

Model Template Predicted Failure Date

Interval in days NA

Active On

Scoring schedule

Run every ▼ 1 ▲ Day ▼

Starting at 12:00 ▼ AM ▼

08/28/2020 □

Cancel Save

Watson Studio timeout issue

- If the group.register() fails due to Time Out error, you can retry it.

```
: # Register model in Maximo-Predict
group.register(model_instance_name="FD_RUL1", model_instance_desc="RUL 4 engine FD001")
```

To get scoring log

- You can use the below query in Monitor DB to see the scoring log file:

```
SELECT STATUS, LOG_MESSAGE, LOGFILE, STARTED_TS, UPDATED_TS  
FROM TEST_MAM.KPI_LOGGING  
WHERE logfile LIKE '%pmlib%'  
AND logfile LIKE '%APM_1021%'      --- 1021 is your asset_group_id  
ORDER BY started_TS DESC LIMIT 10;
```

Failure Probability Prediction-Binary Classification data requirement

- The percentage of failure records can not be greater than 50% of the total training data. For example, the number of total training data set is 800 and there are 700 failure records. This will cause training failure. You need to reduce the number of failure records.

WML deployment failed for Failure Date prediction (Survival model)

<https://github.ibm.com/asset-performance/APM-PM/issues/1763>

<https://github.ibm.com/asset-performance/APM-PM/issues/1666>

Work around is to deploy the model in Monitor

Enhancement pending with WML team. <https://github.ibm.com/NGP-TWC/ml-planning/issues/17651>

WML SROM Wrapper uses deprecated runtimes instead of software spec

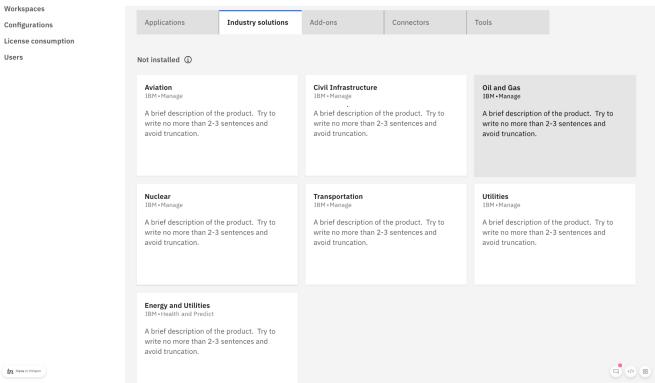
WML wrapper for SROM (version v4 API client) uses runtimes construct which is being deprecated. Software Spec should be used for compliant with the future direction. **This issue can be ignored for now.**

```
Warn: fixing name with non-alphanumeric characters
Warn: name is now feature_engg_deployment_
Warn: fixing name with non-alphanumeric characters
Warn: name is now srom_1_2_1rc3_zip_
WARNING!! 'runtimes' is DEPRECATED. Use 'software_specifications' instead to create and manage runtimes/specifications
Warn: fixing name with non-alphanumeric characters
Warn: name is now feature_engg_deployment_
WARNING!! 'runtimes' is DEPRECATED. Use 'software_specifications' instead to create and manage runtimes/specifications
WARNING!! 'runtimes' is DEPRECATED. Use 'software_specifications' instead to create and manage runtimes/specifications
Warn: fixing name with non-alphanumeric characters
Warn: name is now feature_engg_deployment_
2020-06-30T15:20:19.956 ModelCollection._save_scikit_pipeline_model DEBUG Creating a new scikit pipeline model: feature_engg_deployment__
2020-06-30T15:20:20.583 watson_machine_learning_client.libs.repo.swagger_client.rest.request DEBUG response body: {
  "metadata": {
    "name": "feature_engg_deployment__",
    "guid": "d179eedd-a65a-4d9b-be2c-9fde43a6d027",
    "id": "d179eedd-a65a-4d9b-be2c-9fde43a6d027",
    "modified_at": "2020-06-30T15:20:20.002Z",
    "created_at": "2020-06-30T15:20:20.002Z",
    "owner": "1000330999",
    "href": "/v4/models/d179eedd-a65a-4d9b-be2c-9fde43a6d027?space_id=9f62f07b-ce9c-483f-a1db-098de3356dd1",
    "space_id": "9f62f07b-ce9c-483f-a1db-098de3356dd1"
  },
  "entity": {
    "name": "feature_engg_deployment__",
    "content_status": {
      "state": "no_content"
    },
    "space": {
      "id": "9f62f07b-ce9c-483f-a1db-098de3356dd1",
      "href": "/v4/spaces/9f62f07b-ce9c-483f-a1db-098de3356dd1"
    },
    "type": "scikit-learn_0.20",
    "runtime": {
      "id": "47af01d7-9598-42ba-862c-c9d398f69674",
      "href": "/v4/runtimes/47af01d7-9598-42ba-862c-c9d398f69674"
    }
  }
}
```

