All you need to know about Predictive Maintenance using Azure ML Studio

Indraneel Pole

Software Developer at daenet GmbH

Blogger, Community Lead

Indraneel Pole

- Software Developer at daenet GmbH – A Microsoft Gold Partner Firm
- Blogger, Technical Author – Blog regularly at developers.de
- Community Lead – Organizer of Azure Meetup Frankfurt

daenet

https://developers.de/author/indraneel

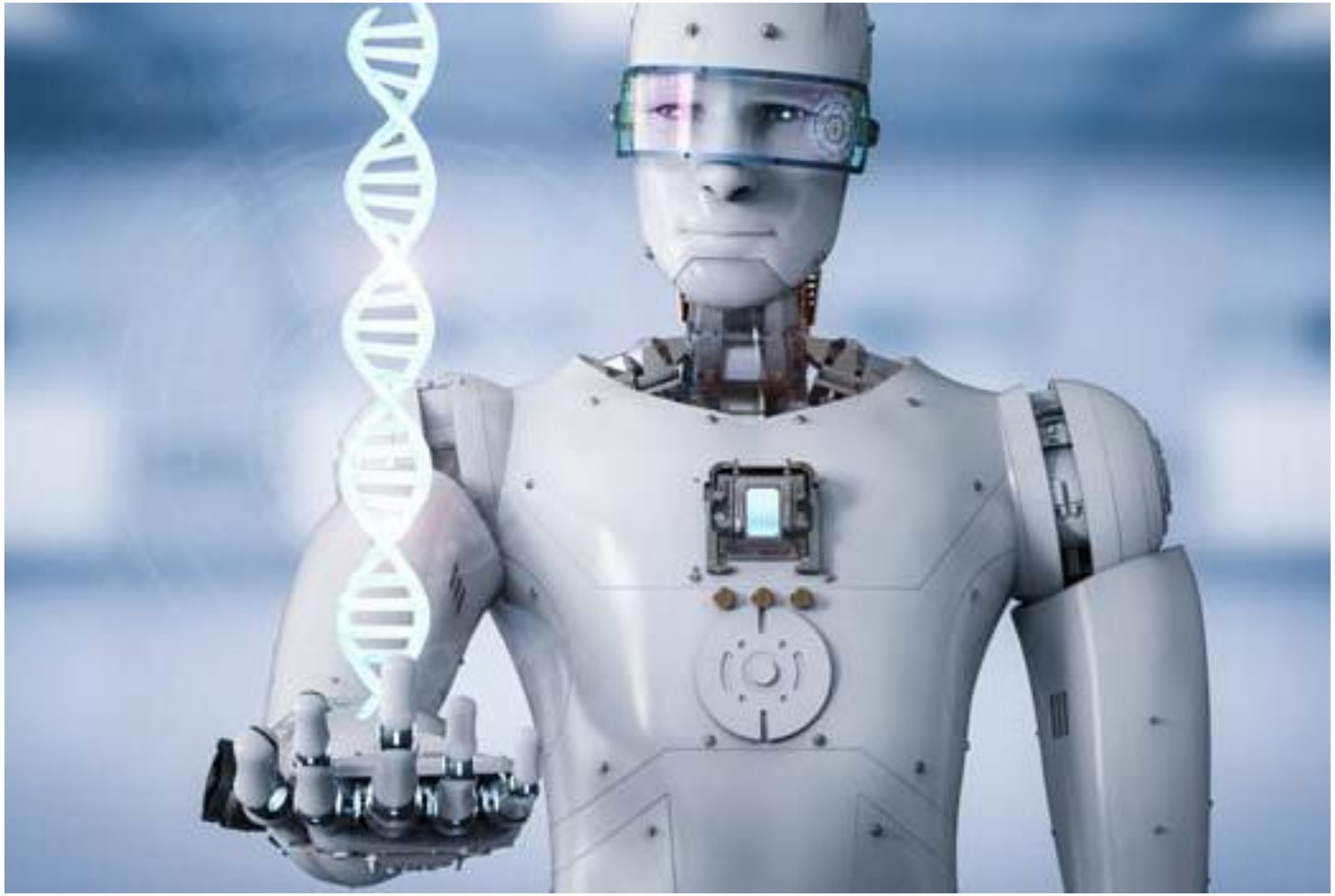
https://www.linkedin.com/in/ipole

@indraneelpole

Agenda

- Machine Learning at a glance
- Predictive Maintenance
- Azure Machine Learning Studio
- Q&A

Machine Learning at a glance



Machine Learning - Expectation

Reality

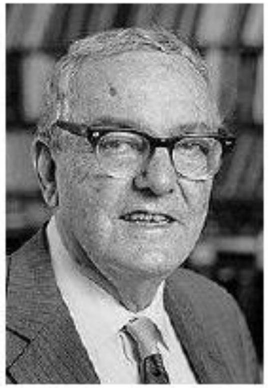
```
    */
    void chasePlayer ()
    {

        Avoid ();

        distanceToPlayer = (enemyTarget.position - transform.position).magnitude;

        if (distanceToPlayer < maxRange)
            follow = true;

        if(follow){
            if(distanceToPlayer < 6)
            {
                animation.CrossFade("footkick");
            }
            else
            {
            animation.CrossFade("walk");
            transform.LookAt(enemyTarget);
            Vector3 direction = transform.forward;
            Vector3 velocity = direction * speed;
            enemyController.SimpleMove(velocity);
            }
        }

        rayCasting ();
    }
```

