MAS Health & Predict

Maximo Application Suite **Health & Predict**

Brief Definition

- **Health and Predict** – Solution designed to bring out "ASSETS" conditions 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.

Type	Business Objective
Health	Score that reflects the overall health and condition of an asset based on key contributing measurements, calculations, inspection values and service history.
Criticality	Score that reflects how important this asset is to the overall business objectives that can include impact to system performance and impacted customers.
Risk	Score that reflects both financial and non-financial impact an asset has if the asset were to stop working or fail
Effective Age	How old the asset is in terms of its remaining useful life based on its calendar age since it was installed, amount of use, operating conditions, location and other factors.

Scores Composed Using

- Internal Formula Engine
- Jupyter Notebook
- SPSS Algorithms

Scores are Calculated Using

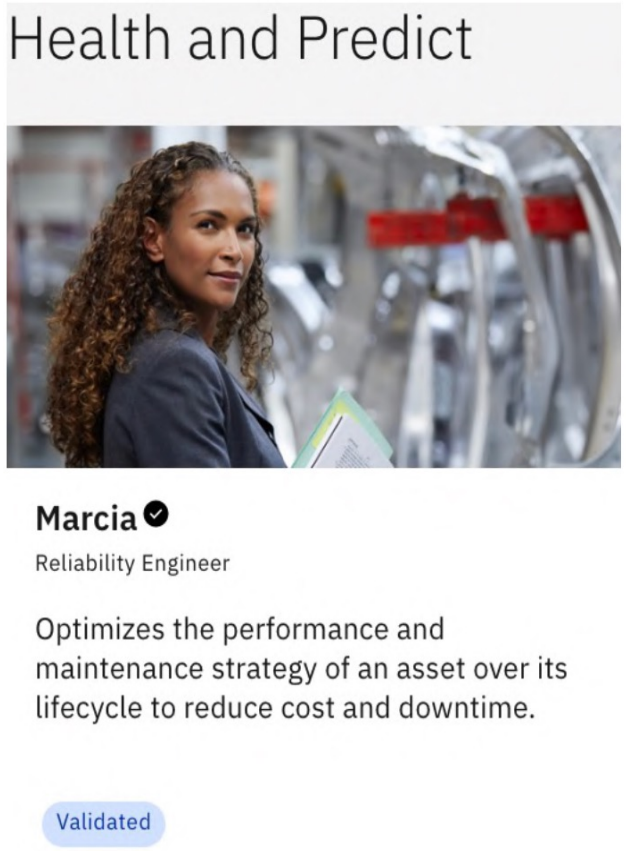
- Machine Learning Service
- Watson Studio Job

Type	Business Objective
Probability of failure	What is likelihood of failure within a time period. How much risk is there? Statistical health score of assets. Preventive maintenance to prolong life, plan for capital replacement, set forecasts.
Predicted failure date	Incorporate modeled failure data to guide scheduling decisions for repair and replacement.
Anomaly detection	Identify situations where operations are abnormal and may impact production, asset performance, quota, etc.
End of life curve	Forecast remaining useful life of the asset to aid capital replenishment strategy.

Persona, Why is it important

– As they understand the importance of challenges of day-to-day operations
– Probable scattered of knowledge or improper documentation
– Right decision makers of business operations

You are Marcia



Job: Reliability Engineer (RE)

Role: Maintain the reliability of the grid

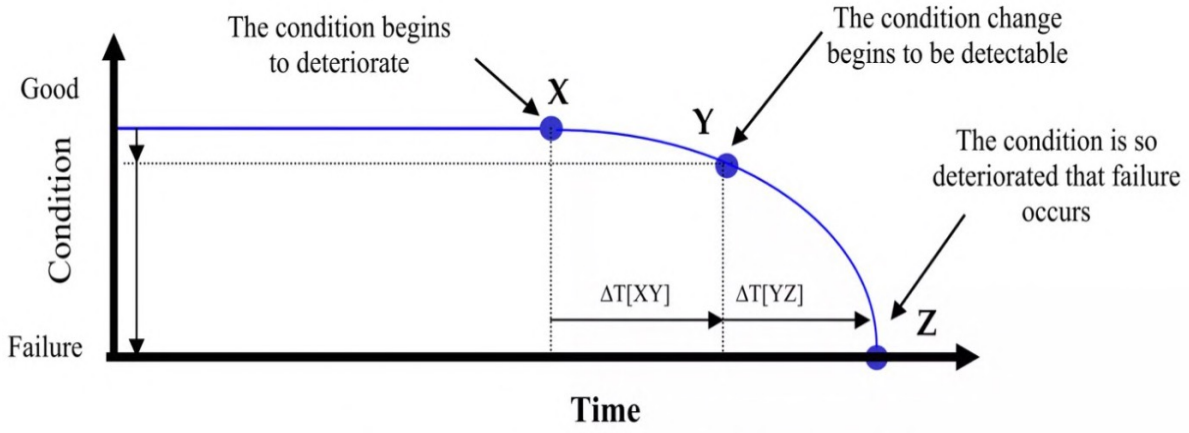
Responsibilities: Ensure that the grid stays up and operational; make short, medium and long-term investment decisions that drive maintenance and replacement policies

Tasks: Asses and prioritize asset condition using operational data and predictive condition insights for Asset Health, Anomaly Detection, Probability of Failure and Risk (now and into the future)

Focus…

Focus on problems you can forecast and action

-At point X, the condition is still good but is starting to change
-At point Y, the condition has changed enough to be detectable by a specific diagnostic technique
-At point Z, the condition has deteriorated sufficiently to cause a transformer failure



KPIS

- Understand frequency of measurement vs rate of change that can lead to failure
- Apply anomaly detection to KPIs rate of change and correlate with other alarms and affects
- Count alerts/alarms exceptions trends (hour, day, week, month, year)
- Measure time in state
- Use inspection to avoid failures from affects you can't forecast or action… lightening grounding systems are present.

Maximo Health and Predict – Value

Business Value

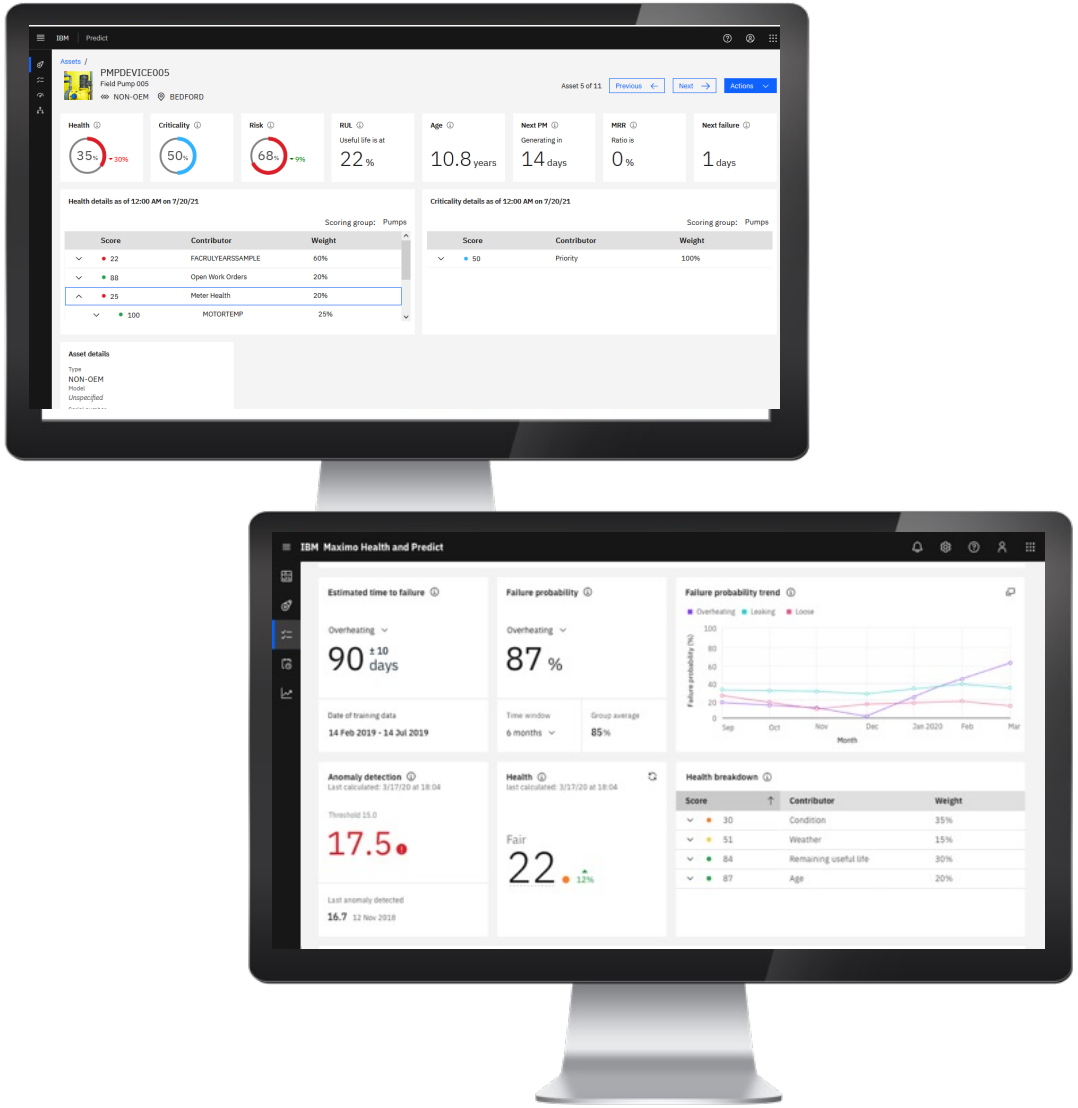
– Reduce operational risk by focusing on the right assets
– Increase asset availability
– Reduce unnecessary preventive maintenance
– Reduce time to make capital replacement planning decisions

Application

– Flexible analysis of the performance comparison of assets
– Work queues provide guided approach to reducing risk
– Dashboard drill down for complete analysis of asset performance indicators including predictions for imminent failure
– Action menu to initiating maintenance or replacement actions to reduce risk

Capabilities

– Dashboard with cards, map view, list view
– Fleet-wide view and health drilldown
– Predicted failures and failure probabilities based on pre-defined model templates
– Flexible health, criticality and risk scoring by asset type or groups



Maximo Health and Predict - Utilities

Business Value

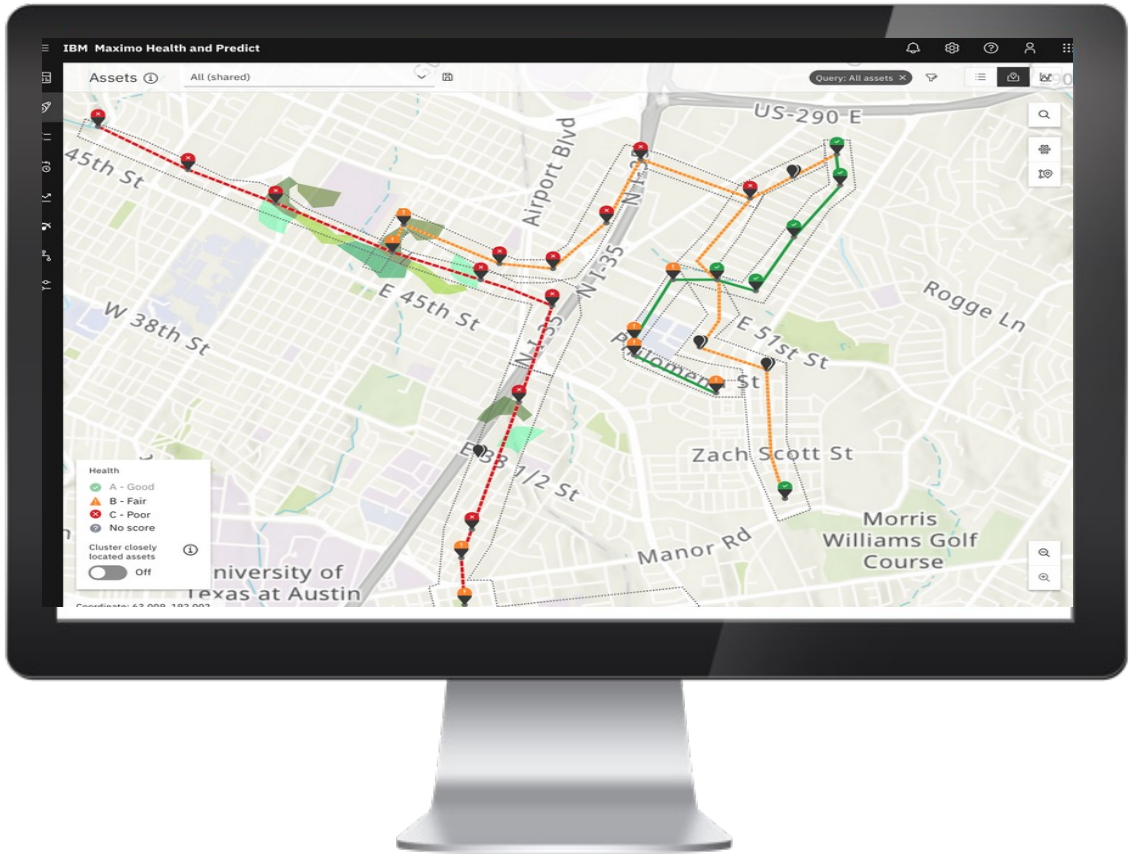
– Reduce operational risk by focusing on the right assets
– Increase asset availability
– Reduce unnecessary preventive maintenance
– Reduce time to make capital replacement planning decisions

Application

– Builds on Maximo Health and Predict
– Extended scoring methodologies for Utilities industry
– Industry standard data model
– Predefined asset models
– Asset Investment Optimzer

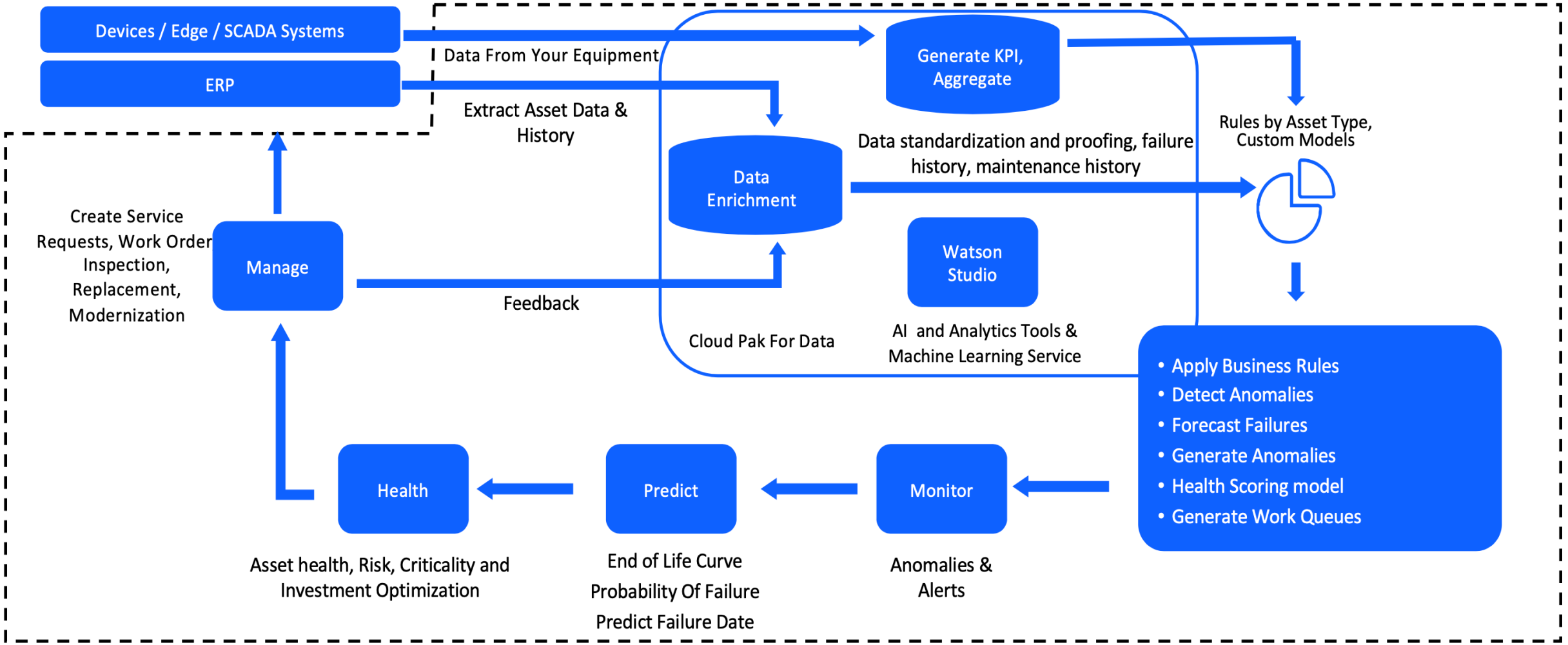
Capabilities

– Dashboard with cards, map view, spreadsheet view
– Fleet-wide view and health drilldown
– Flexible health, criticality, and risk scoring by asset type or groups