End

Maximo Application Suite v8.x.0 Health & Predict - Utilities

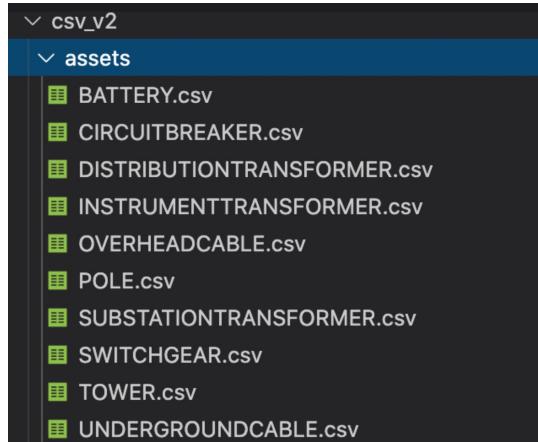
Maximo Health & Predict - Utilities tile



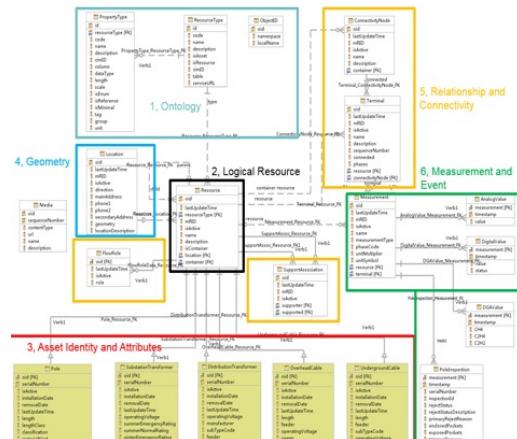
6 asset class python models

```
[Bushing Condition]
type = contributor
desc = Transformer Bushing Condition
func = common_calculate_condition_by_meter
# parameter.<type>.<item>
parameter.meter.name = Bushing Condition meter
parameter.meter.type = string
parameter.meter.format = meter
parameter.meter.desc = meter to measure Bushing Condition
parameter.meter.default = B-CONDIT
# table.<table name>
table.condition = condition, score, description
"A", "4", "Bushings are not broken and are free of chips,
"B", "3", "Bushings are not broken, however minor chips and
"C", "2", "Bushings are not broken, however major chips, and
"D", "1", "Bushings are broken/damaged or cementing and fasteners are broken/damaged
"E", "0", "Bushings, cementing or fasteners are broken/damaged"
```

AppConnect data loader scripts



Industry standard data model



Extended scoring methodologies (effective age, end of life probability)

Watson Studio notebook scoring

Watson Studio notebook scoring interface showing a scoring group named "Distribution transformers". The interface includes fields for Name, Description, and a list of notebooks. The "Notebooks" section shows a search bar and a table for managing notebooks.