Machine Learning is not AI

Textbook Definition

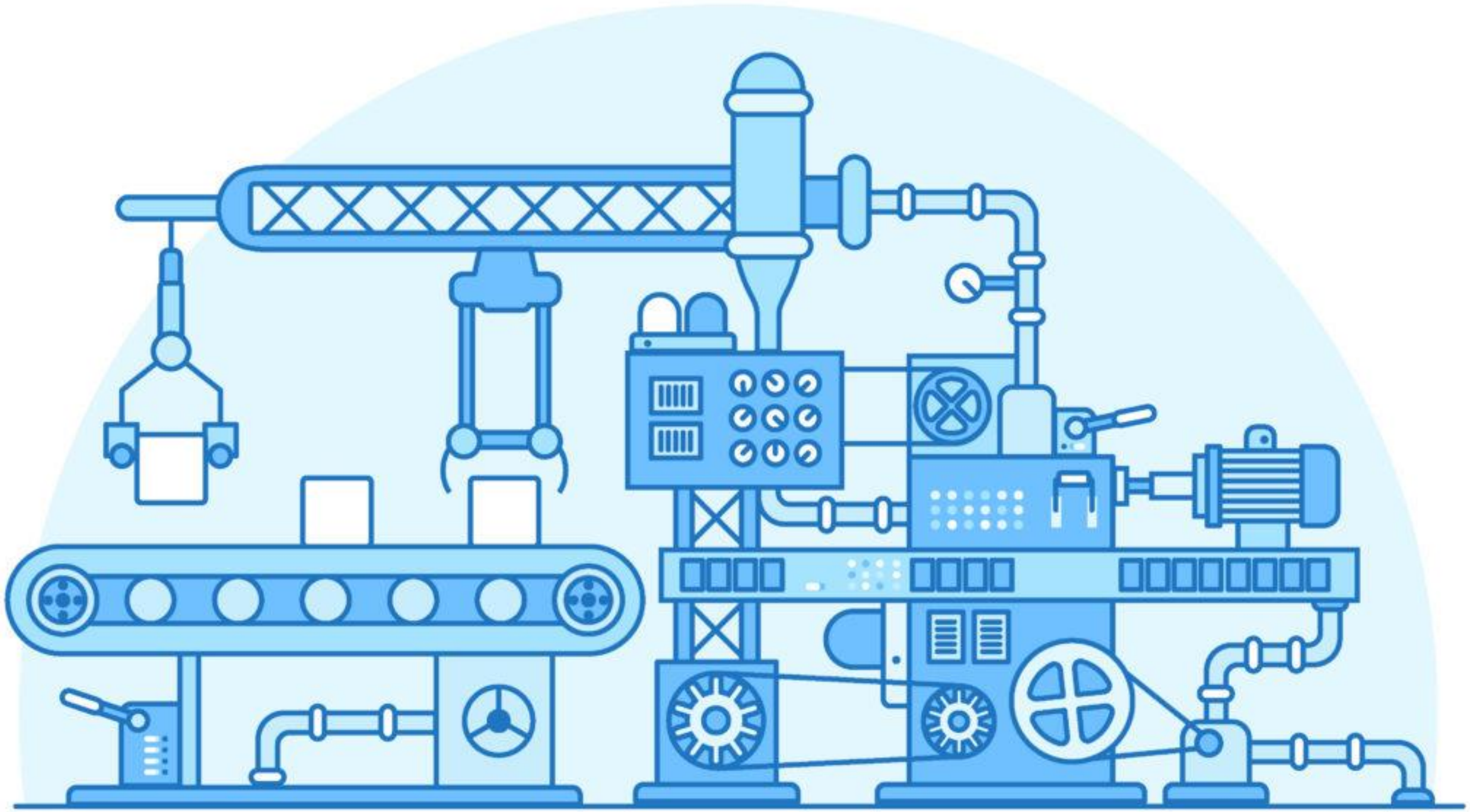
Machine Learning

- **Herbert Alexander Simon**: "Learning is any process by which a system improves performance from experience."
- "Machine Learning is concerned with computer programs that automatically improve their performance through experience. "