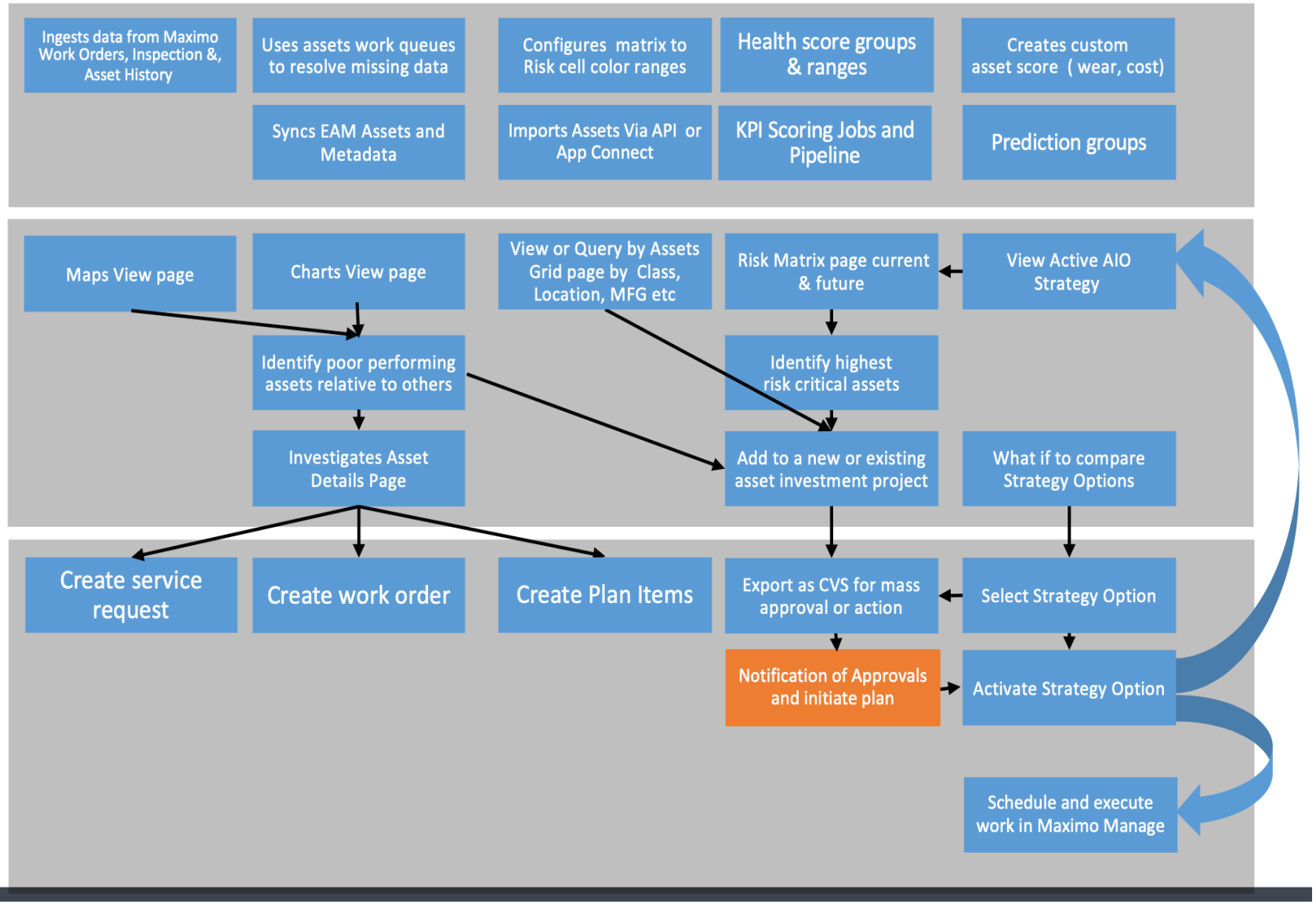
How does Health & Predict work

Maximo Application Suite – Closed Loop with Continuous Optimization

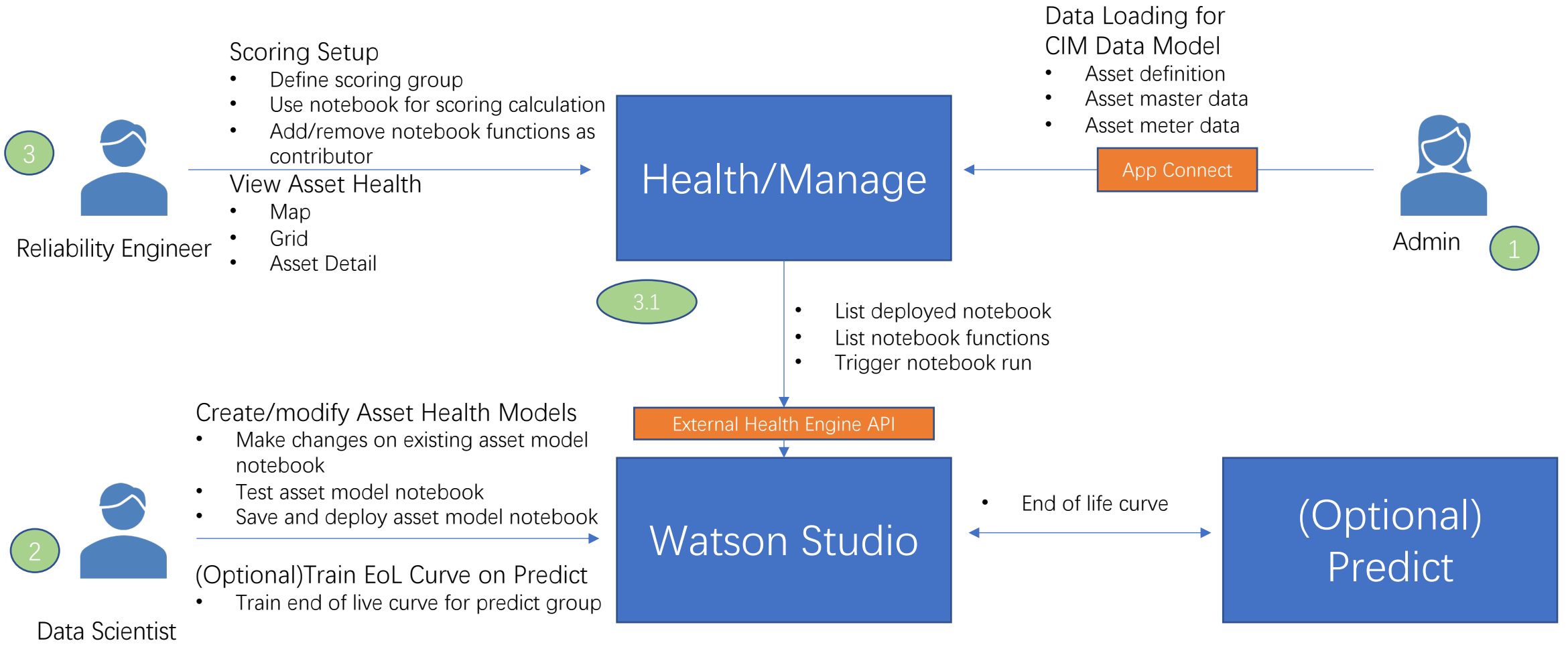


Asset Investment Setup, assessment and actioning asset

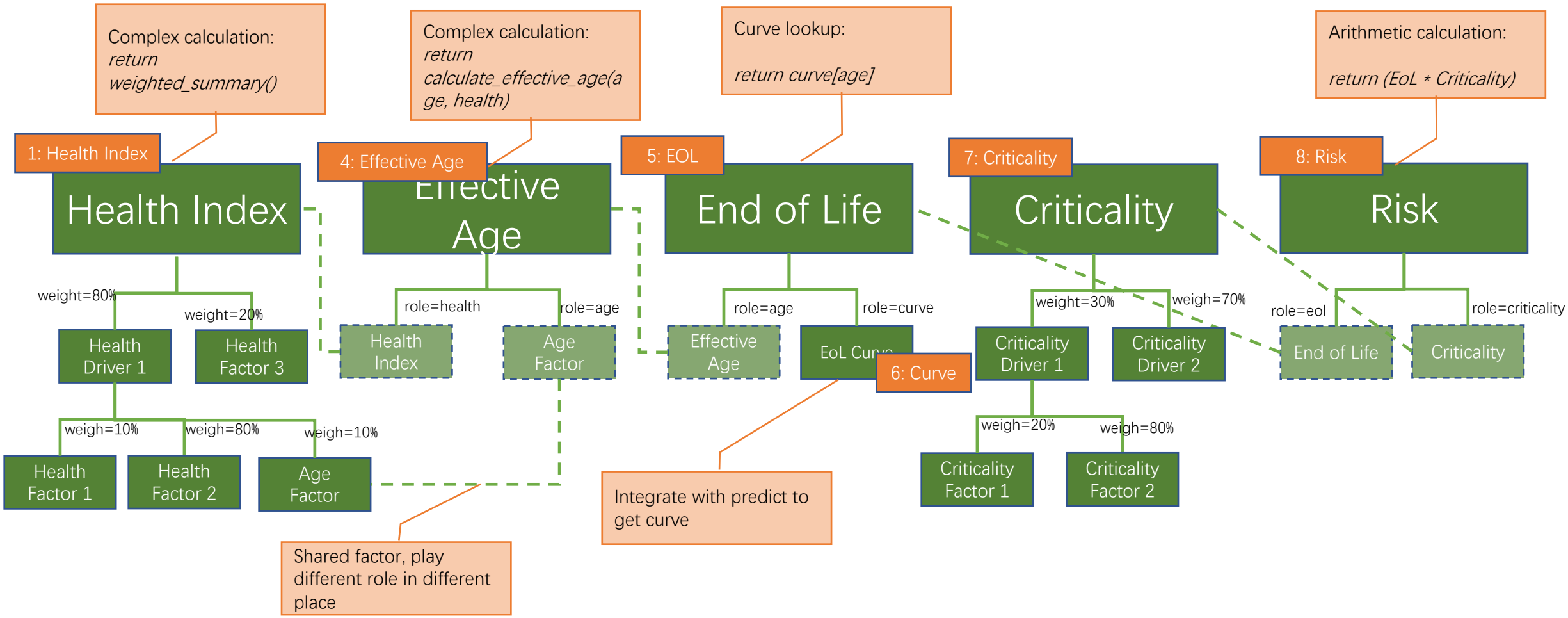
1. Admin – Sets up assets, Health, Risk, EOL Probability, Risk, Asset replacement plan template and Criticality Score and Color ranges and AI predictions
2. Reliability Engineer identifies and investigates high risk assets plans and selects best investment option
3. Take action on critical and poor at risk assets



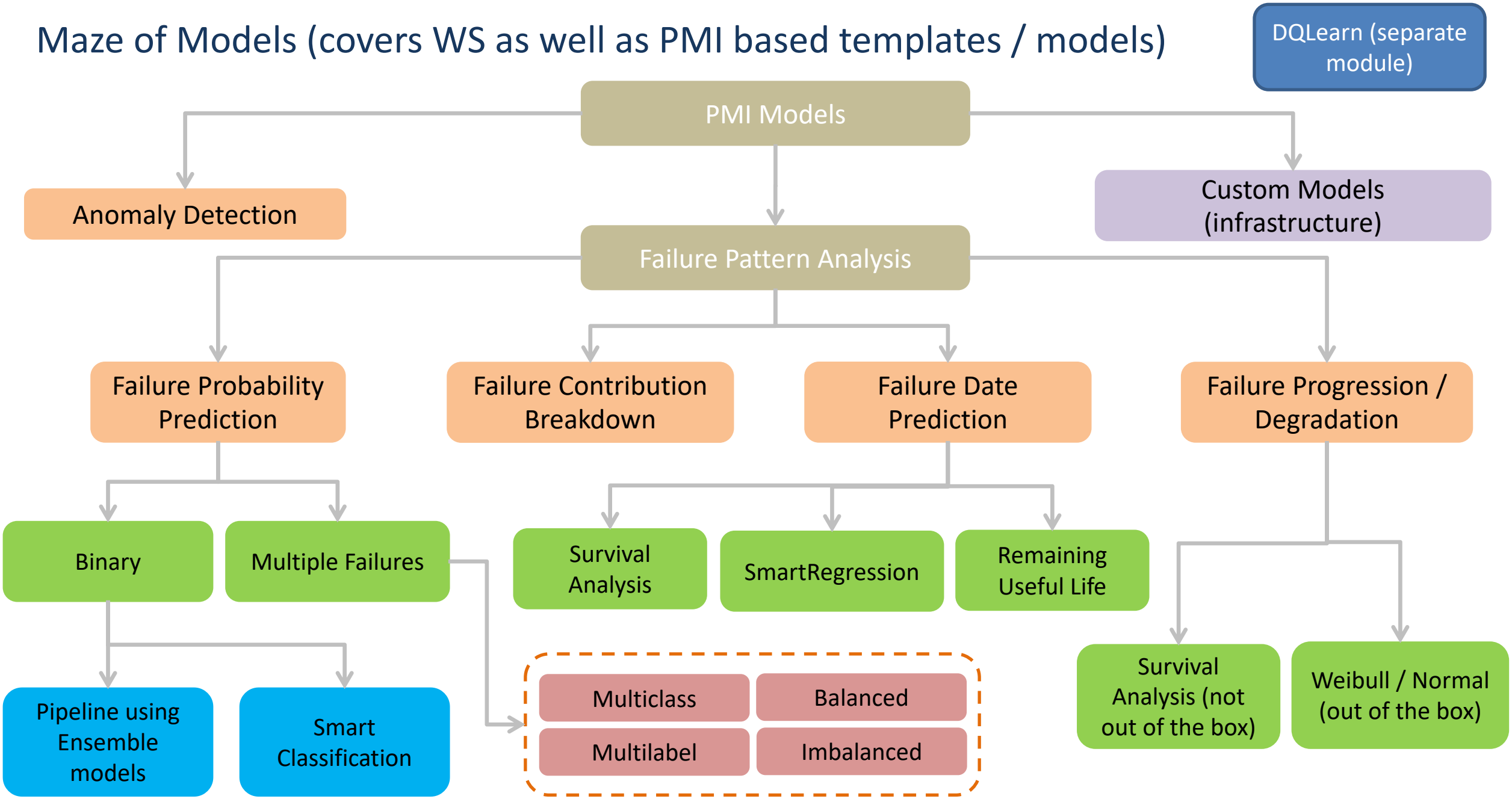
Maximo Application Suite **Health & Predict** - External Engine for E&U



Contributing factors for Health score calculation



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

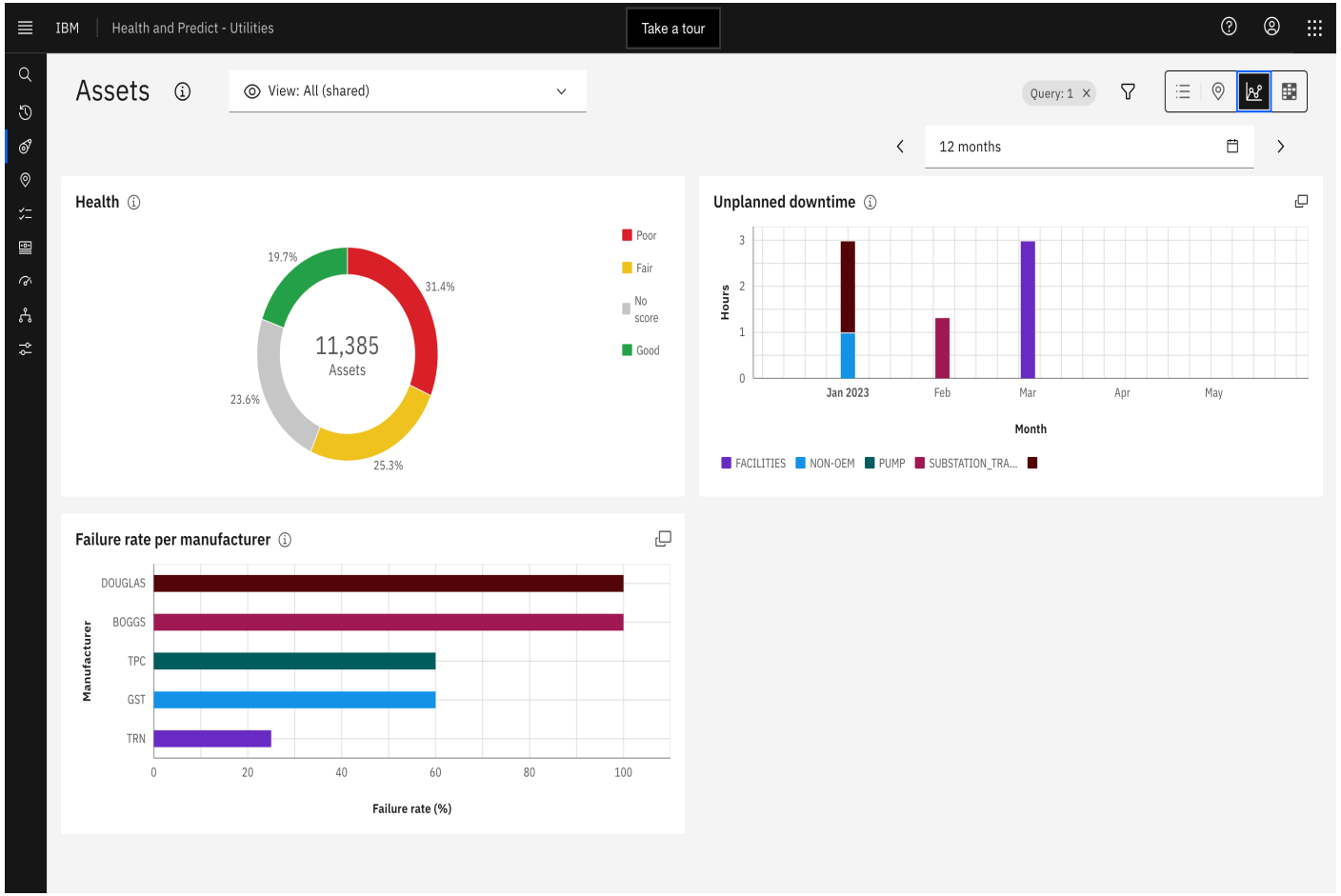


Understanding performance using charts

Value

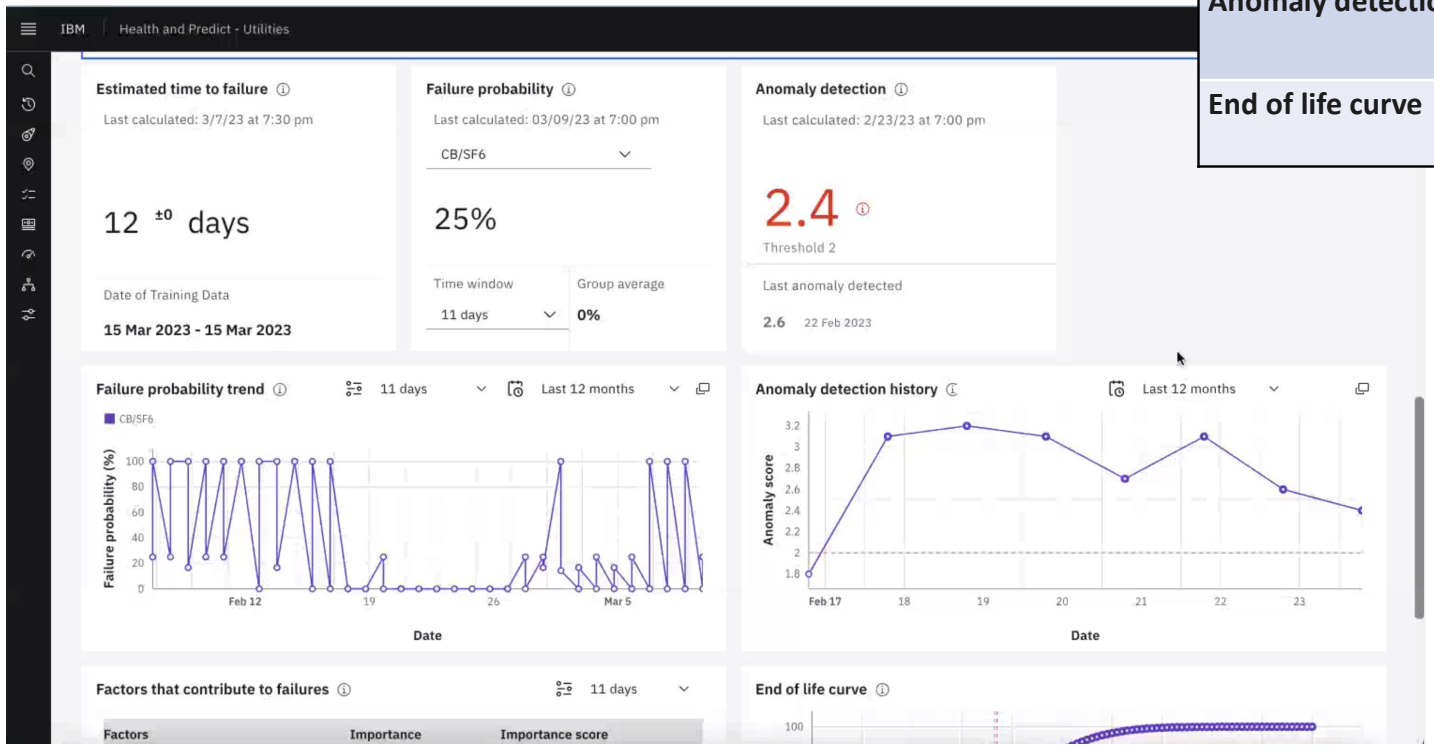
View key KPI summaries for a group of assets to help identify assets that might require investigation

- **Health** – understand the distribution of assets based on their health score
- **Unplanned downtime** – understand the total unplanned downtime hours over a selected date range for each asset type
- **Failure rate per manufacturer –** view top manufacturers with highest failure rate



Predict and Forecast Asset Performance and Anomalies

Type	Business Objective
Probability of failure	What is likelihood of failure within a time period. How much risk is there? Statistical health score of assets. Preventive maintenance to prolong life, plan for capital replacement, set forecasts.
Predicted failure date	Incorporate modeled failure data to guide scheduling decisions for repair and replacement.
Anomaly detection	Identify situations where operations are abnormal and may impact production, asset performance, quota, etc.
End of life curve	Forecast remaining useful life of the asset to aid capital replenishment strategy.