MAS Health & Predict Utilities – Asset Investment Optimization

Use case

- Create asset investment projects based on risk and budget
- Leverage replacement templates for common asset types
- Run optimization engine to determine when assets should be replaced and when
- Compare strategies for maintaining risk, reducing risk, or optimize within budgets

Value

- Easily identify the best plans for asset replacement
- Submit plans for approval or additional financial analysis

The screenshot shows the 'Asset investment optimizer / Grid assets at Bedford' page. It displays a list of investment strategies:

- Maintain risk**: Last updated: 2020-03-02 08:00. Status: Current. Average risk score: 44.1. Constraints: 2.
- Maintain risk**: Last updated: 2020-03-02 08:00. Status: Current. Average risk score: 44.1. Constraints: 2.
- Reduce risk**: Last updated: 2020-03-02 08:00. Status: Current. Target risk score: 40.0. Constraints: 1.
- Reduce risk**: Last updated: 2020-03-02 08:00. Status: Pending. Target risk score: 30.0.

Below the strategies, there is a table titled 'Optimized schedule and costs' showing asset details, site ID, health, risk, replacement in, total CAPEX, and total OPEX for various assets like AH012 and SITE1234.

The screenshot shows the 'Asset investment optimizer / Grid assets at Bedford / Compare investment strategies' page. It compares four strategies across different goals:

Goal	Maintain risk	Reduce risk	Stay in budget
Duration	5 years	5 years	5 years
Target start year / end year	1Q 2021 / 4Q 2025	1Q 2021 / 4Q 2025	1Q 2021 / 4Q 2025
Total cost (USD)	120,000,000	90,000,000	50,000,000 (Lowest)
CAPEX	90,000,000	70,000,000	30,000,000 (Lowest)
OPEX	30,000,000	20,000,000	20,000,000 (Lowest)
Average yearly total cost (USD)	400,000	470,000	100,000 (Lowest)
2021 total	250,000	110,000 (Lowest)	110,000 (Lowest)
2022 total	150,000	450,000	100,000 (Lowest)
2023 total	150,000	150,000	100,000 (Lowest)
2024 total	150,000	350,000	100,000 (Lowest)
2025 total	150,000 (Lowest)	750,000	100,000 (Lowest)
Average risk score for duration	5.1 (Lowest)	9.0	13.3
Highest risk score for duration	Risk score: 11.0 (2022), Trend from current: -1.5	Risk score: 10.0 (2022), Trend from current: -1.5	Risk score: 10.0 (2022), Trend from current: -1.5
Distribution transformer	3.3 (2023)	5.3 (2023)	9.0 (2023)
Transmission transformer	7.1 (2022)	12.0 (2022)	12.0 (2022)
Overhead cable	4.1 (2021)	12.0 (2022)	12.0 (2022)
Switch	1.1 (2024)	1.1 (2024)	1.1 (2024)
Underground cable	1.1 (2024)	1.1 (2024)	1.1 (2024)
Total replacements	120	120	120

MAS 8.x Utilities Scoring Model extensions

- Leverages pre-defined python scoring models in Watson Studio Notebooks

Includes 6 pre-defined asset models

- Circuit Breaker Air
- Circuit Breaker Oil
- Gas Insulated Switchgear
- Substation Transformer
- Distribution Transformer
- Instrument Transformer