Herbert Simon
Turing Award 1975
Nobel Prize in Economics 1978

Typical Machine Learning Applications

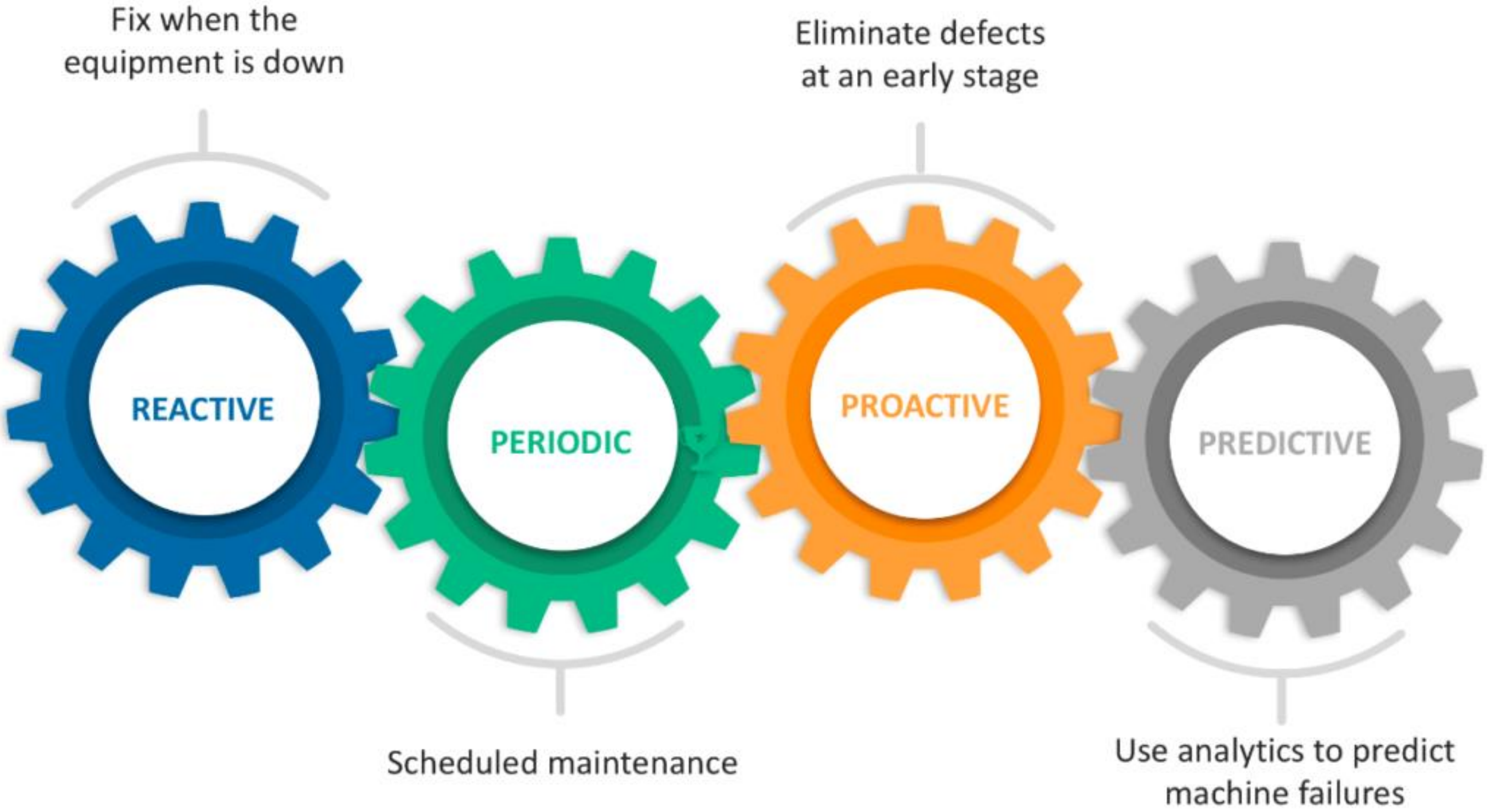


Predictive Maintenance

What is Predictive Maintenance?

- Application of Predictive Analysis
- High Asset Optimization and Low Operational Costs

Predictive Maintenance



Business case

- Most businesses rely on Corrective Maintenance
- Some businesses practice Preventive Maintenance
- Predictive Maintenance optimizes the balance between corrective and preventative maintenance, by enabling *just in time* replacement of components.

Problems

Detect	Detect anomalies in equipment or system performance or functionality.
Estimate	Estimate the remaining useful life of an asset
Identify	Identify the main causes of failure of an asset

Goals

- Reduce operational risk of mission critical equipment
- Control cost of maintenance by enabling just-in-time maintenance operations
- Discover patterns connected to various maintenance problems
- Provide Key Performance Indicators

Problem Qualification

- The problem has to be predictive in nature
- Record of the operational history for both success and failure
- Domain experts

Sample Use Cases

- Aviation – Flight delays / cancellation due to mechanical problems
- Finance – ATM failures, link fails
- Transportation and Logistics

Data Requirement

Relevant data – Machine data, telemetry, error logs

Sufficient data – How many failures? How many records?