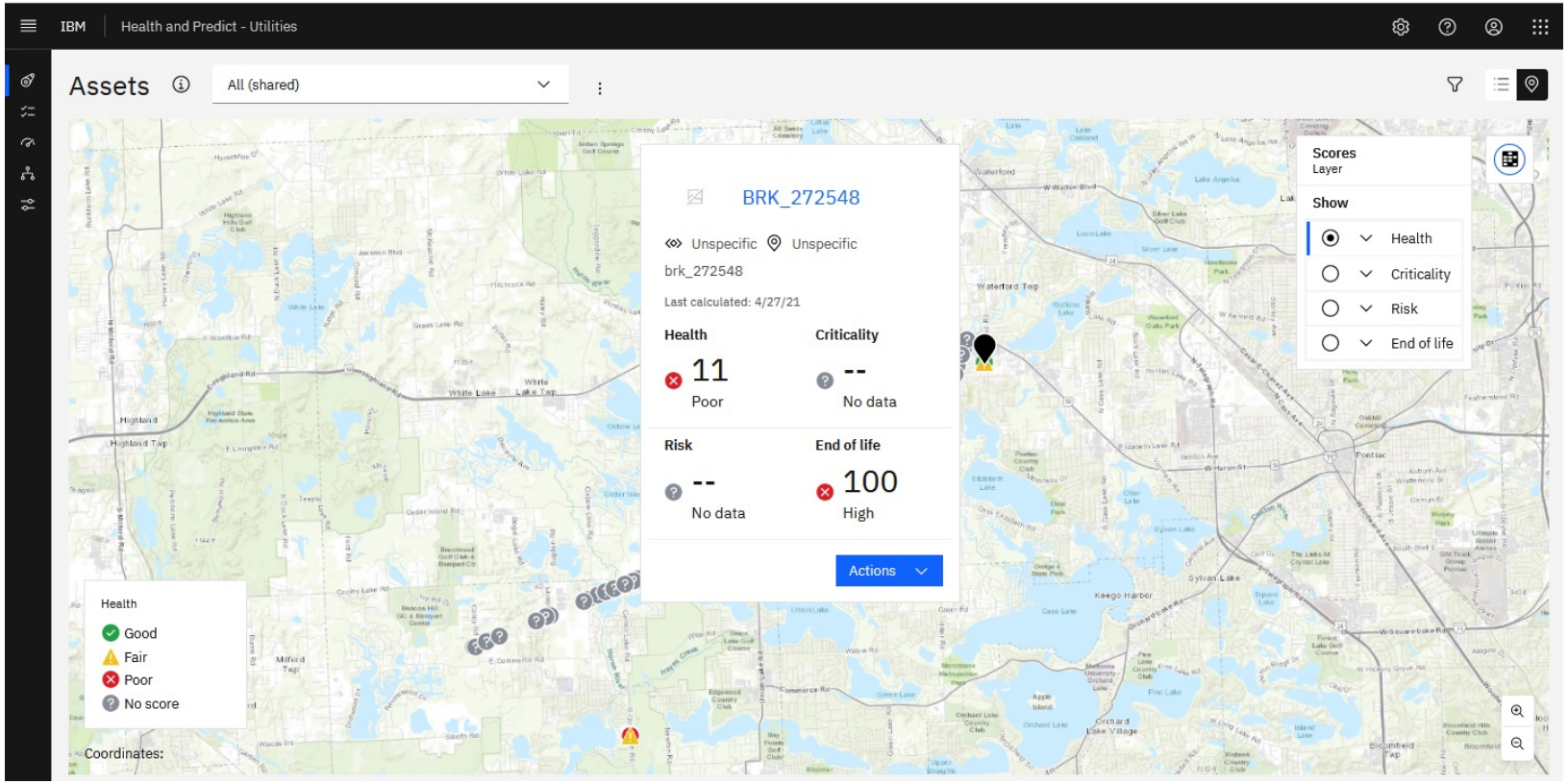


Out of the box algorithms & templates for health and predictions must be trained for your specific asset classes and assets

MAS 8.x Map view

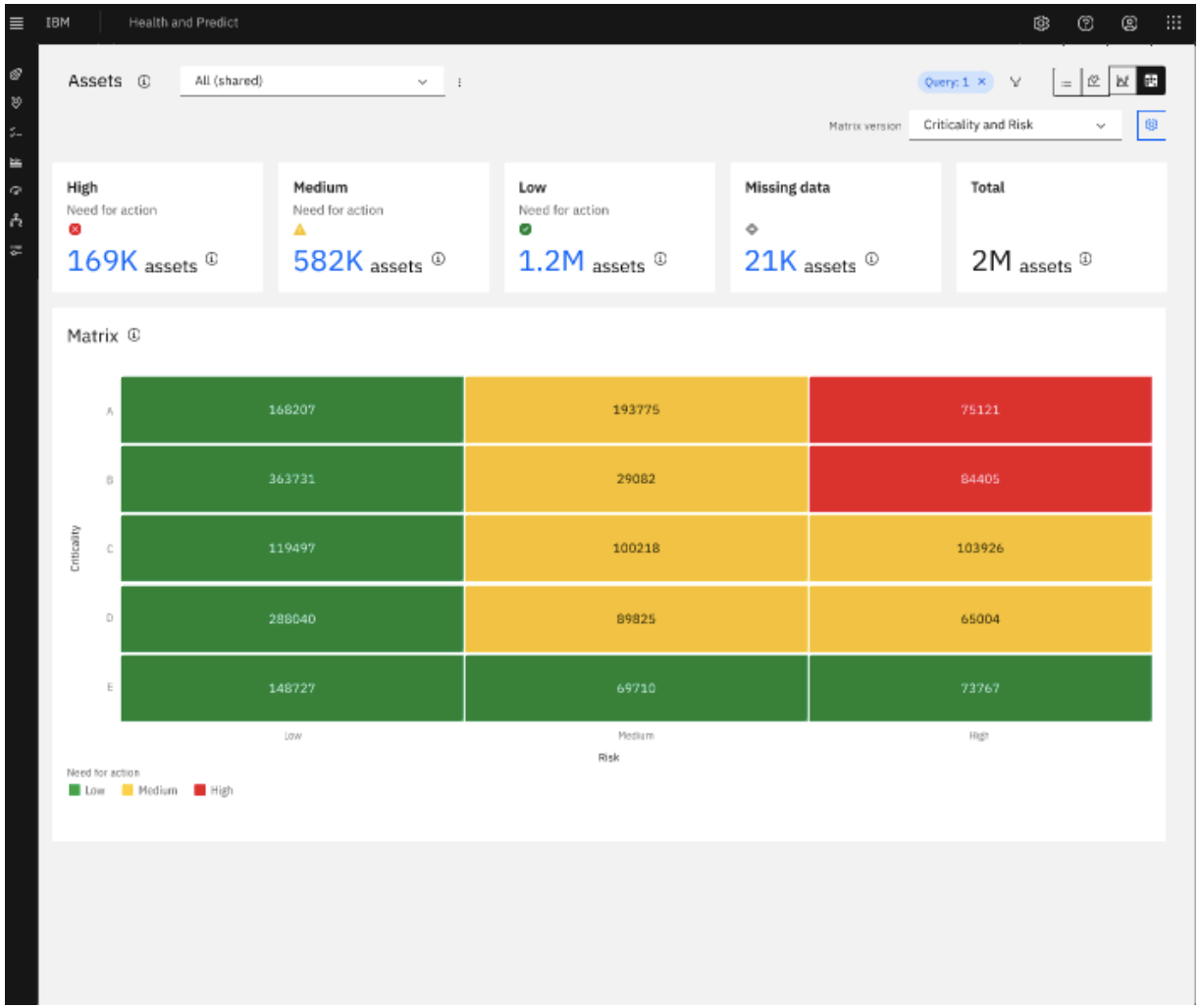
- Provides full screen geographical view of the performance of assets
- Allows the reliability engineer to see if there are pockets of risk in certain areas

- 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



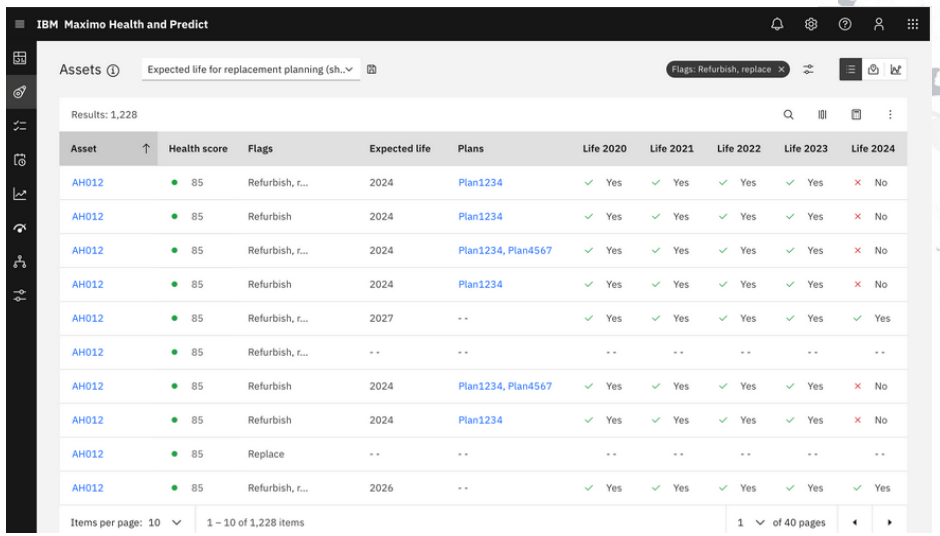
Asset Risk -- Multi-asset analysis

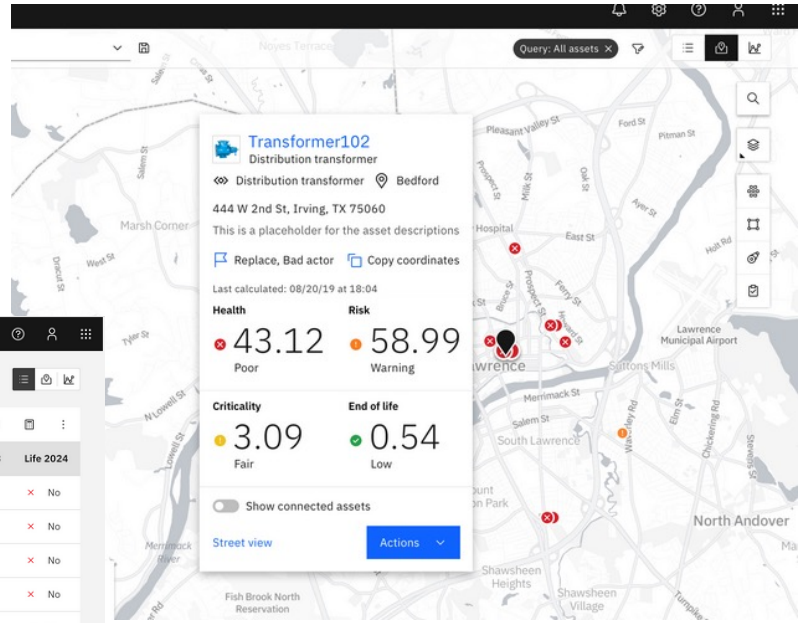
- Use matrix to identify assets in need
  - Criticality and risk
  - Criticality and health
  - Criticality and end of life
- Drill down into the list of assets in each category
- Add assets to a csv file or an asset investment project
- Take action on assets at risk



Maximo Application Suite v8.x **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



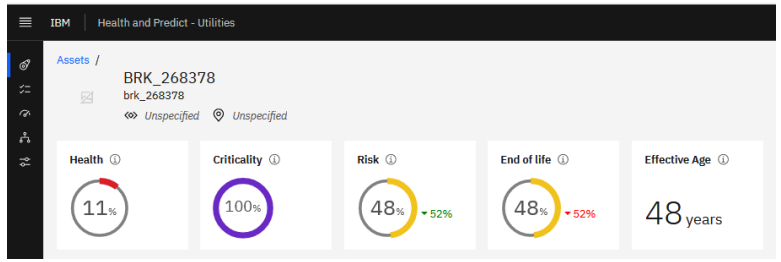


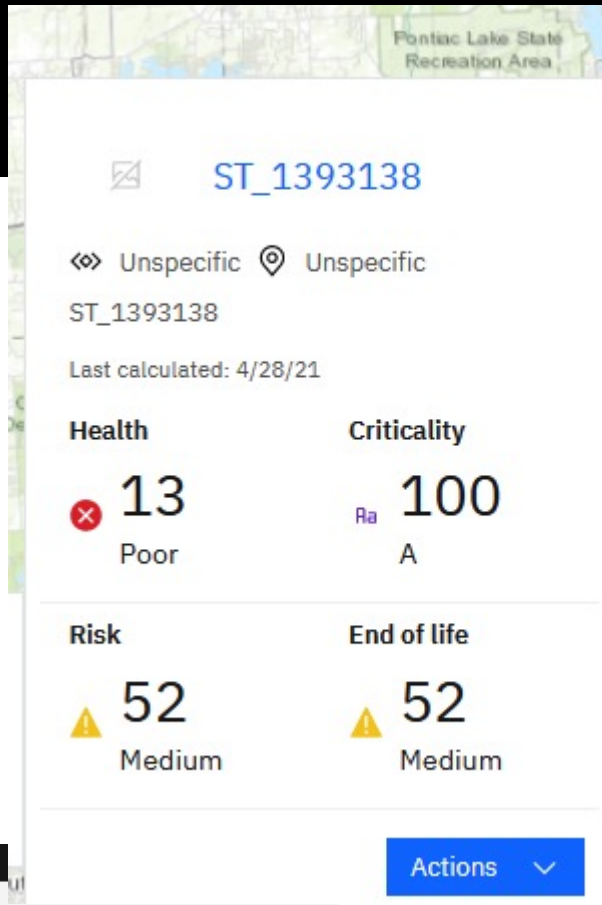
MAS 8.x Scoring Model extensions

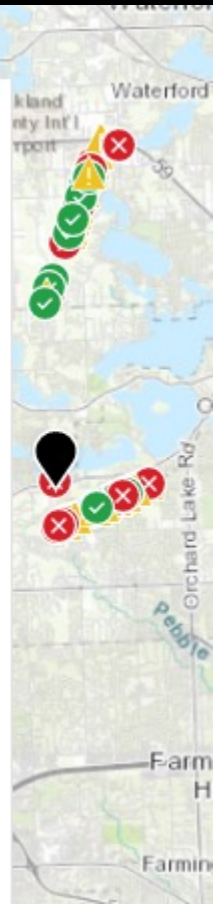
- Provides support for BCTC scoring methodology for Transmission & Distribution Assets
- Leverages pre-defined python scoring models in Watson Studio Notebooks

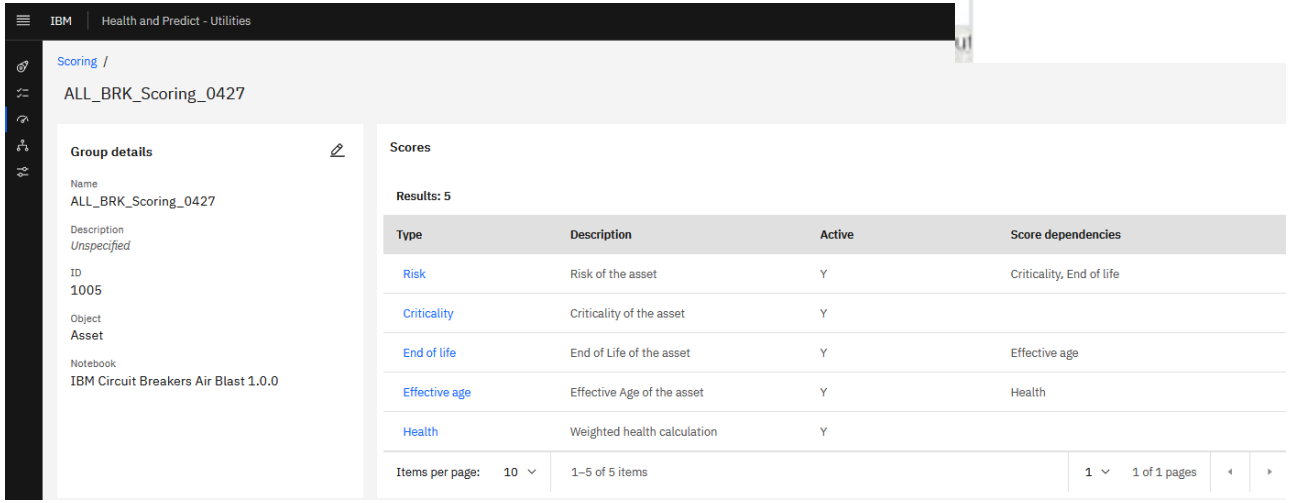
Includes 2 new score types
- Effective Age (based on Health)
- End of Life probability based on degradation curve

Enables ability to use one score type within another
- Effective Age based on Health
- Risk based on Criticality and EOL probability









CP4D Update – Introduction

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

Watson Anywhere

Cloud Pak for Data

A one of a kind, pre-integrated set of data and AI services delivered within an open and extensive cloud native platform

Collect Data | Organize Data | Analyze Data | Infuse AI

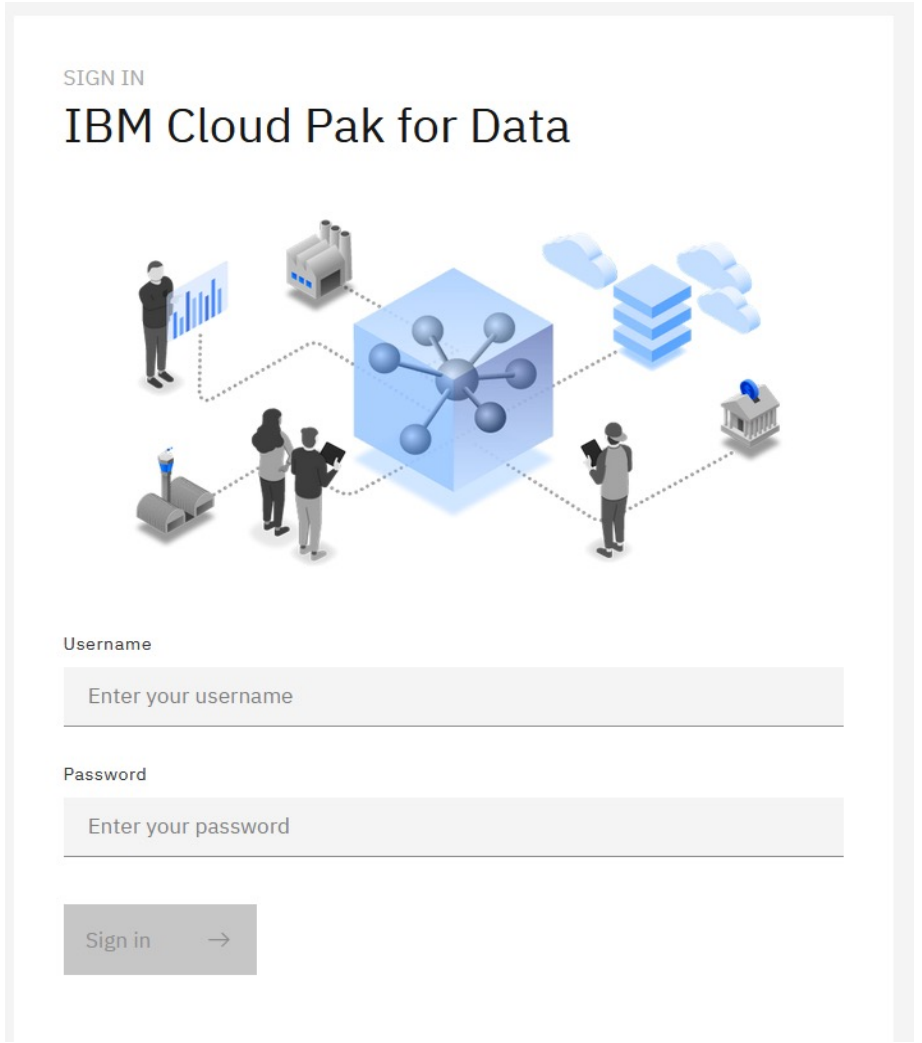
Red Hat OpenShift Container Platform

Cloud-native container platform and operational services

Red Hat OpenShift

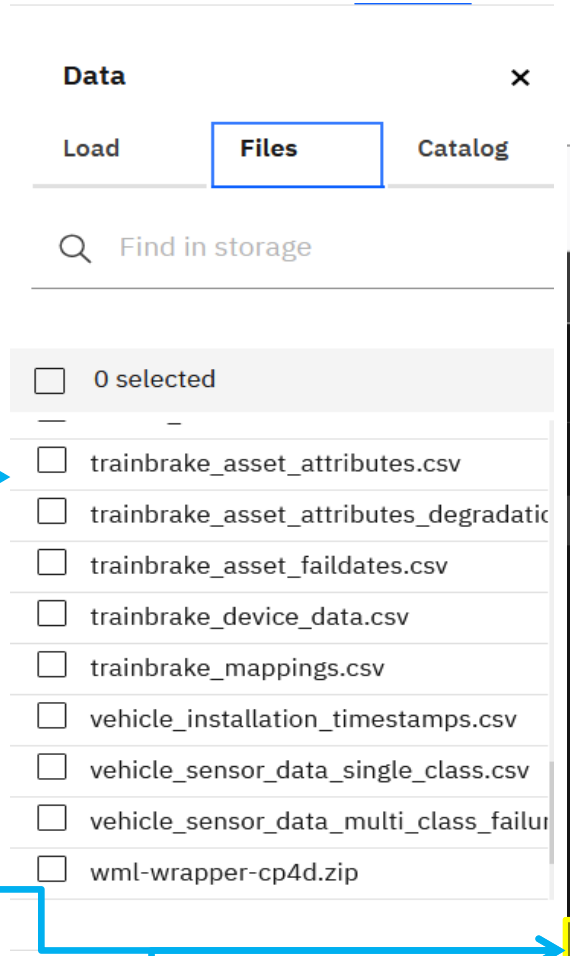
IBM Cloud | OPENSHIFT | aws | amazon web services | Azure | openstack | Google Cloud