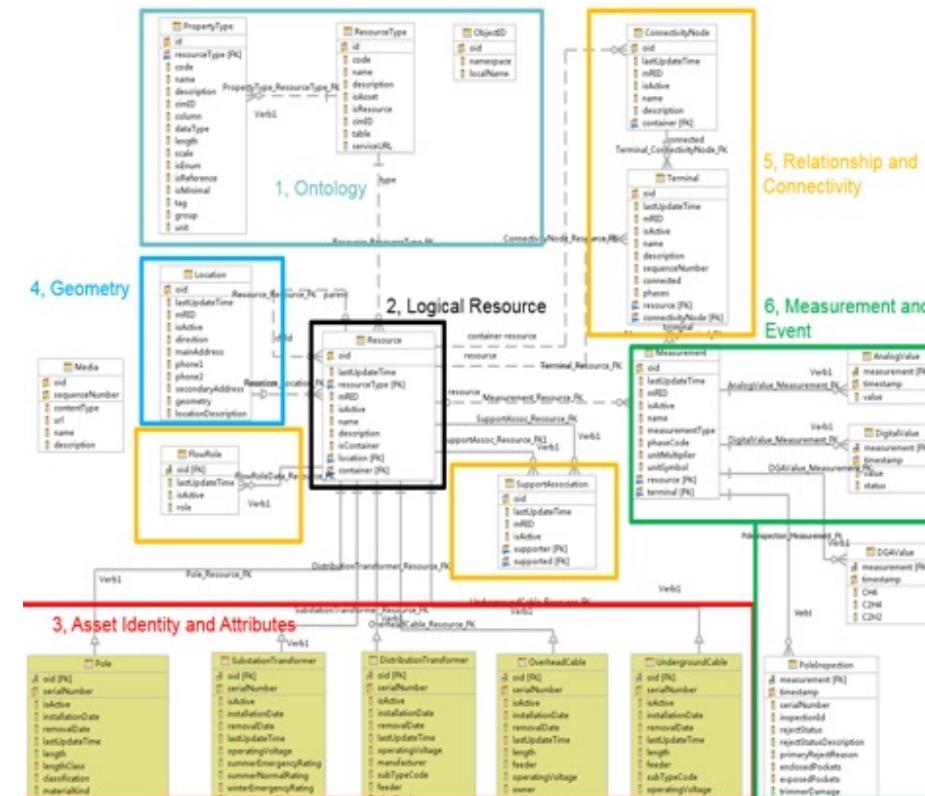
Ability to associate an asset group in Health with a notebook containing pre-defined scoring models

The screenshot shows the 'Scoring' section of the IBM Maximo Health and Predict interface. It includes a 'Create a scoring group' form with a 'Name' field set to 'Distribution transformers'. Below it is a 'Select notebook' section with a search bar and a table of notebooks. The table has columns for 'Notebook name' and 'Notebook description here...'. A red arrow points from this interface to a code snippet.

```
[Bushing Condition]
type = contributor
desc = Transformer Bushing Condition
func = common_calculate_condition_by_meter
# parameter.<type>.<item>
parameter.meter.name = Bushing Condition meter
parameter.meter.type = string
parameter.meter.format = meter
parameter.meter.desc = meter to measure Bushing Condition
parameter.meter.default = B-CONDIT
# table.<table name>
table.condition = condition, score, description
"A", "4", "Bushings are not broken and are free of chips,
"B", "3", "Bushings are not broken, however minor chips and
"C", "2", "Bushings are not broken, however major chips, a
"D", "1", "Bushings are broken/damaged or cementing and fasteners are broken/dam
"E", "0", "Bushings, cementing or fasteners are broken/damaged"
```