Quality and accuracy of data

Data Preparation

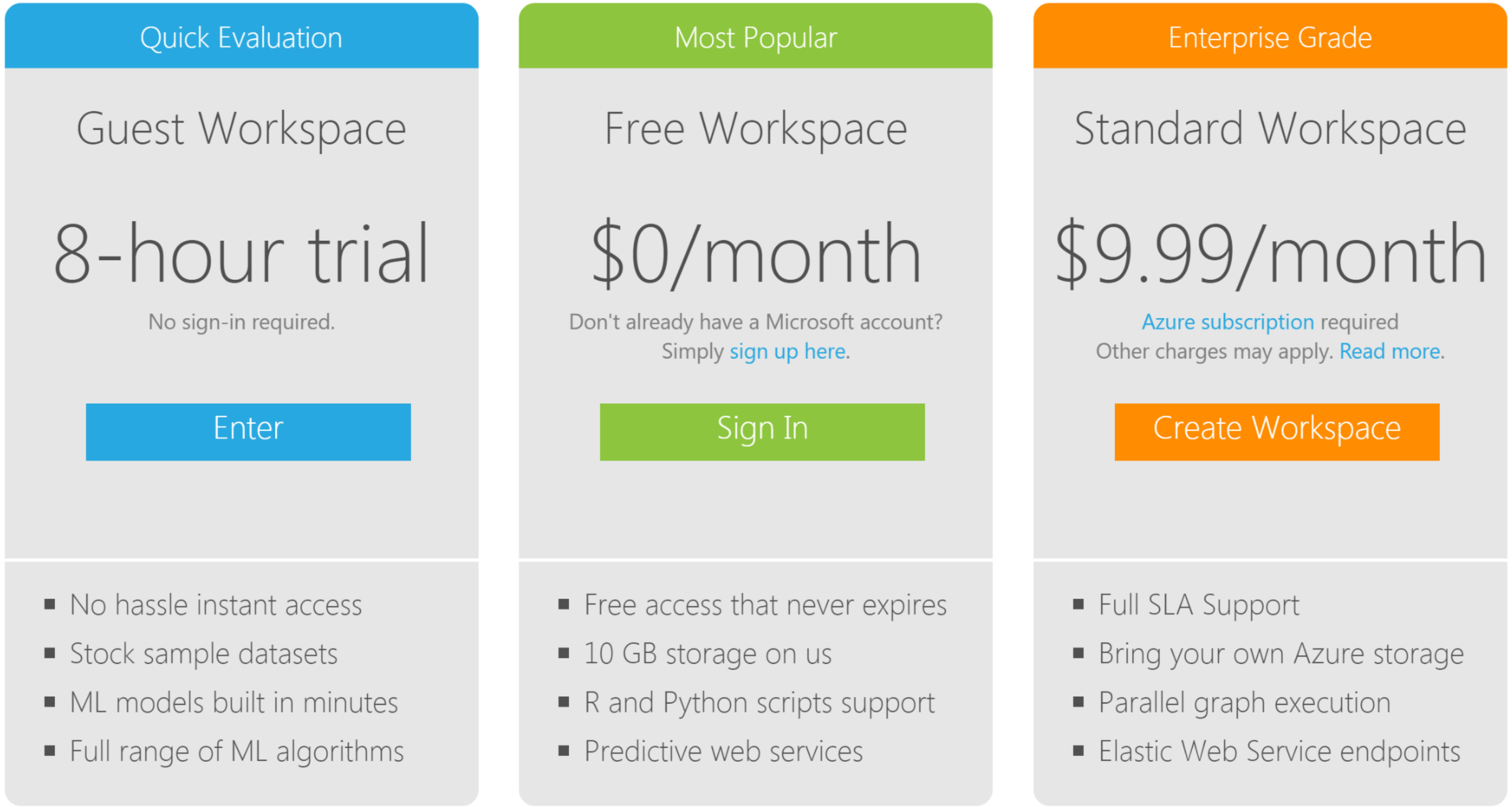
- Failure history
- Maintenance/repair history
- Machine operating conditions
- Equipment metadata