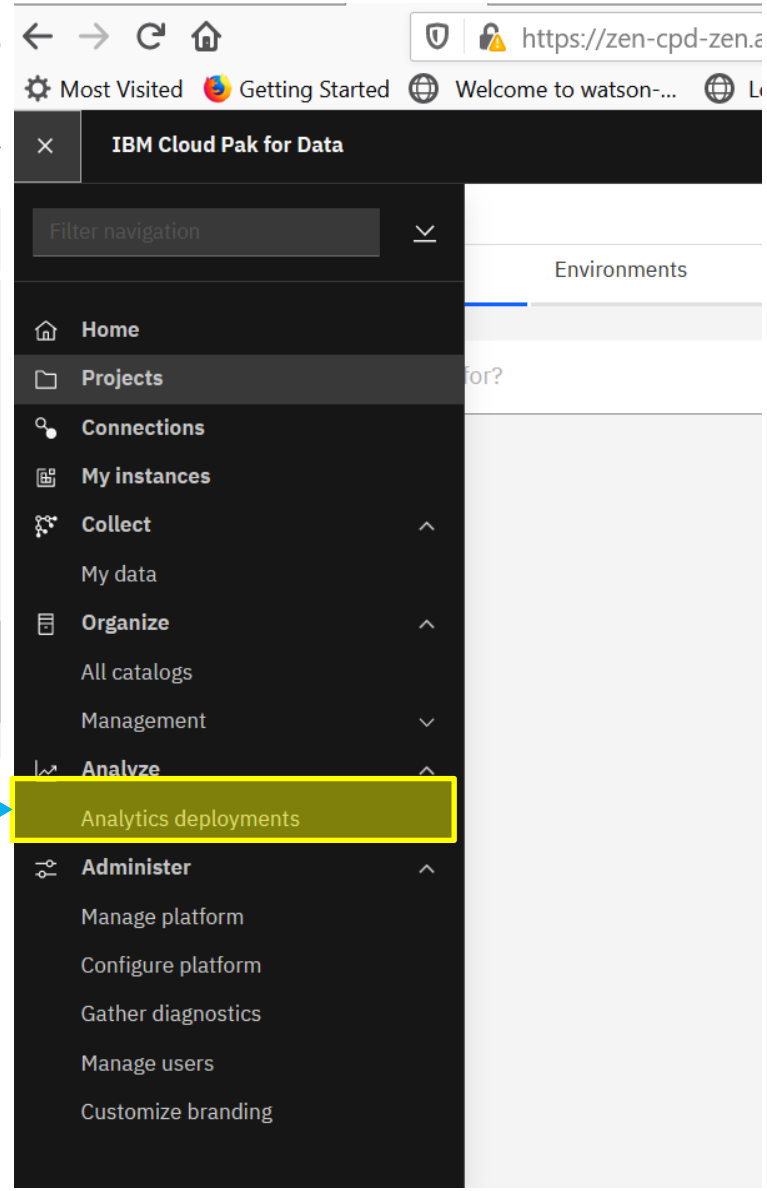
Hyperconverged
Private Cloud System



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

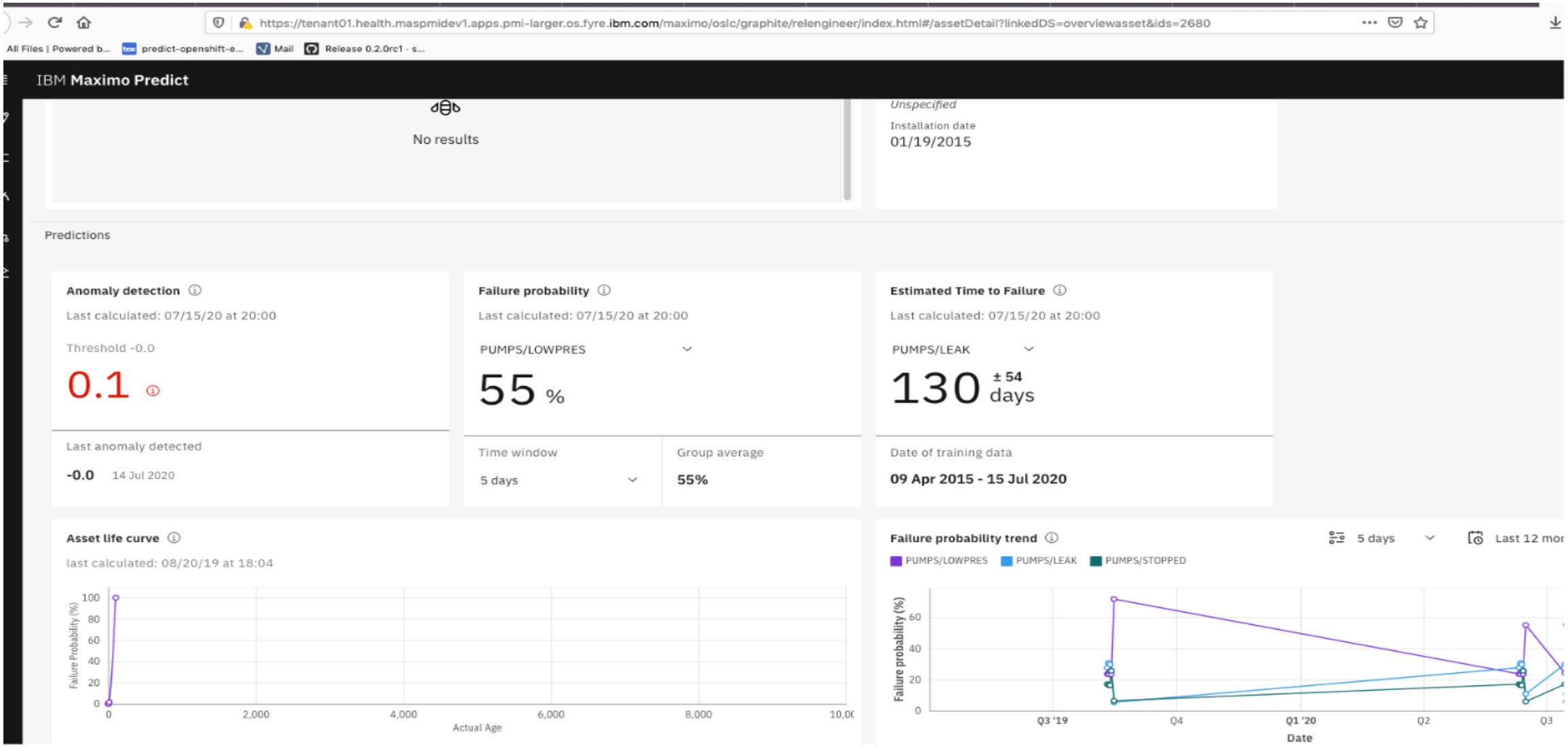




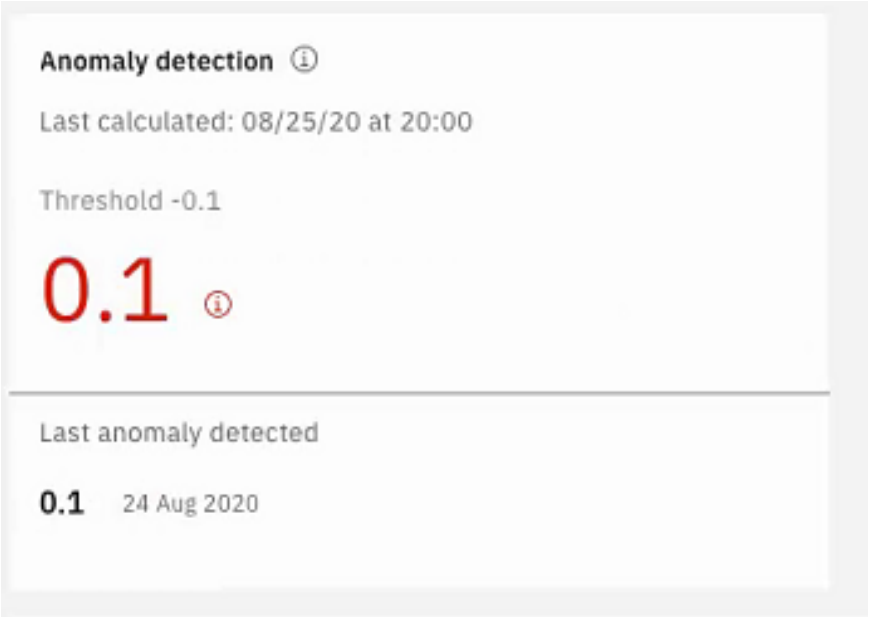

```
import os
execution_env = os.environ.get('AX_EXT_DEPLOYMENT_TYPE', 'wscloud') #'icp alternatively cloud.
store_artifacts = False
bucket_name_for_pickled_models = 'maximo-predict-pickled-models' # Keep in mind that neither
cters in the name, or only those allowed (hyphen or dash is allowed by ICOS)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

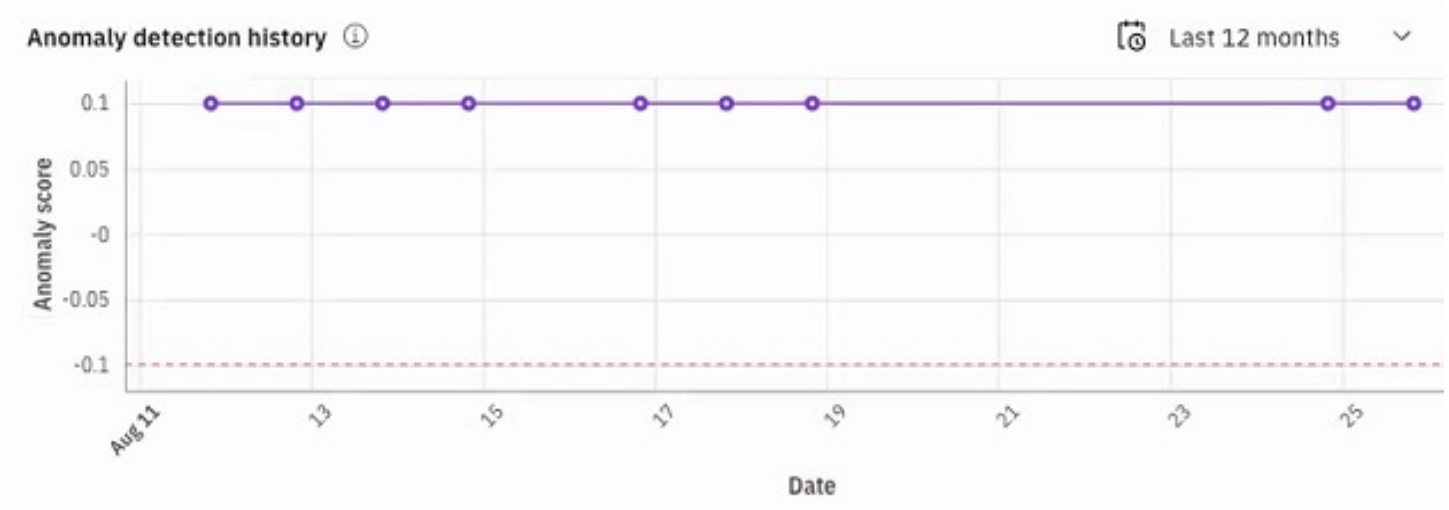
UI flows and insights

RE Dashboard

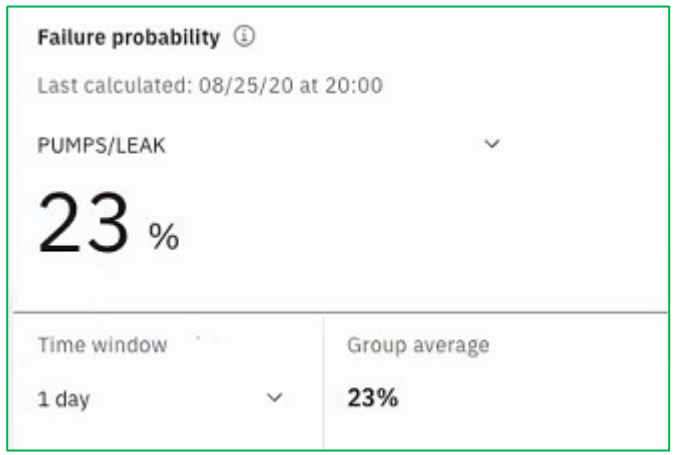


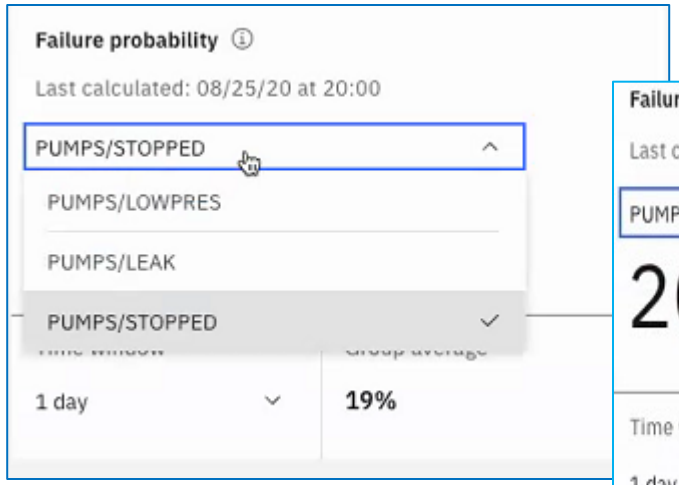
Anomaly Detection

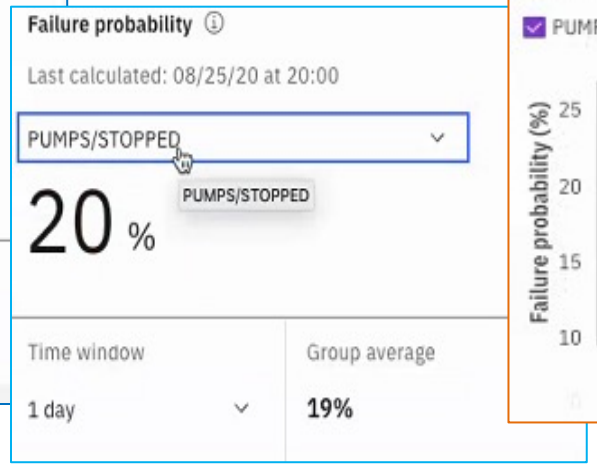




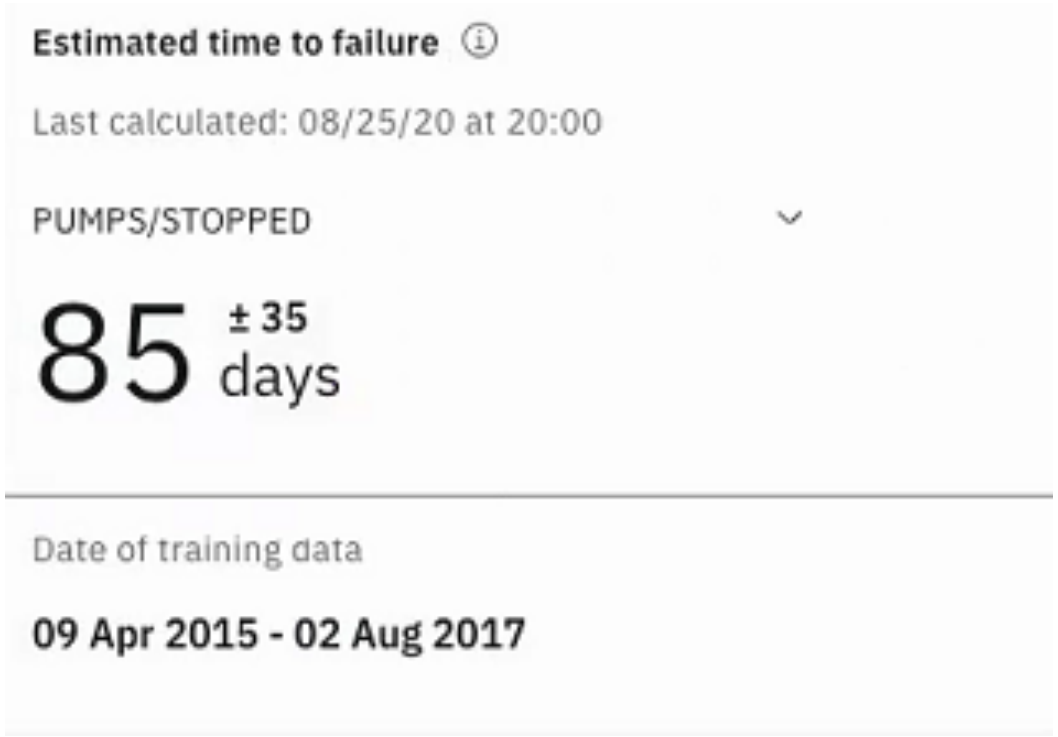
Failure Probability Prediction

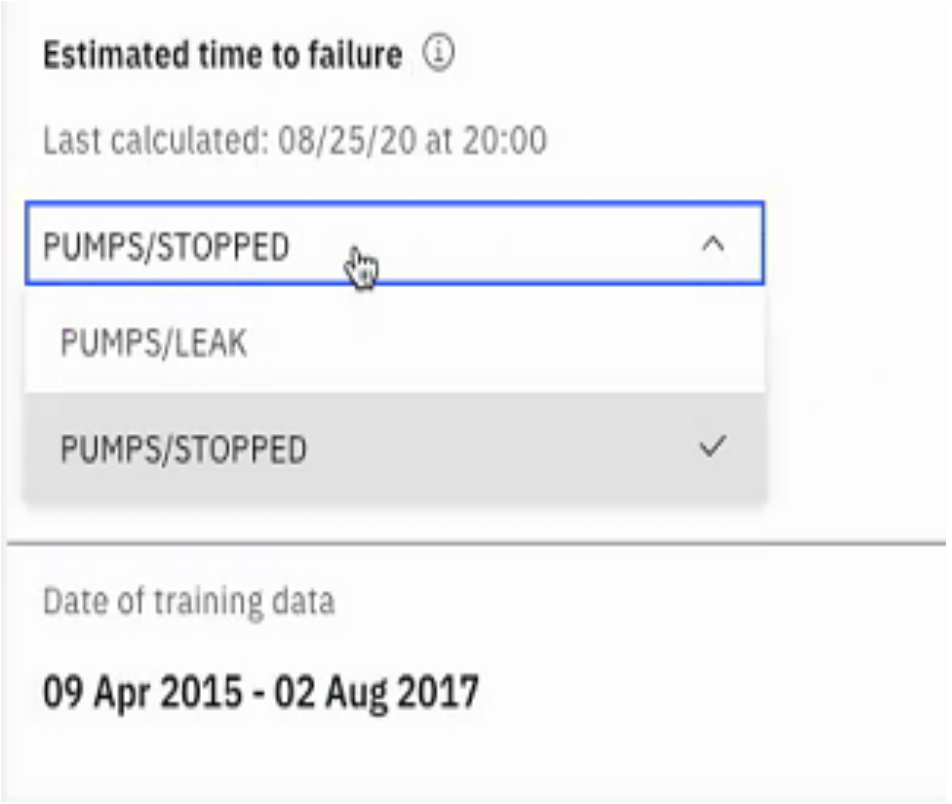






Predicted Failure Date / Estimated Time to Failure





Failure Contribution Breakdown

Factors that contribute to failures ⓘ

1 day

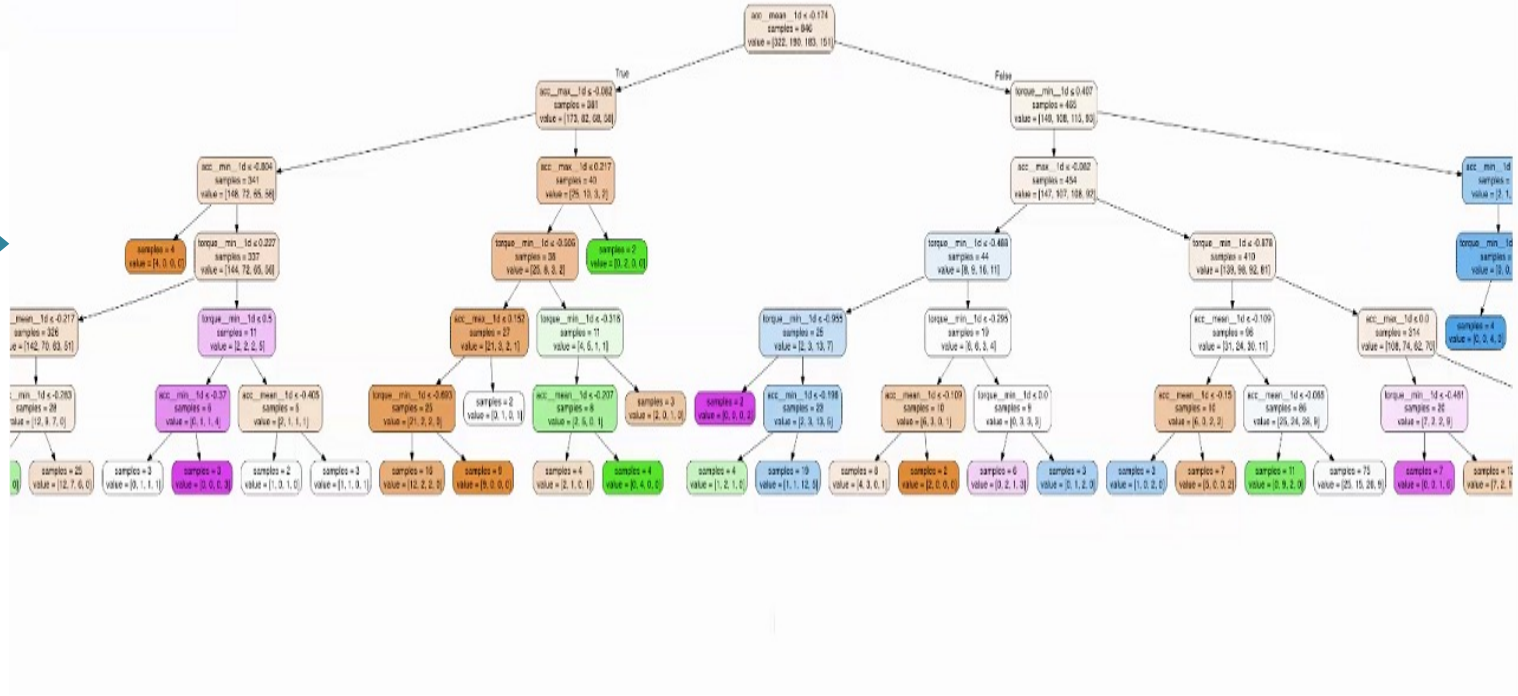
Factors	Importance	Importance score
torque__min__1d	High	0.41156
acc__mean__1d	High	0.31813
acc__max__1d	Moderate	0.16503
acc__min__1d	Moderate	0.10528

View full analysis tree

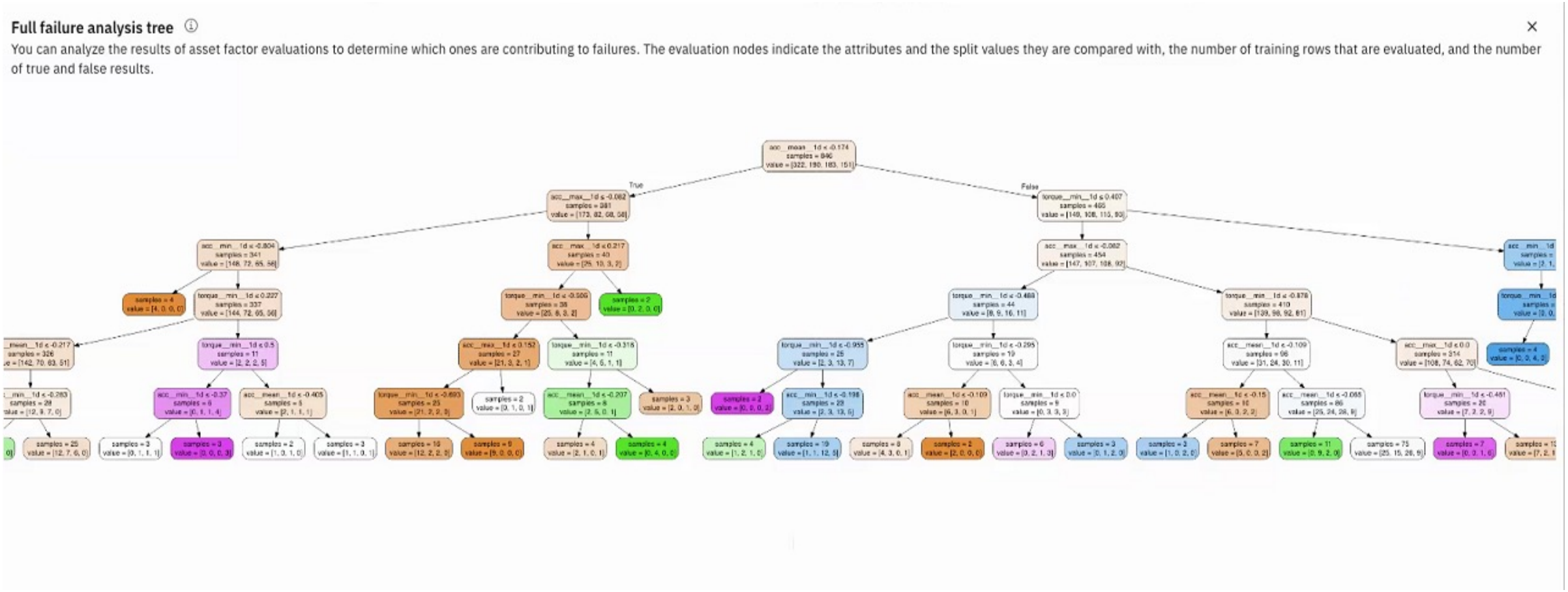
Full failure analysis tree ⓘ

X

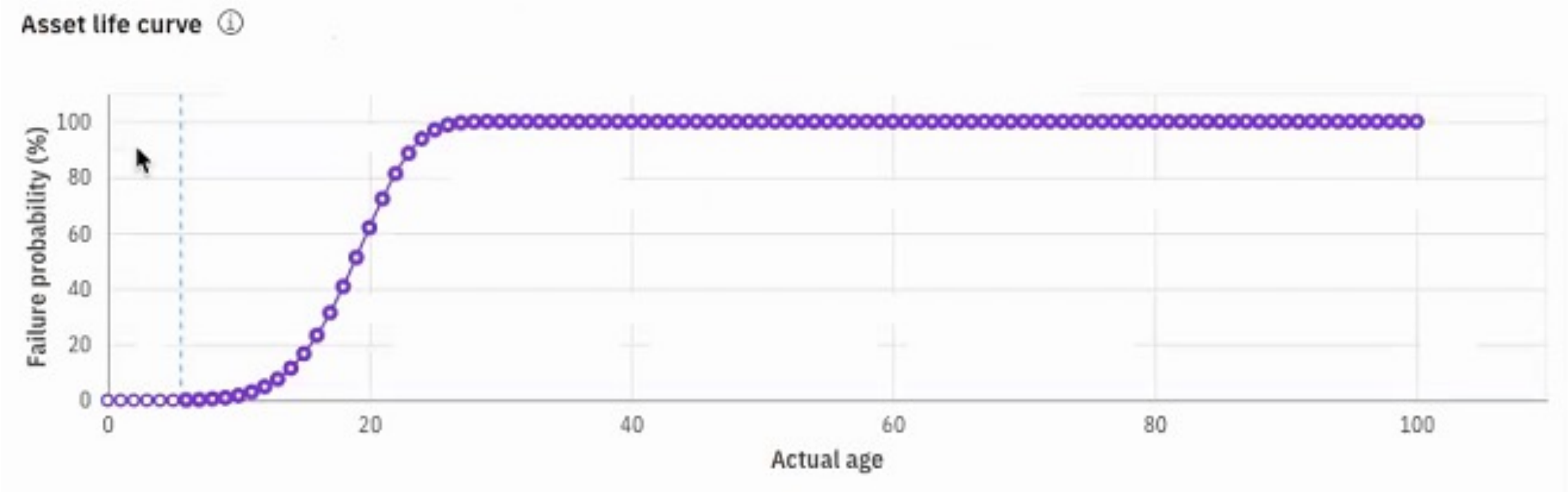