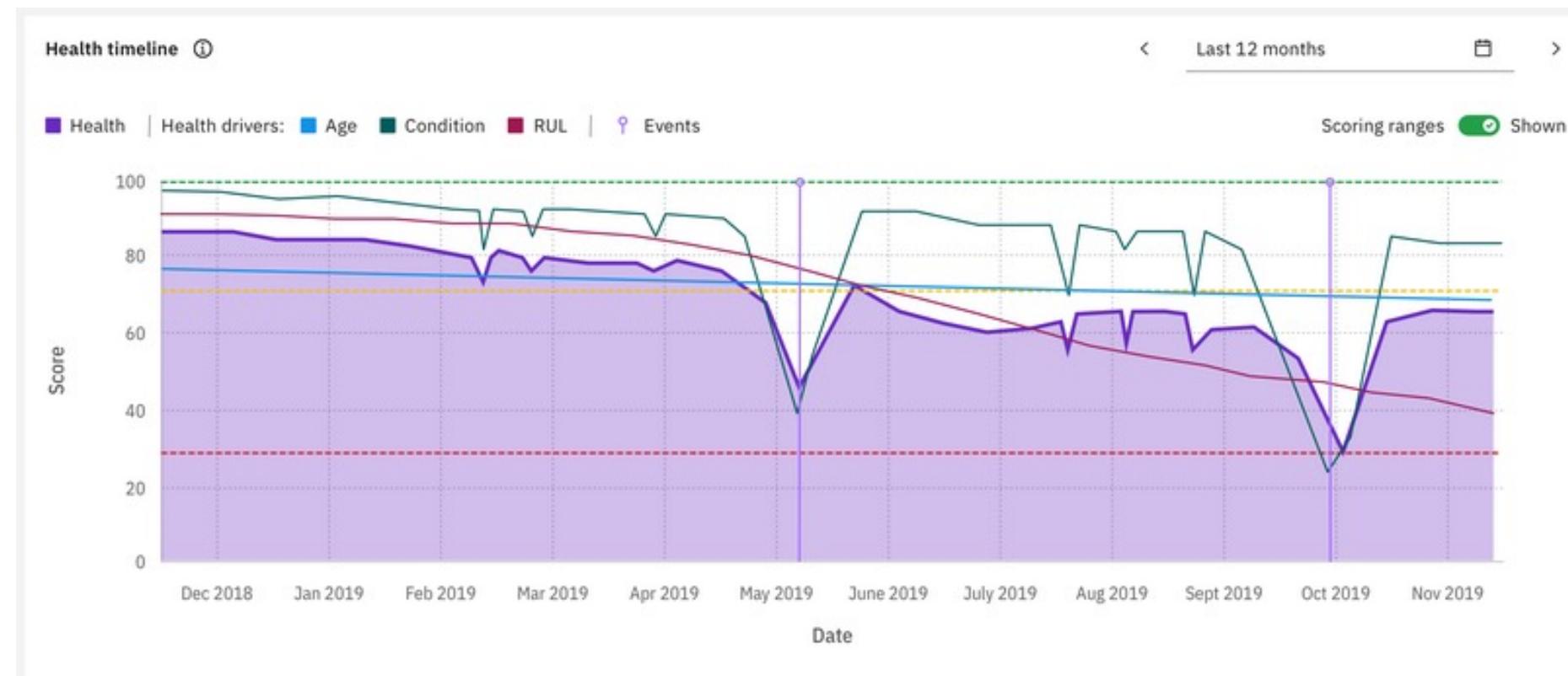
MAS 8.x Utilities Common Information Model (CIM)

- CIM is a standard data model for describing assets information in Utilities industry
 - Allows the customer to import data formatted based on CIM data model into Health or Manage
-
- Leverages AppConnect
 - Includes pre-built scripts for data model loading



MAS 8.x Health history

- Provides view of the trends for the health score of an asset over time
- Trends are valuable in assessing actions to take based on a low health score
- Includes surface area for overall health
- Lines for individual contributors
- Demarcation of ranges
- Toggle on/off drivers, ranges
- Customizable time ranges



Maximo Health and Predict – Utilities – Code Execution flow

Standard order of Python code execution

- Environment connection to underlying EAM / MAXIMO application --> i.e., validation of "Predict_Envs.json" file
- Validation of "PMLIB" Library, PMLIB is IBM's library holding majority of pre-defined models
- Download of Other important libraries
- There are generally 2 types of python programs available as OOB to derive predictive scores
 - Stand alone python program starting with "PMI—XXX"
 - Detail python programs starting with "WS-XXX"
- WS-xxx programs are mainly executed to run multiple machine learning models given the datasets
- Register the models for frequent Run with-in the UI application
- Options available for Data Scientists to run "multiple" mathematical models to evaluate different outcomes
- Repeated execution of different mathematical models & evaluation of outcomes to determine best score methods