Azure Machine Learning Studio

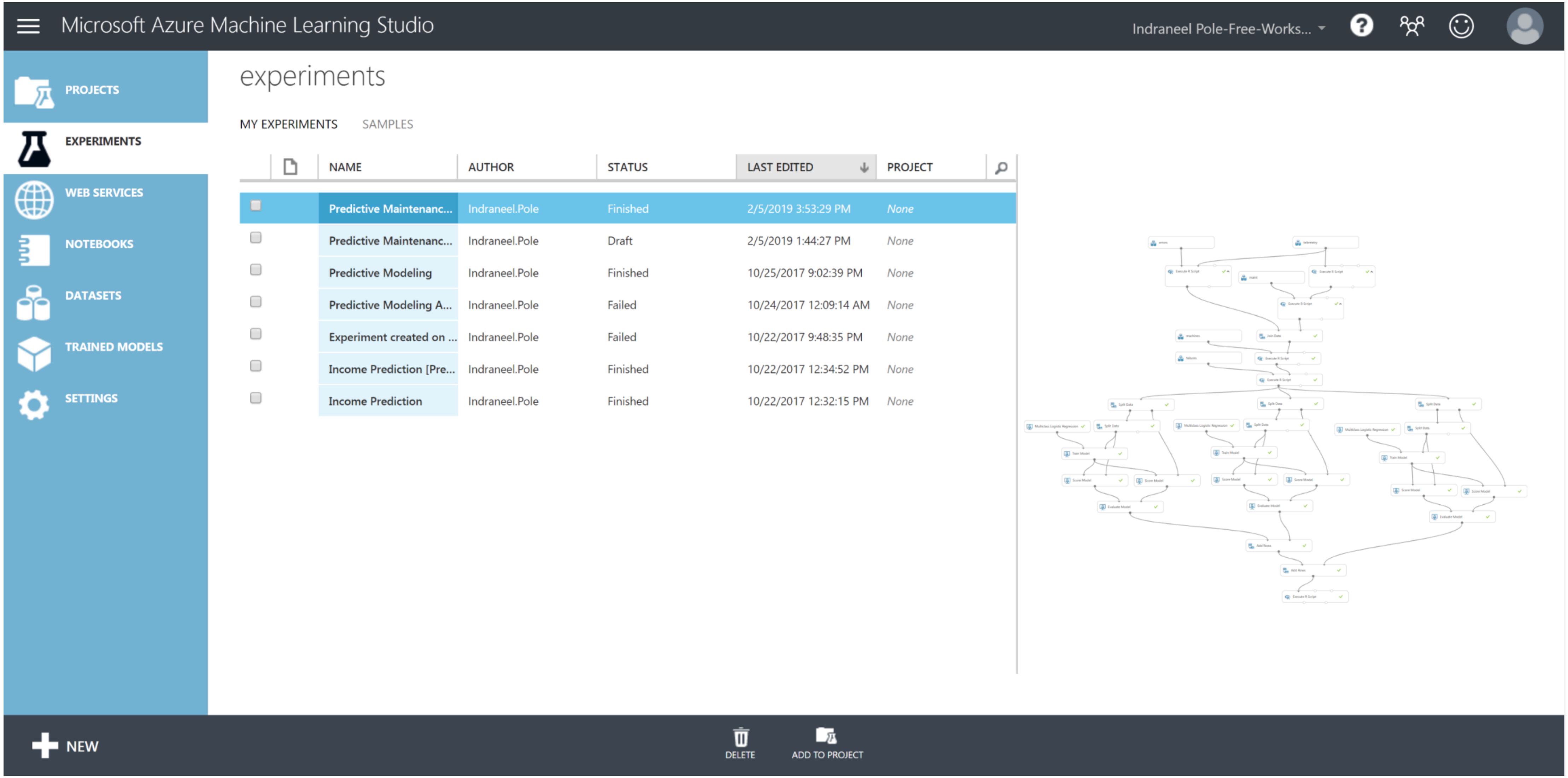
- A fully-managed cloud service that enables you to easily build, deploy, and share predictive analytics solutions
- Designed for Applied Machine Learning
- Interactive Workspace
- Drag and Drop modules
- Prebuilt R and Python packages