You can analyze the results of asset factor evaluations to determine which ones are contributing to failures. The evaluation nodes indicate the attributes and the split values they are compared with, the number of training rows that are evaluated, and the number of true and false results.



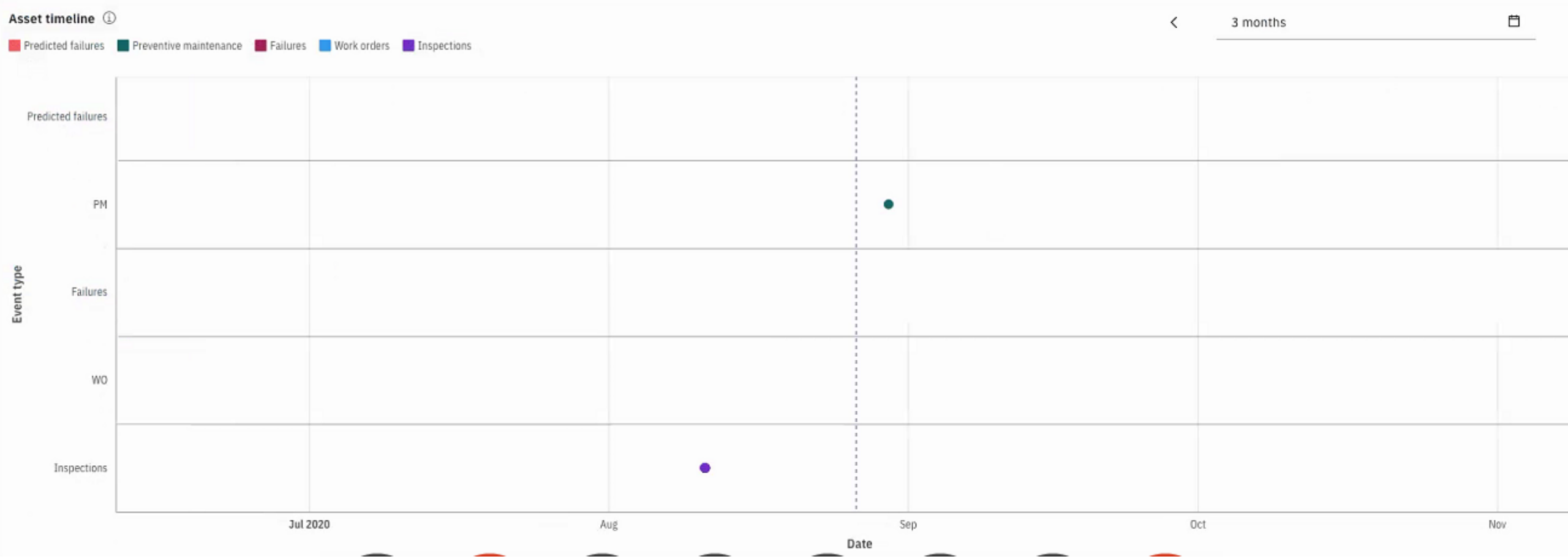
Failure Analysis Tree



Asset Life Curve



Asset Timeline



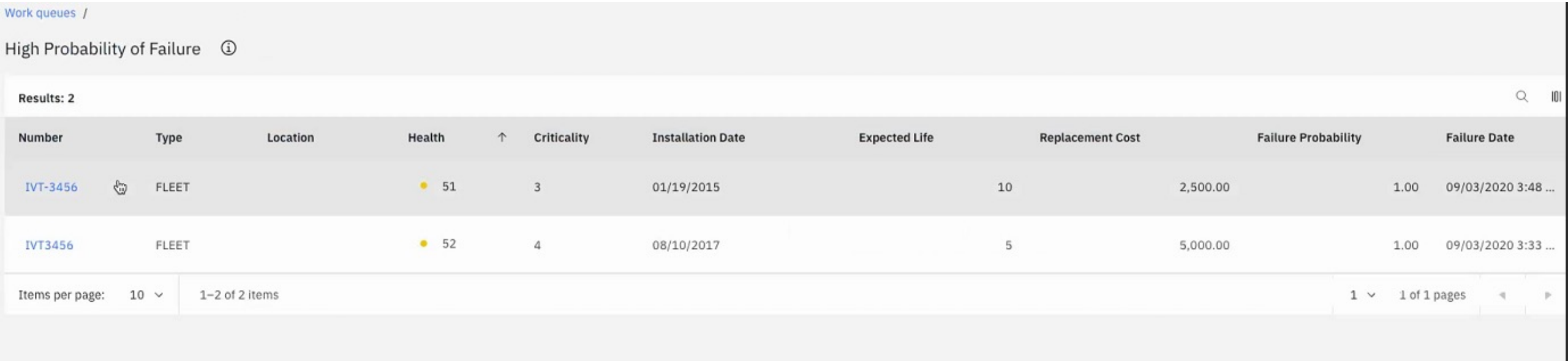
Work Queues

Work queues

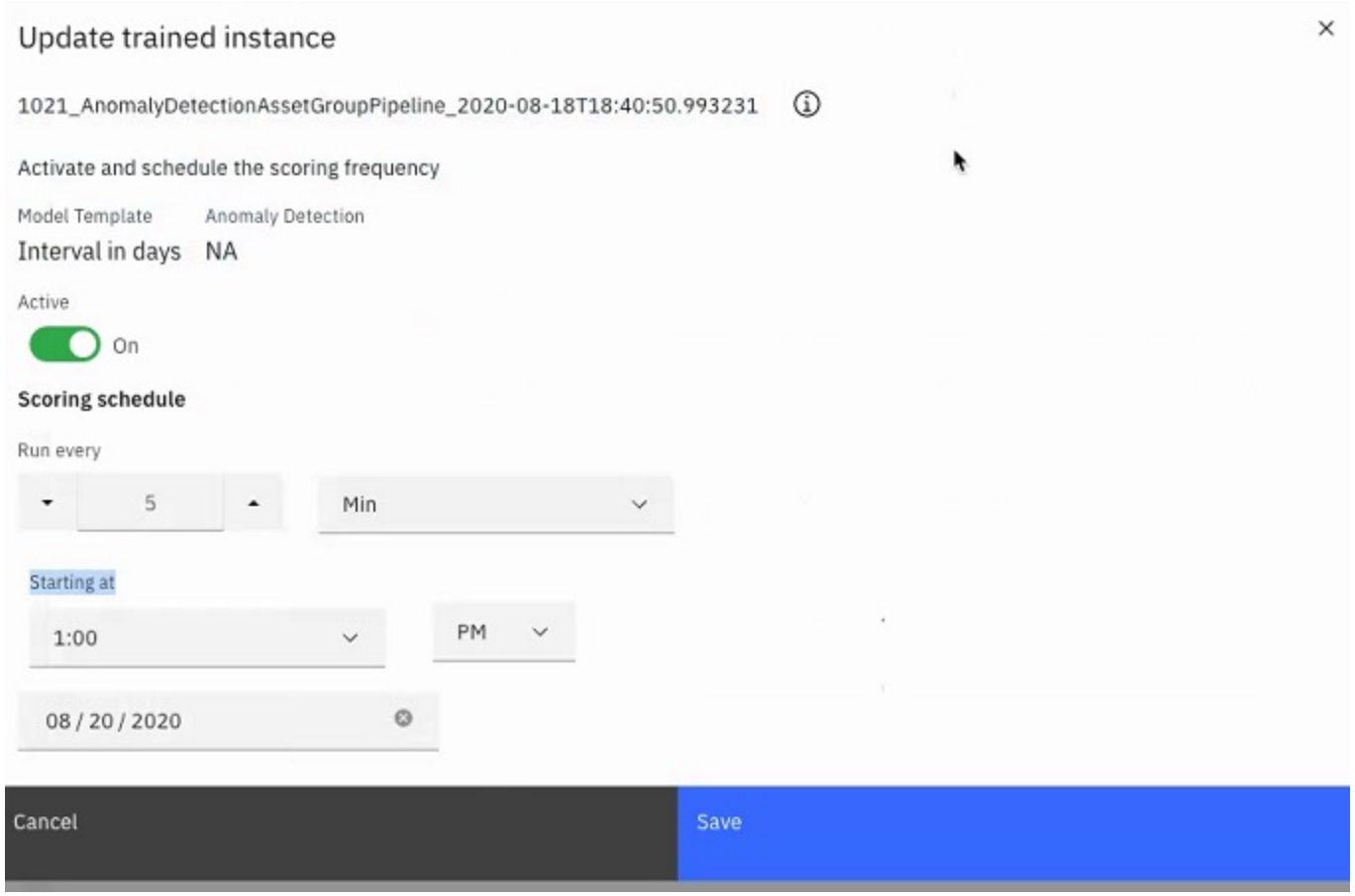
Refresh queues

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...

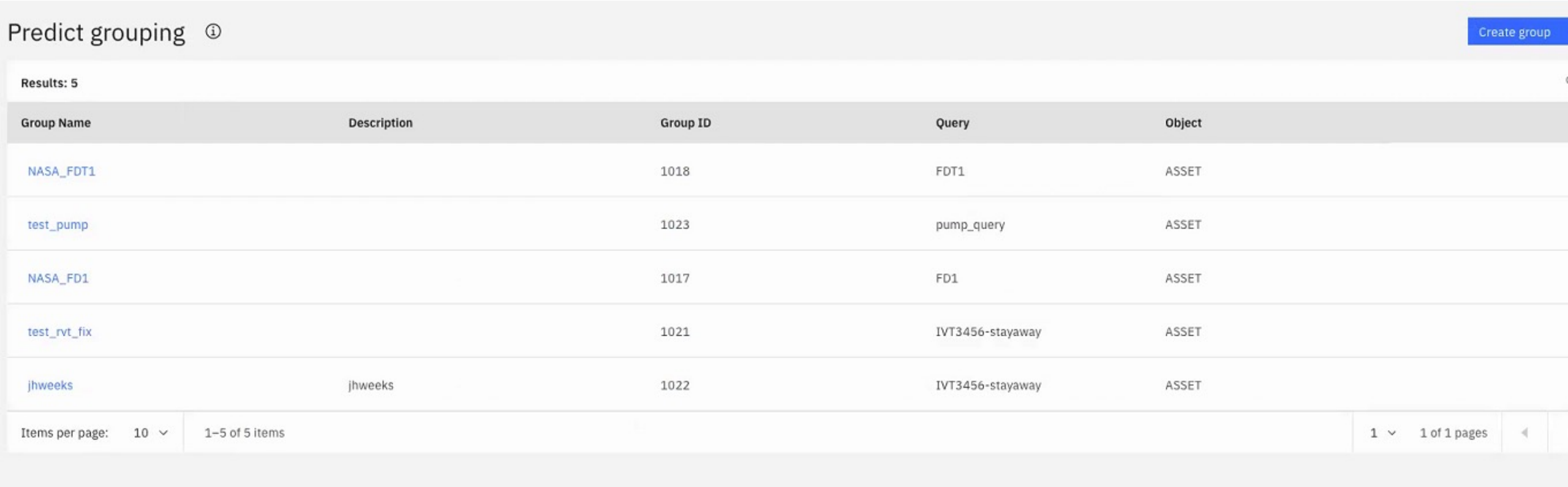
Work Queue – assets prone to high failure probability



Trained model scheduling



Predict Grouping



Workflow

Model / Template Specifics

MAS Predict

Train & Register Models using PMI Pipeline on Watson Studio

Connect to Maximo – Predict

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

Install Maximo Predict SDK

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

Configure the pipeline

```
In [3]: from pmlib.failure_prediction import FailurePredictionAssetGroupPipeline

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

Train the model

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

Register the model

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

MAS Predict: Anomaly Detection

Model Description

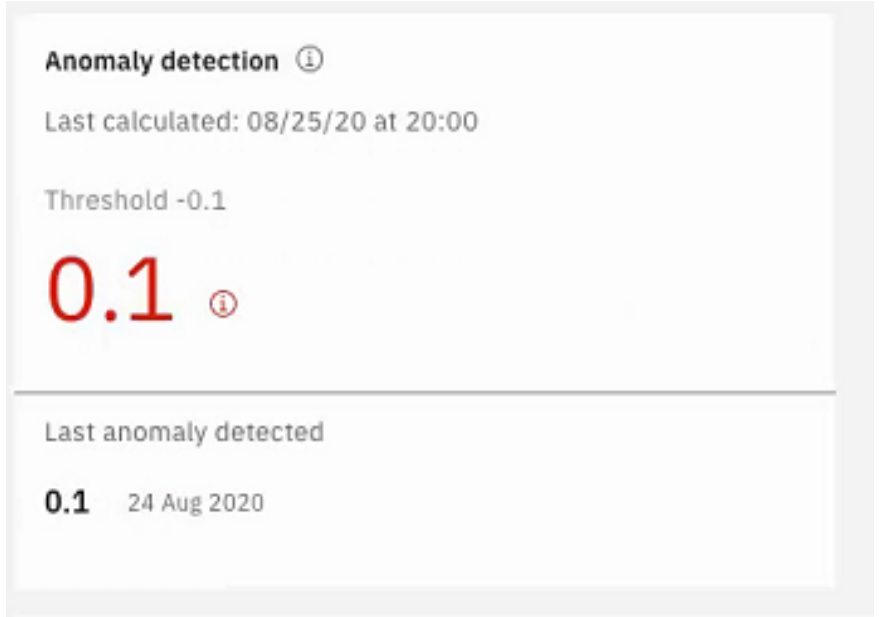
- Detect anomalies in data
- Leveraged with sparse failure event data