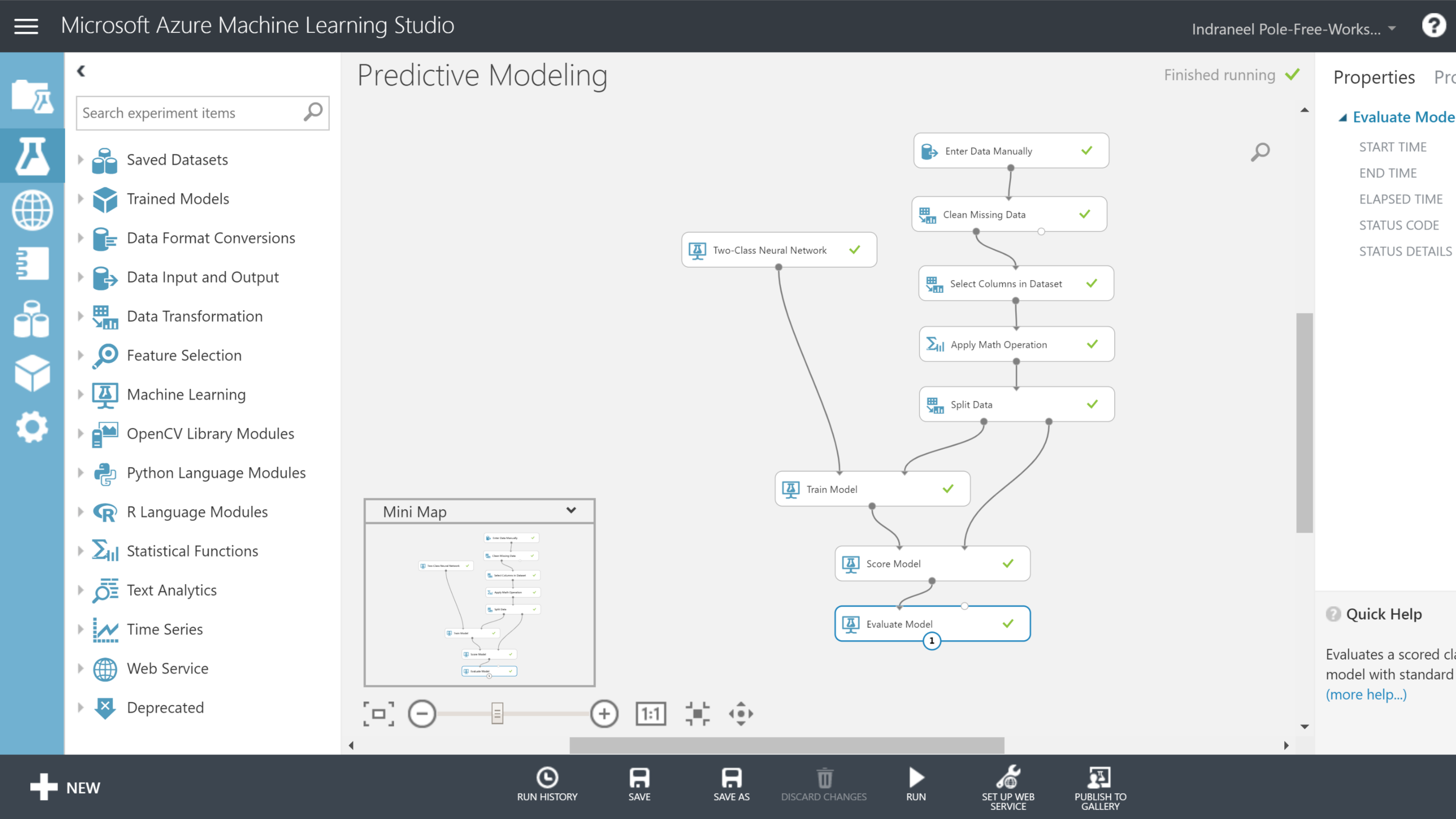
Azure Machine Learning Studio

https://studio.azureml.net/



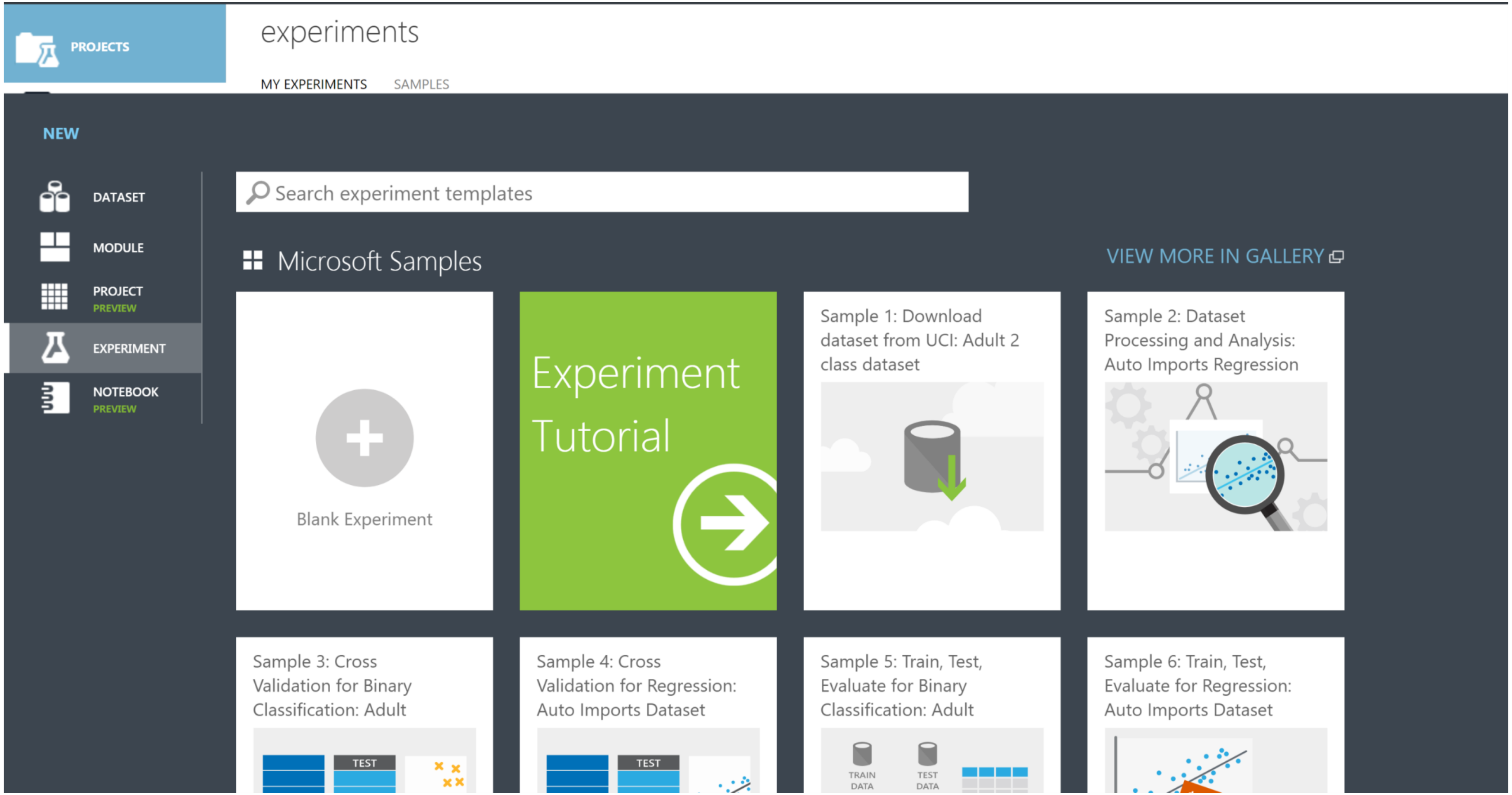
ML Studio

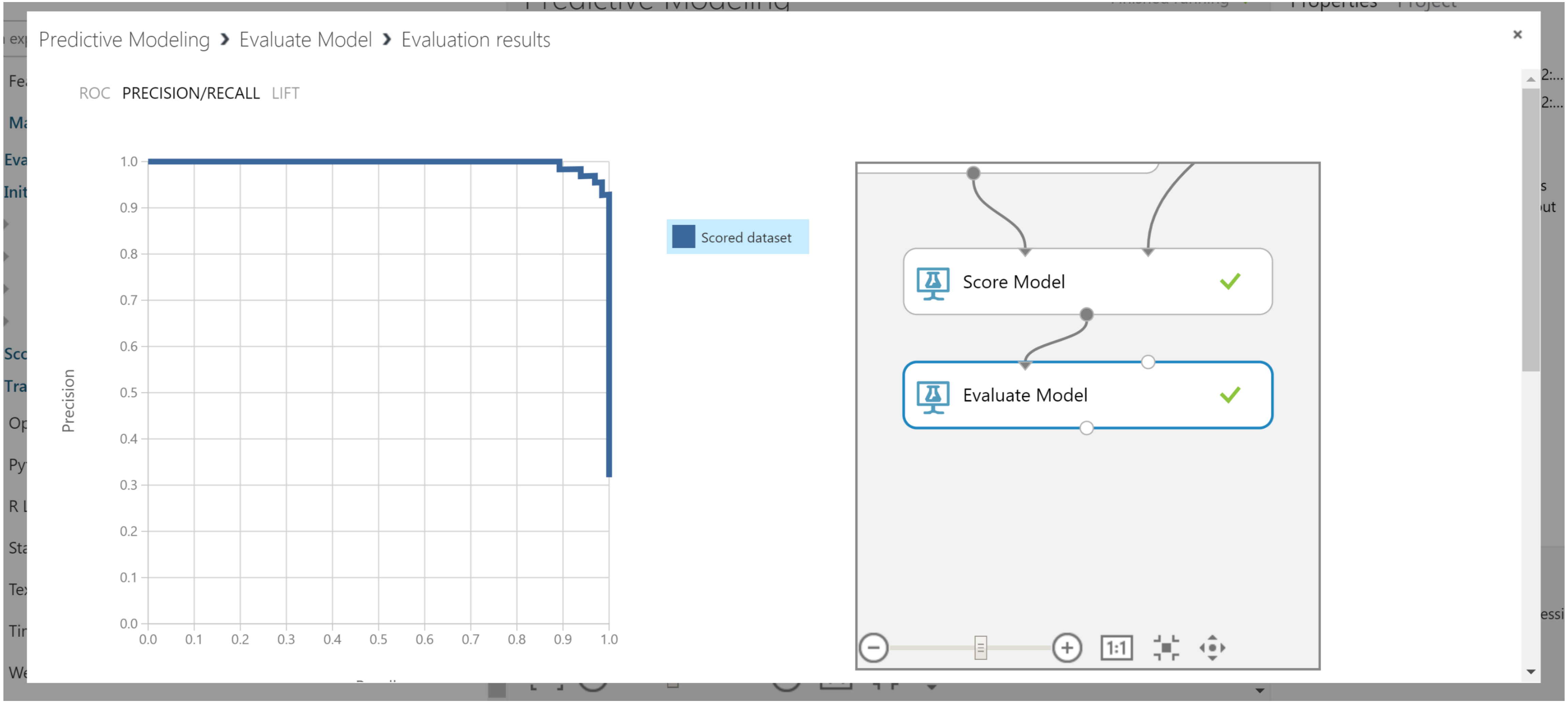

Microsoft Azure Machine Learning Studio
Indraneel Pole-Free-Works...
PROJECTS
EXPERIMENTS
WEB SERVICES
NOTEBOOKS
DATASETS
TRAINED MODELS
SETTINGS
experiments
MY EXPERIMENTS
SAMPLES
NAME
AUTHOR
STATUS
LAST EDITED
PROJECT
Predictive Maintenanc... Indraneel.Pole Finished 2/5/2019 3:53:29 PM None
Predictive Maintenanc... Indraneel.Pole Draft 2/5/2019 1:44:27 PM None
Predictive Modeling Indraneel.Pole Finished 10/25/2017 9:02:39 PM None
Predictive Modeling A... Indraneel.Pole Failed 10/24/2017 12:09:14 AM None
Experiment created on ... Indraneel.Pole Failed 10/22/2017 9:48:35 PM None
Income Prediction [Pre... Indraneel.Pole Finished 10/22/2017 12:34:52 PM None
Income Prediction Indraneel.Pole Finished 10/22/2017 12:32:15 PM None
NEW
DELETE
ADD TO PROJECT