Inputs Required

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

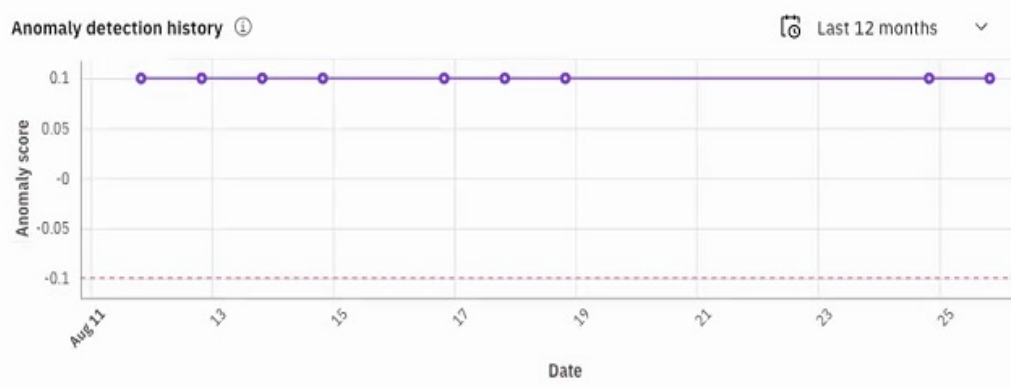
Questions Answered

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



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



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

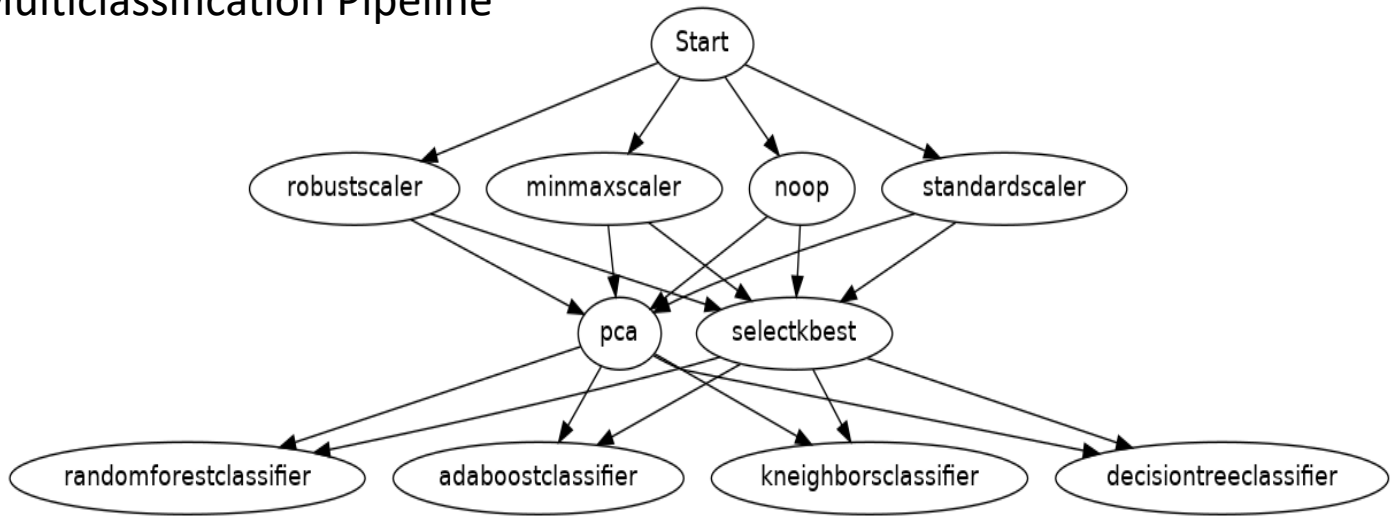
Failure Probability Prediction in a given temporal window

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

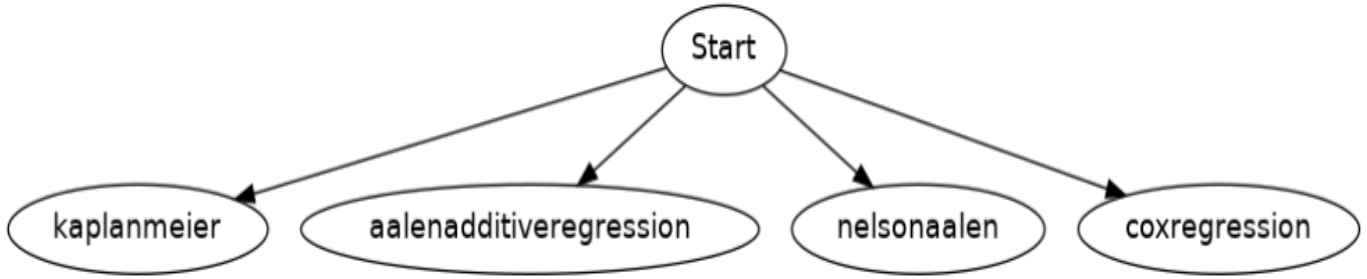
Inputs Required

- 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()),
    ("decisiontreeregressor", DecisionTreeRegressor()),
    ("mlpregressor", MLPRegressor()),
    ("gradientboostingregressor", GradientBoostingRegressor()),
    ("adaboostregressor", AdaBoostRegressor()),
    ("baggingregressor", BaggingRegressor()),
    ("randomforestregressor", 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

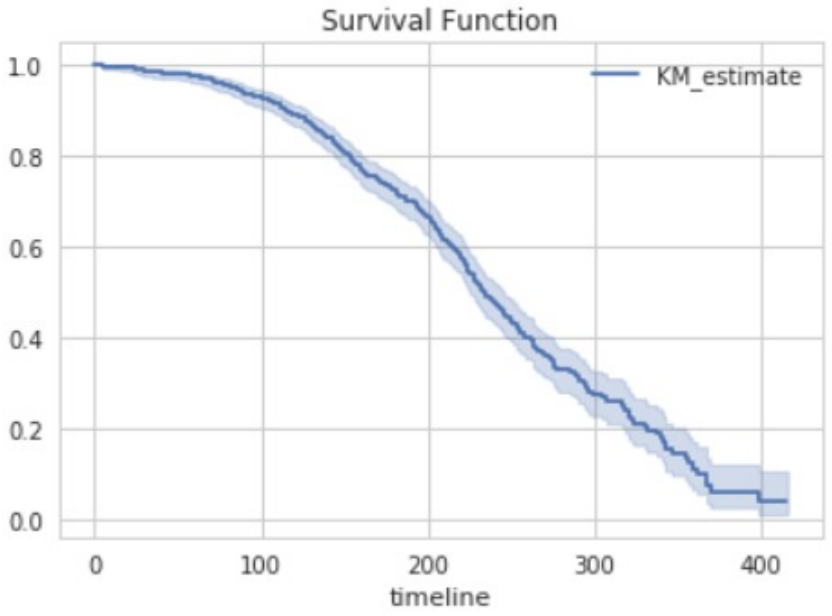
Sensor Data

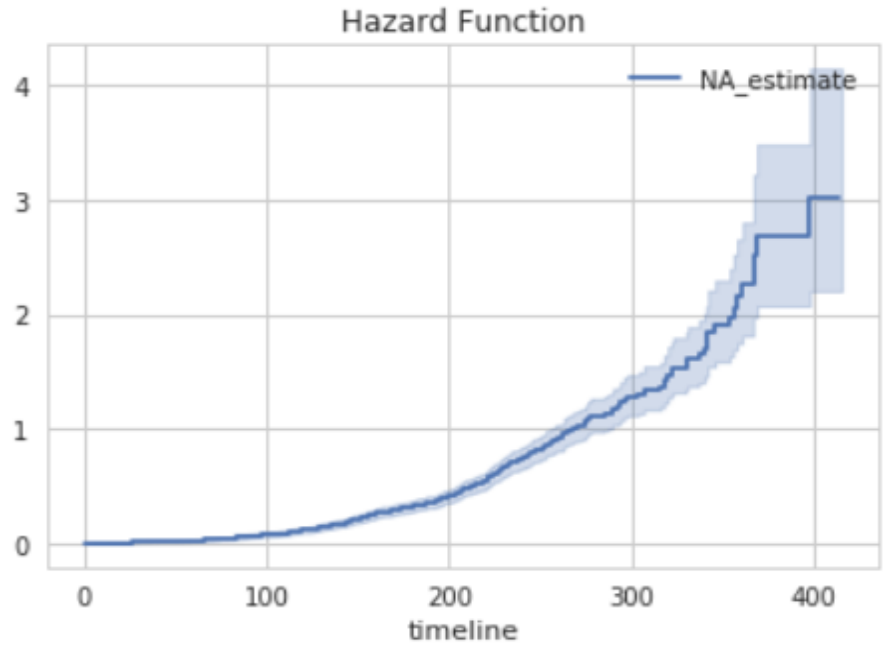
asset_id (string)	timestamp (UTC recommended)	variable 1 (float)	variable 2 (float)	variable 3 (float)	variable n (float)
P-0983	2006-08-27 19:56:41	73.54	22.56	38.54	39.392
A-3748	2006-08-28 11:32:17	80.44	-26.384	43.965	7.493

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

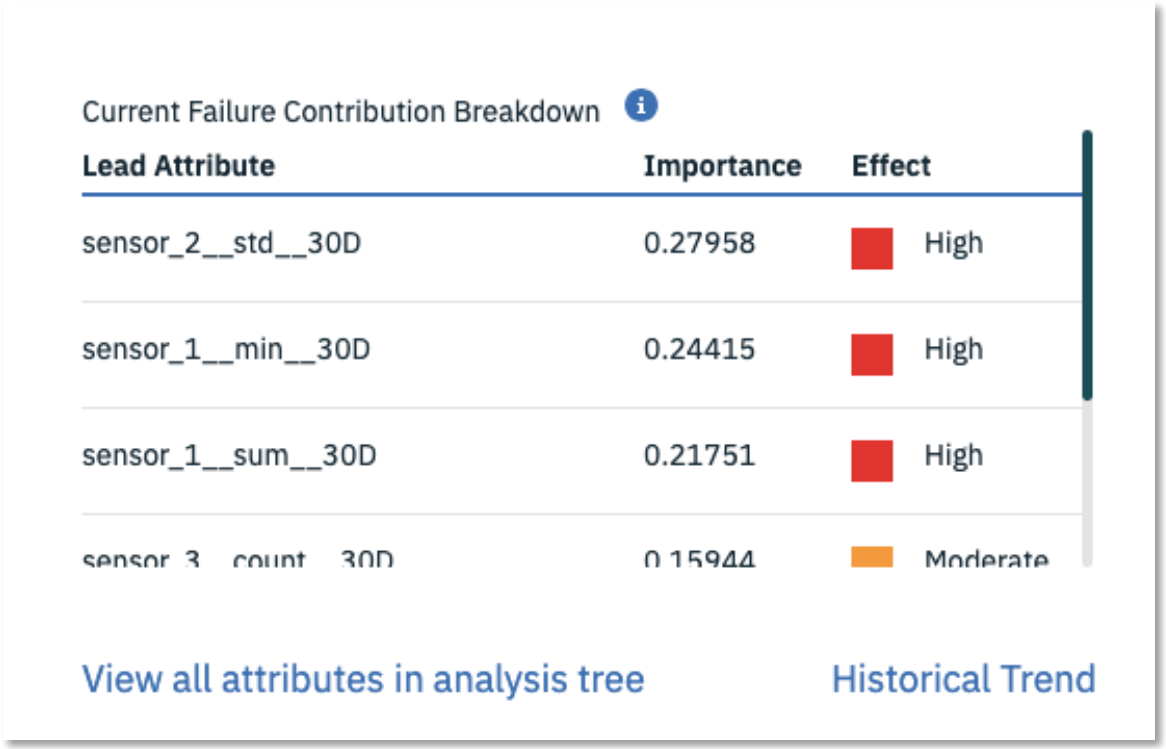
- Root cause analysis of bad outcome

Inputs Required

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

Questions Answered

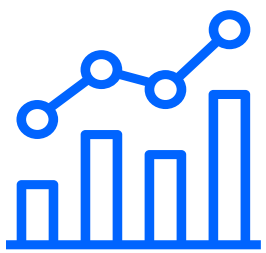
- What are the top features that contribute to failure?



Failure Probability Curve

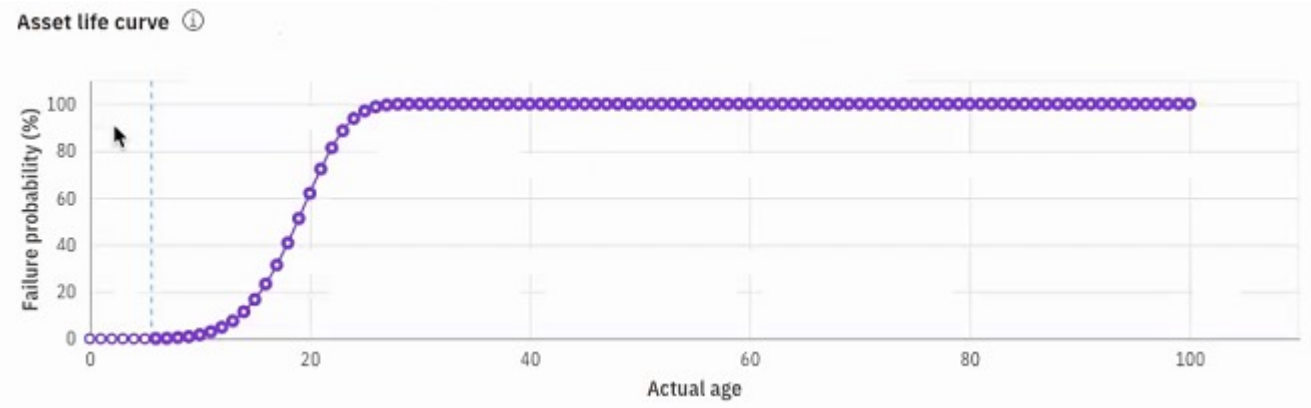
- Statistically, to evaluate mean life of assets the sample mean or the average age method is acceptable if a big population has end-of-life information. But asset such as generators, transformers, reactors, cables, etc. have a relatively long life up to and even beyond 40 years and generally there are 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**: $\{f(x;\lambda,k)=k\lambda(x\lambda)k-1e-(x/\lambda)kx\geq 0,\ 0x<0,$

where k > 0 is the shape parameter and λ > 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': [':faildate'],
    'predictions': ['anomaly_score', 'anomaly_threshold'],
    'pre_failure_window_size': 20,
    'pre_failure_failure_size': 10,
    'srom_training_options': {
        'exectype': 'spark_node_random_search'
    }
})
```

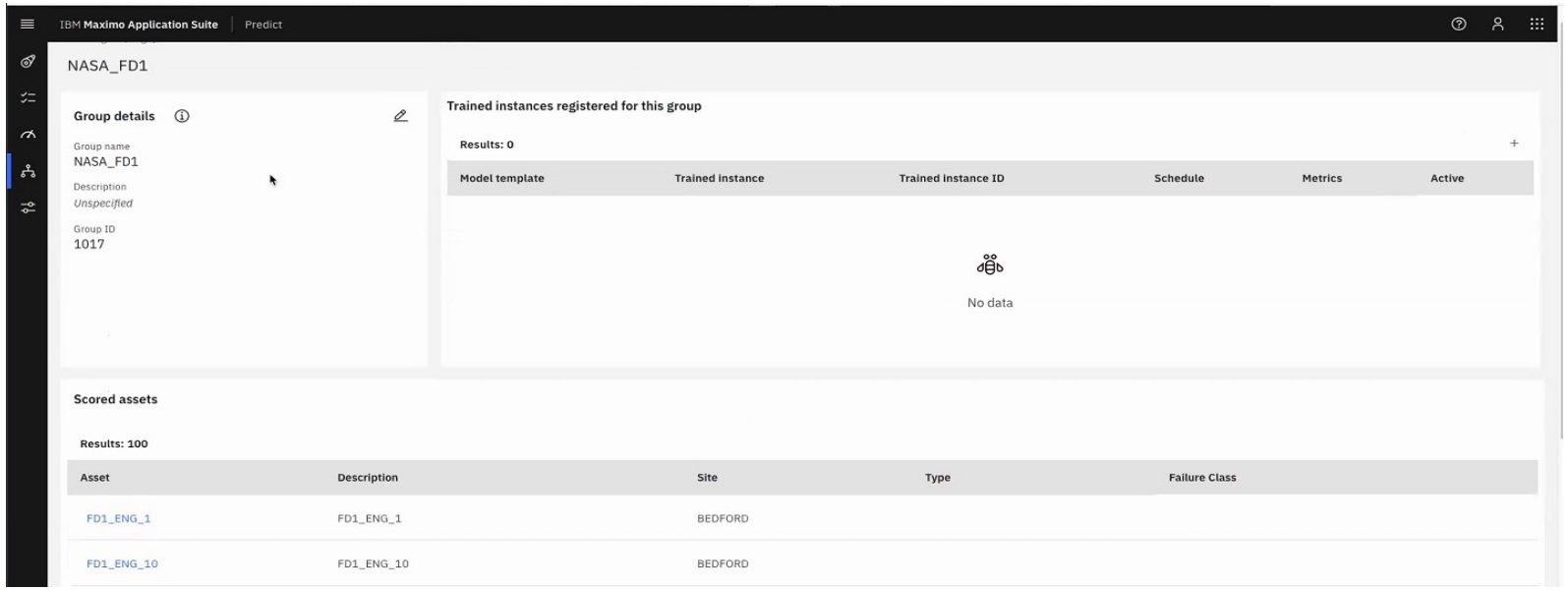
```
TimeToFailureAssetGroupPipeline(
asset_group_id='ID of an asset group',
model_pipeline={
    'features': ['DeviceTypeOne:temperature', 'DeviceTypeTwo:Humidity']
    'features_for_training': [':installdate', ':faildate'],
    'predictions': ['predicted_time_to_failure'],
    'srom_training_options':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

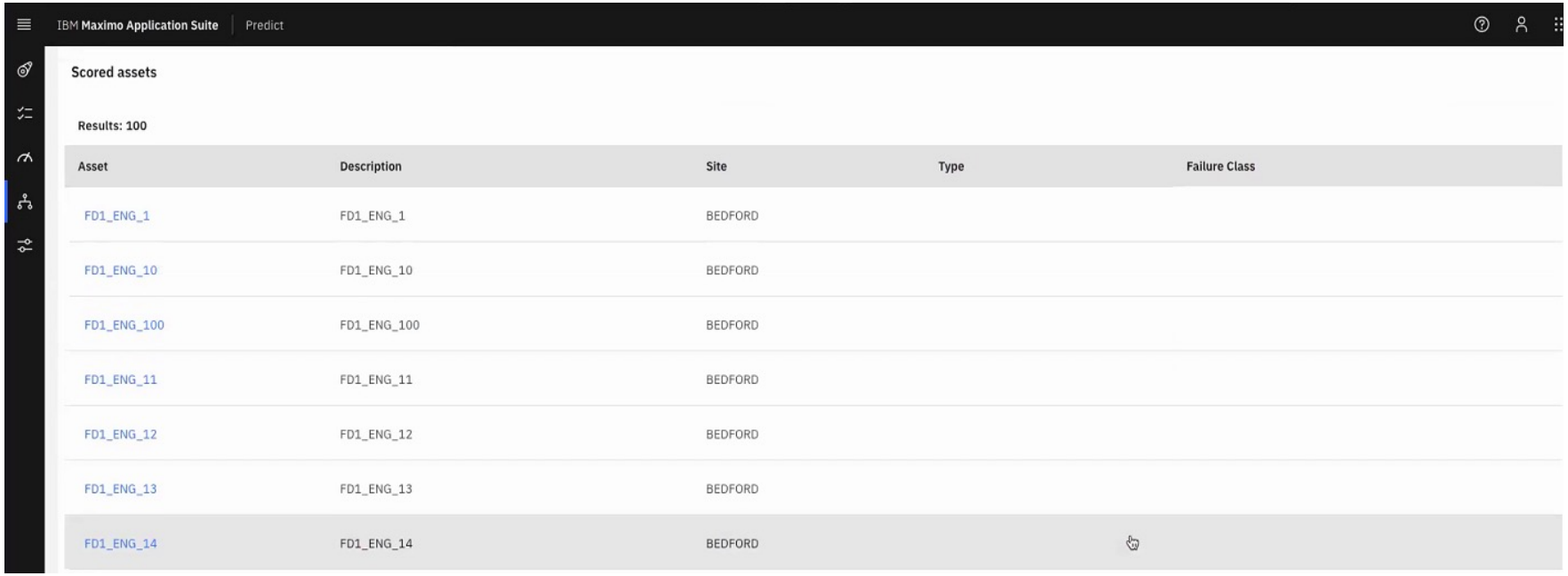
Additional Examples of PMI Pipeline definition

```
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': {
            'exectype': '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

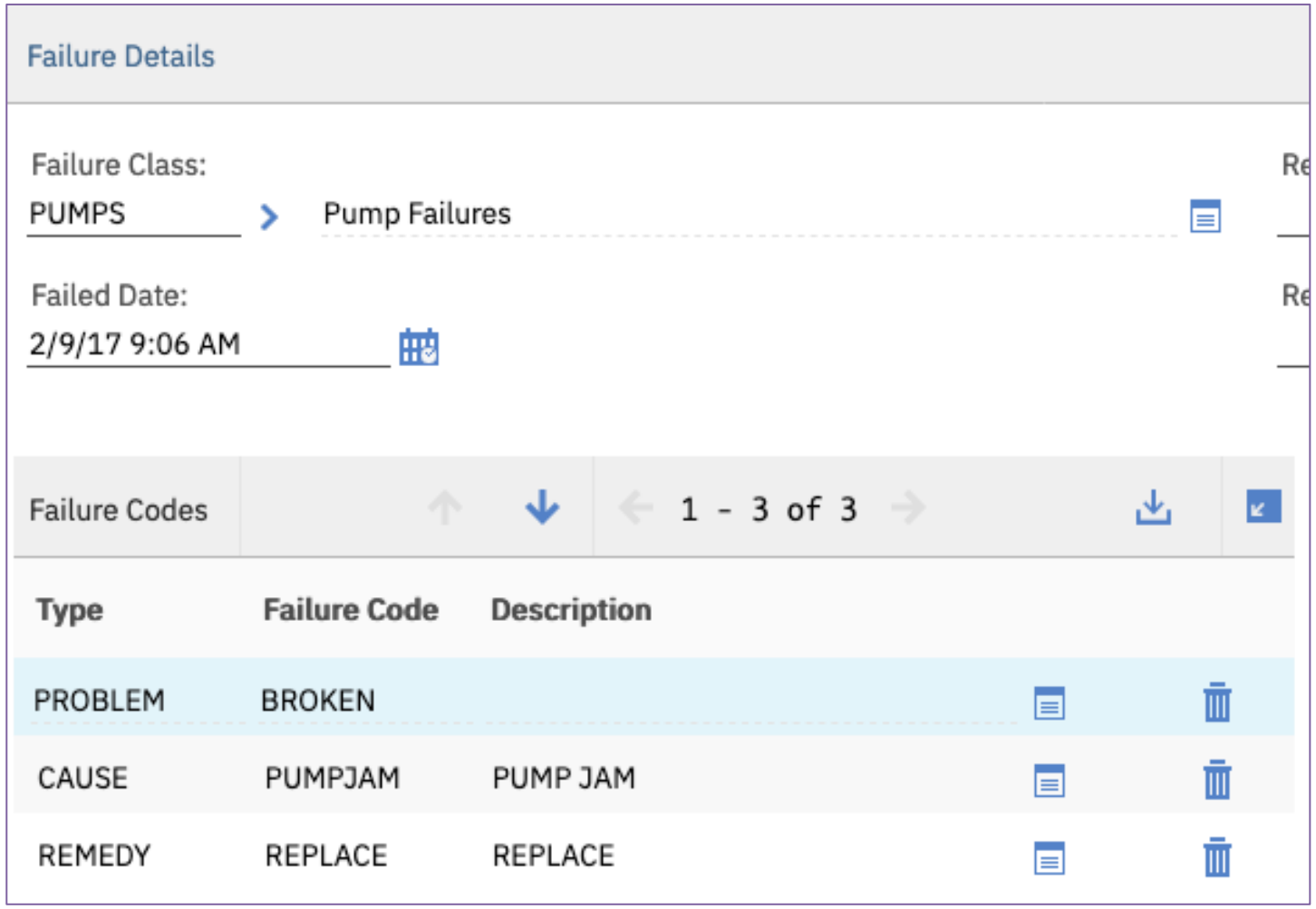




Maximo Objects to read failures

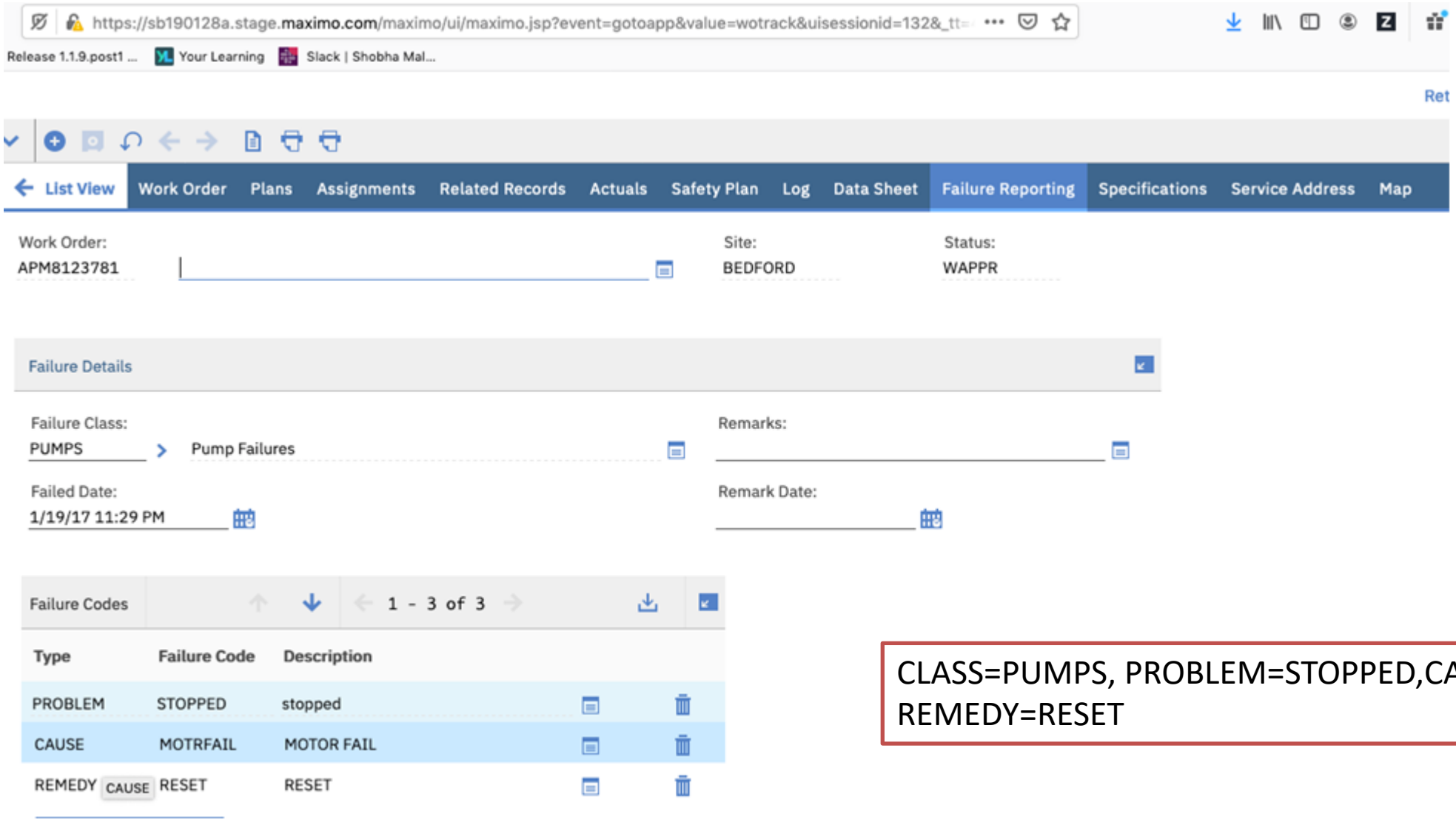
- In Maximo, Failure codes are part of the categorization mechanism of Work Orders and Tickets. They are usually entered when closing the WO/Ticket or when the reason of the failure is known.
  Failures codes represent the hierarchical structure describing the cause of the failure and suggesting the 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 Codes:
[PUMPS/BROKEN]
[PUMPS/BROKEN/PUMPJAM]
[PUMPS/BROKEN/PUMPJAM/REPLACE]

Maximo Failure Codes – A screenshot



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
cters 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

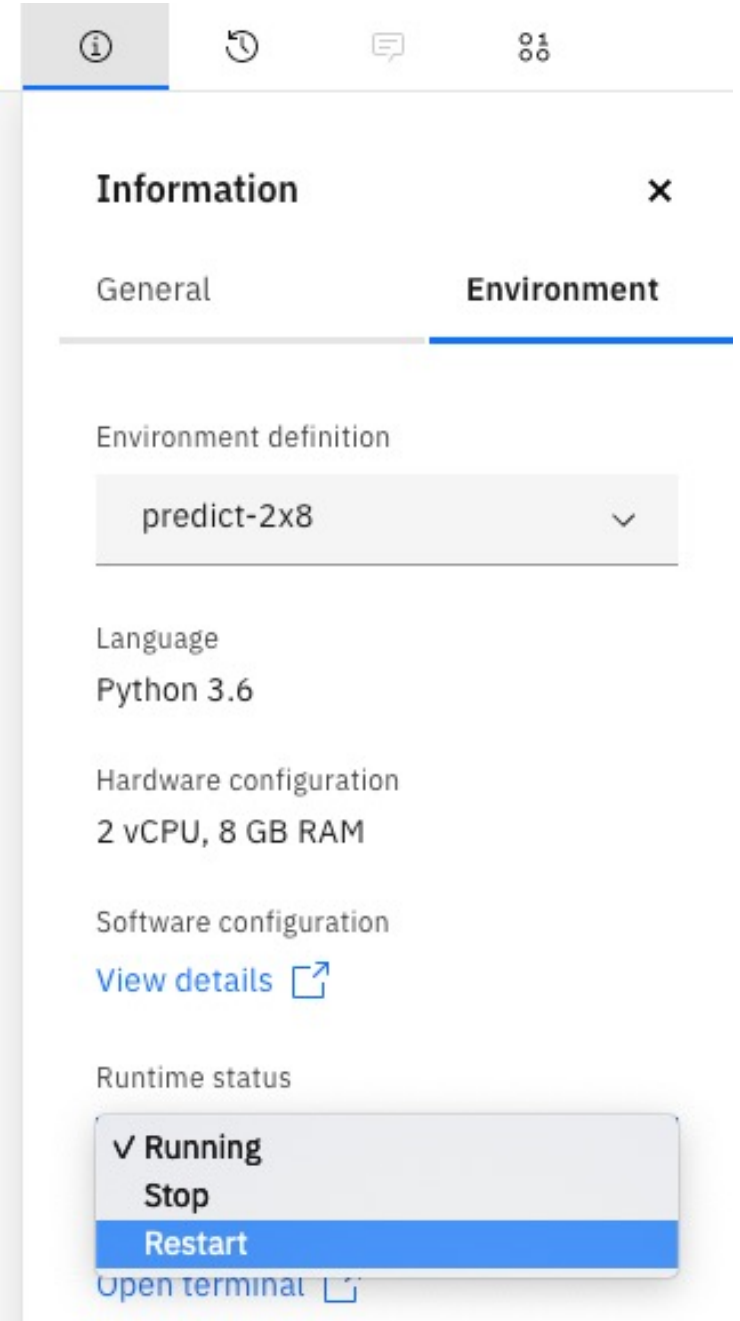
Minor error message can be ignored.

```
!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
```

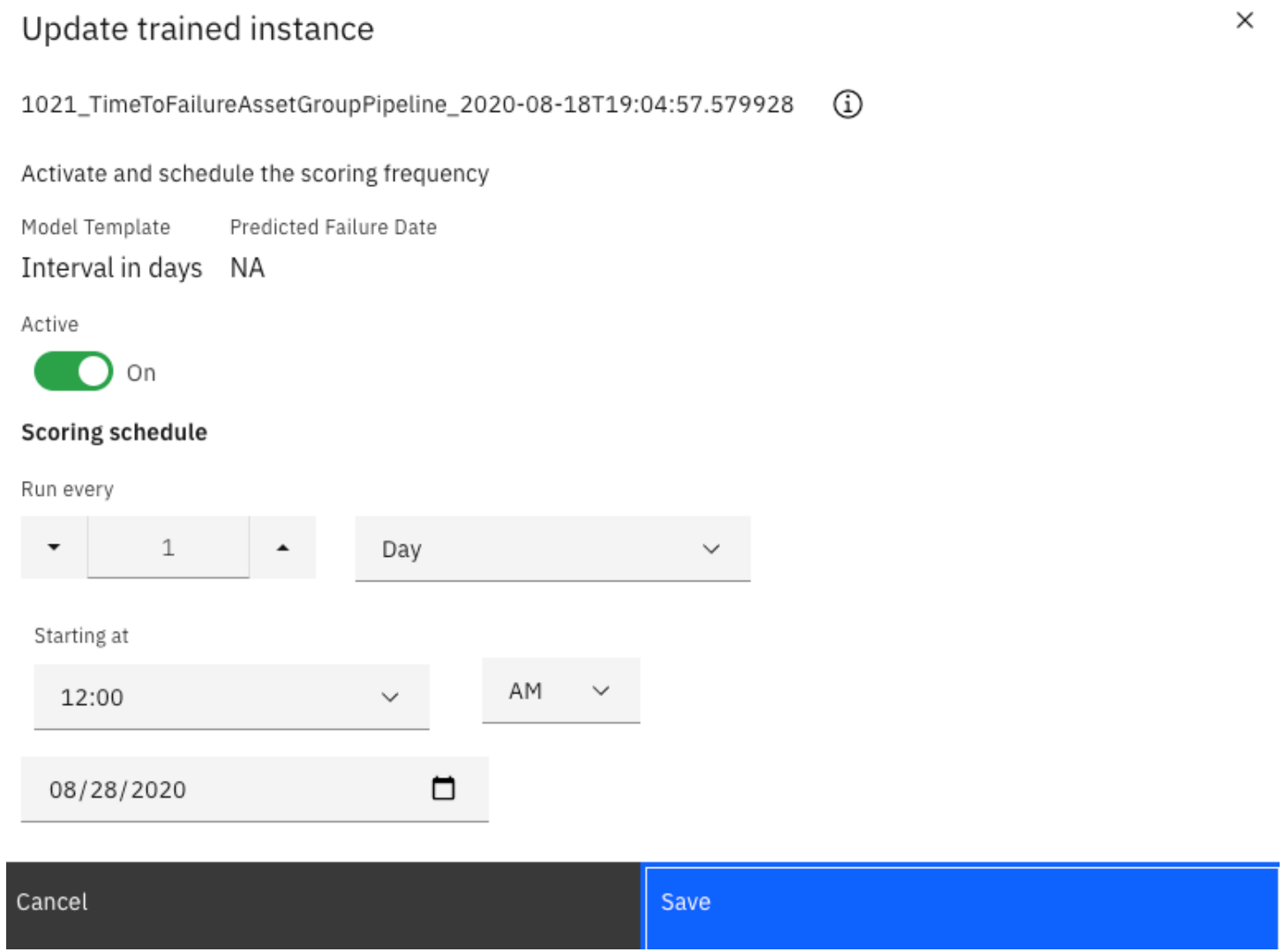
CP4D runtime issue

- 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"})
```



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"
    }
  }
}
```

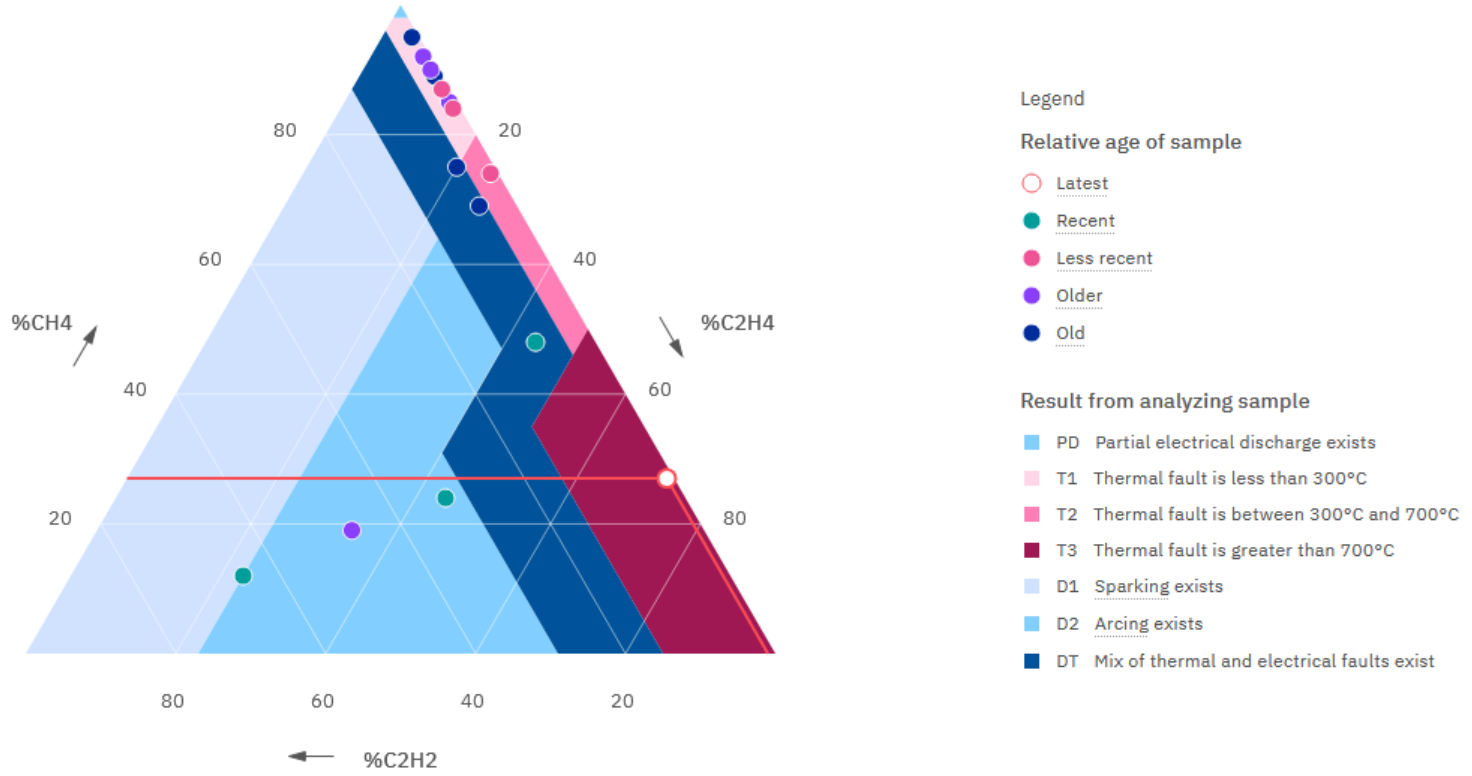
End

Maximo Health and Predict – Utilities

Pre-defined asset classes for transmission & distribution

Scoring Models (Health, Criticality, Effective Age, EOL probability, Risk, and dissolved gas analysis for transformers using Duval triangle visualization)

- Python-based
- Modifiable using included Watson Studio Jupyter notebooks



- Distribution Transformer
- Substation Transformer
- Instrument Transformer
- Switchgear Gas Insulated
- Metal Support Structure
- Overhead (OH) Transmission Wire
- Wood Power Pole
- Circuit Breaker (CB) Oil
- CB Air Blast
- CB Air Magnetic
- CB Vacuum
- CB SF6
- Underground (UG) Transmission Manholes / Splices
- UG Transmission Cable – High Pressure Fluid Filled (HPFF) Pipe Type Cables
- UG Transmission Cable – Mass Impregnated (MI) Cables
- UG Transmission Cable – Extruded Cross Linked Polyethylene (XLPE) Cables
- Conductors Model
- MI Cables Model
- SCFF Cables Model
- Transformers Tap Changers Model
- Transformers Tap Changers DGA Model

Maximo Health and Predict - Utilities

Build projects to optimize asset investments for transmission and distribution projects

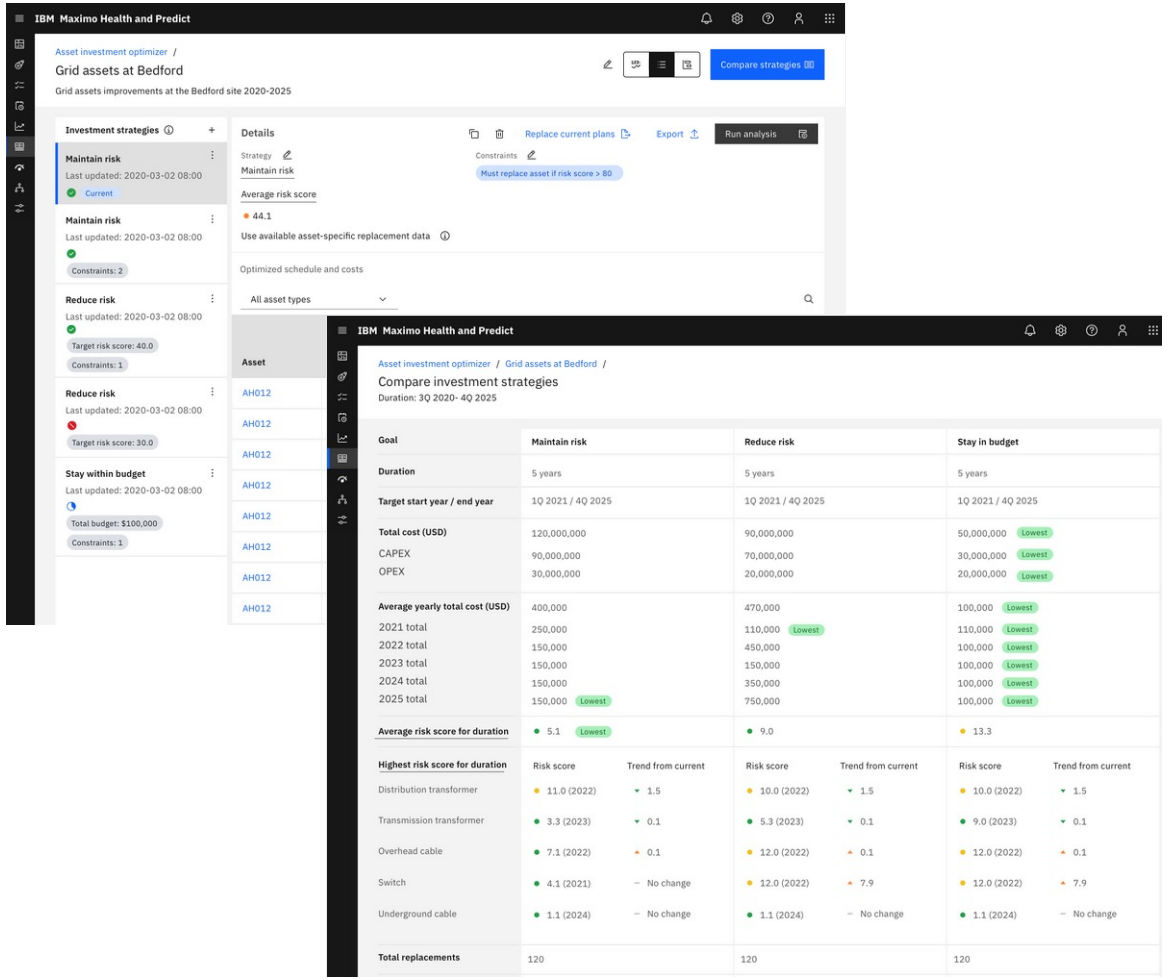
Asset Investment Optimization

Build projects, add assets to those projects, and run replacement scenarios to determine which assets should be replaced and when

Use replacement information from templates built for common asset types OR from individual asset replacement plans

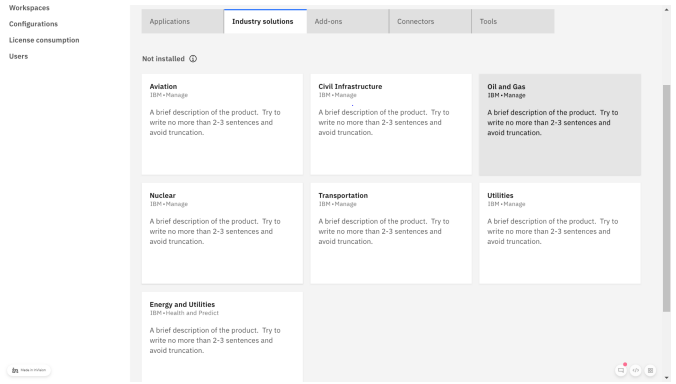
Compare current plans and optimized strategy

Submit plans for approval or for additional finance analysis in a full AIP solution



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

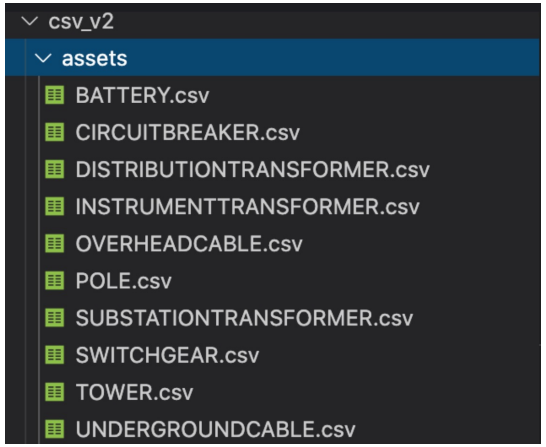
Maximo Health & Predict - Utilities tile



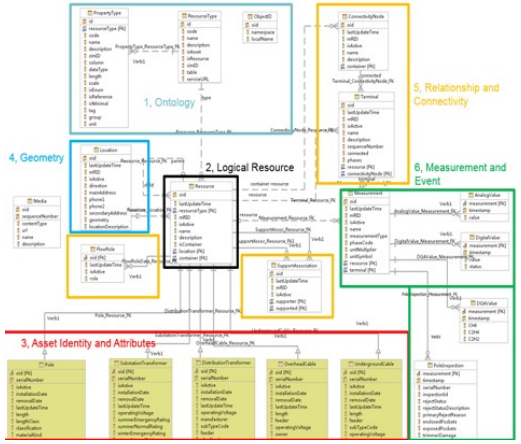
6 asset class python models

```
[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 an
  "C", "2", "Bushings are not broken, however major chips, a
  "D", "1", "Bushings are broken/damaged or cementing and fa
  "E", "0", "Bushings, cementing or fasteners are broken/dam
```

AppConnect data loader scripts

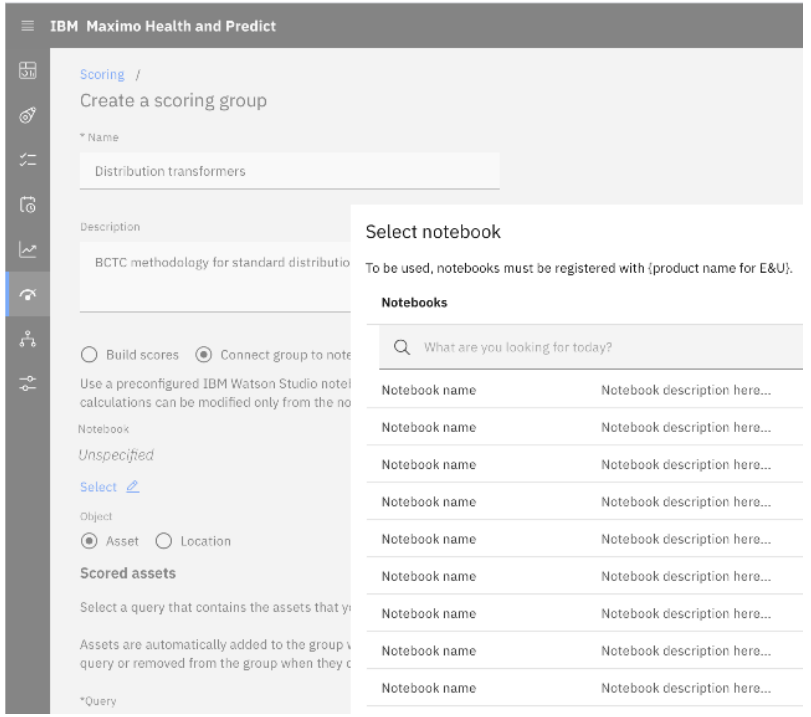


Industry standard data model



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

Watson Studio notebook scoring



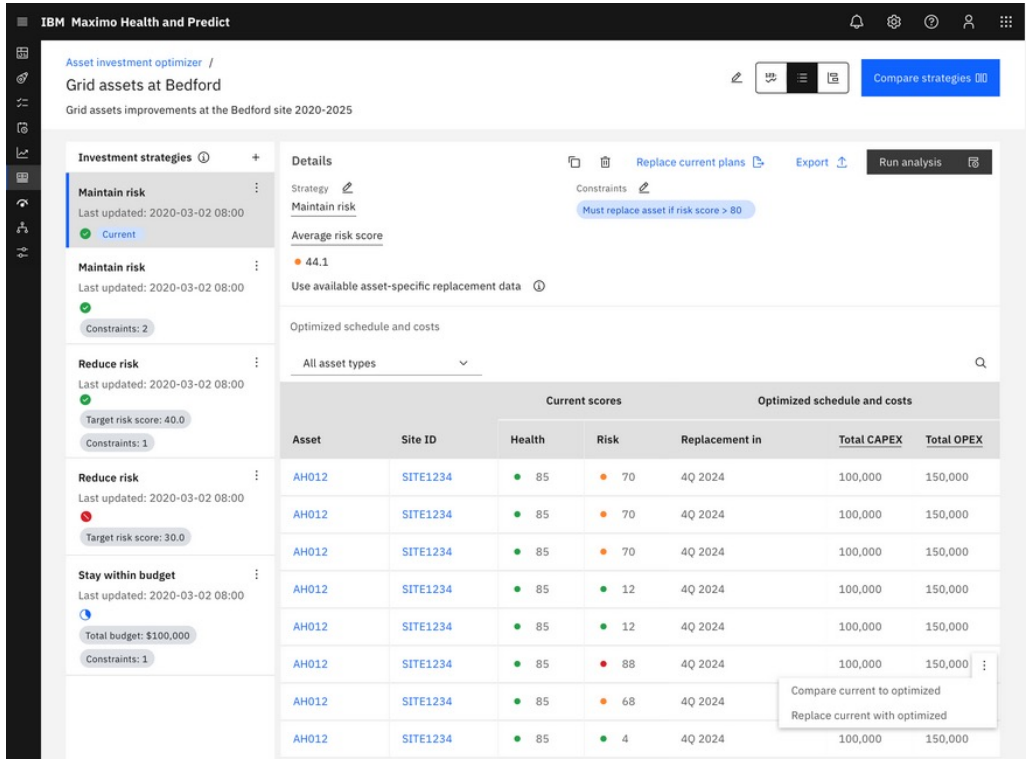
MAS Health & Predict Utilities – Asset Investment Optimization

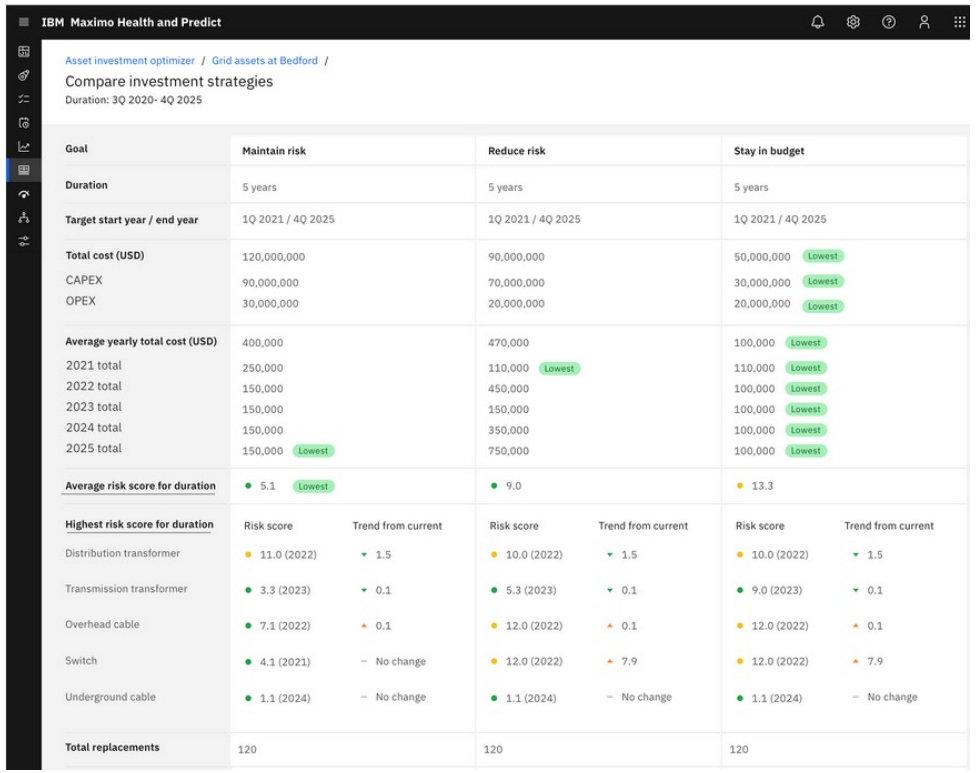
Use case

- 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



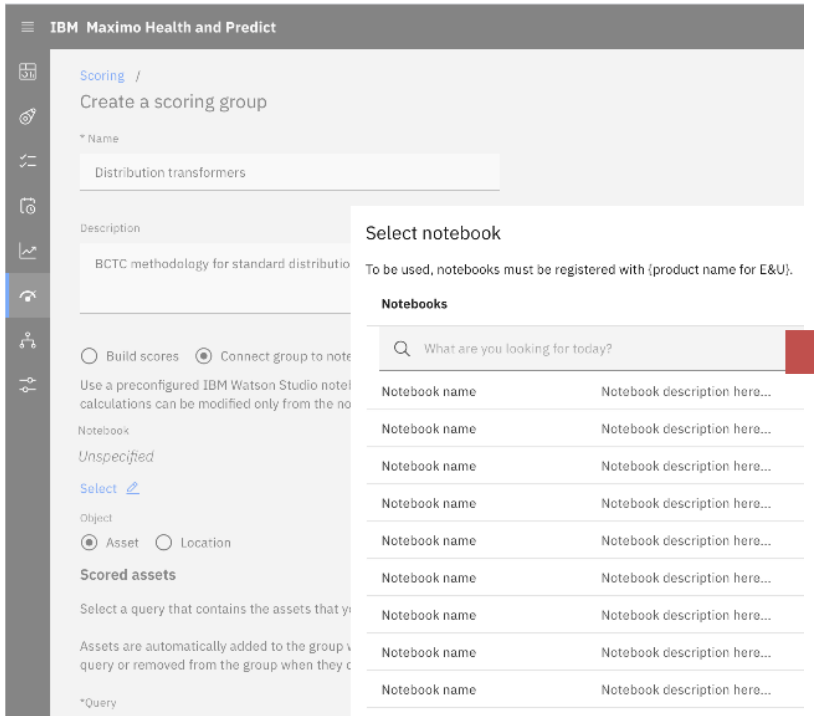


MAS 8.x Utilities Scoring Model extensions

- 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

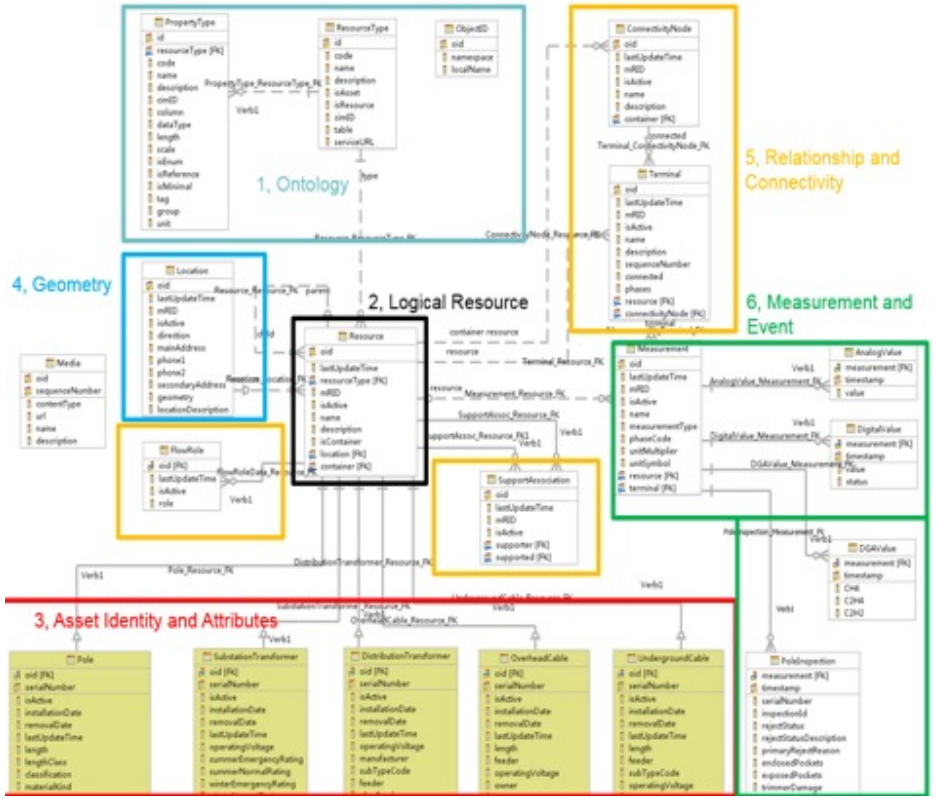



```
[Bushing Condition]
type = contributor
desc = Transformer Bushing Condition
func = common_calculate_condition_by_meter
# parameter.<type>.<item>
parameter.meter.name = Bushing Condition meter
parameter.meter.type = string
parameter.meter.format = meter
parameter.meter.desc = meter to measure Bushing 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 an
  "C", "2", "Bushings are not broken, however major chips, a
  "D", "1", "Bushings are broken/damaged or cementing and fa
  "E", "0", "Bushings, cementing or fasteners are broken/dam
```

MAS 8.x Utilities Common Information Model (CIM)

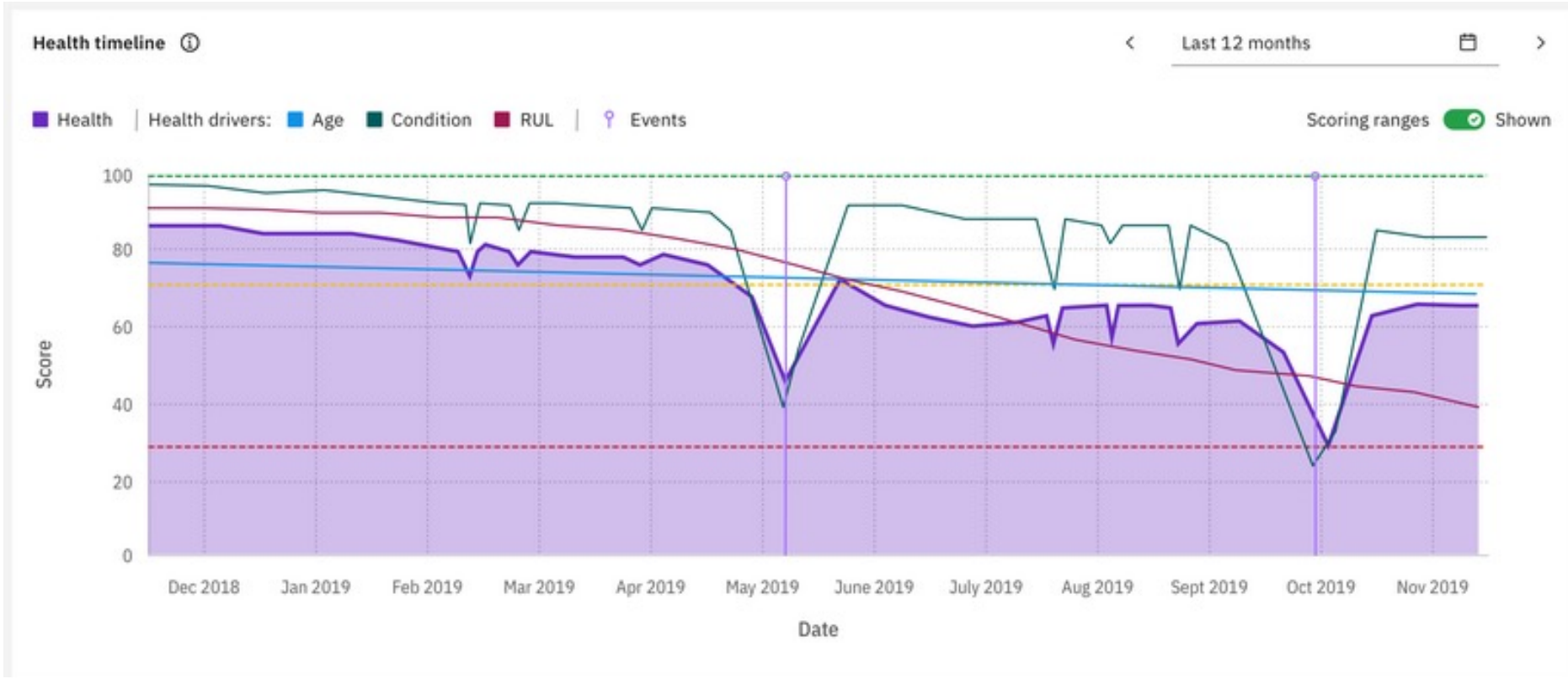
- CIM is a standard data model for describing assets information in Utilities industry
- Allows the customer to import data formatted based on CIM data model into Health or Manage

- 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