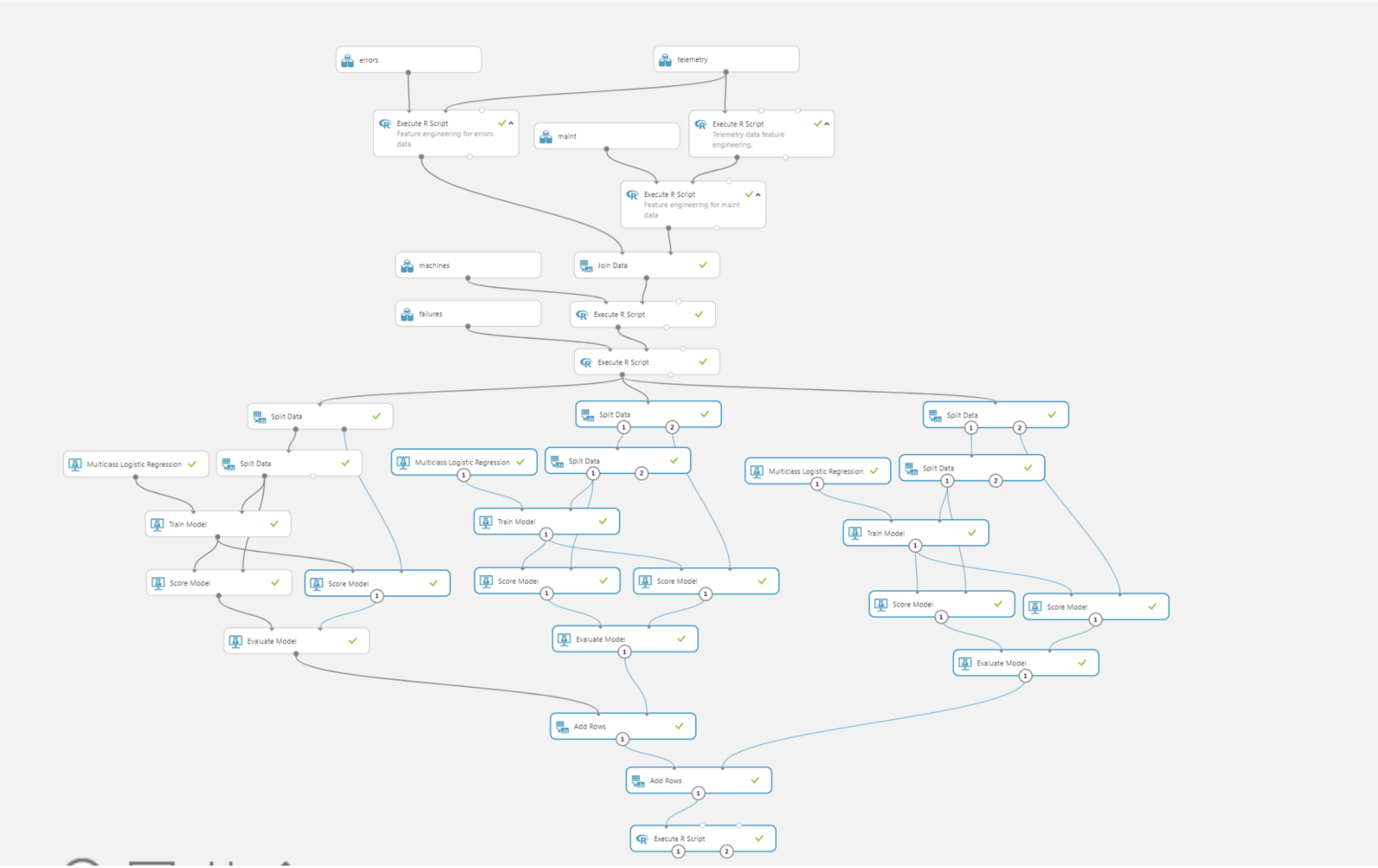
Microsoft Azure Machine Learning Studio
Indraneel Pole-Free-Works...
Search experiment items
Saved Datasets
Trained Models
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Time Series
Web Service
Deprecated
Predictive Modeling
Finished running
Enter Data Manually
Clean Missing Data
Two-Class Neural Network
Select Columns in Dataset
Apply Math Operation
Split Data
Train Model
Score Model
Evaluate Model
Mini Map
Properties
Evaluate Mode
START TIME
END TIME
ELAPSED TIME
STATUS CODE
STATUS DETAILS
Quick Help
Evaluates a scored cl
model with standard
(more help...)
NEW
RUN HISTORY
SAVE
SAVE AS
DISCARD CHANGES
RUN
SET UP WEB SERVICE
PUBLISH TO GALLERY

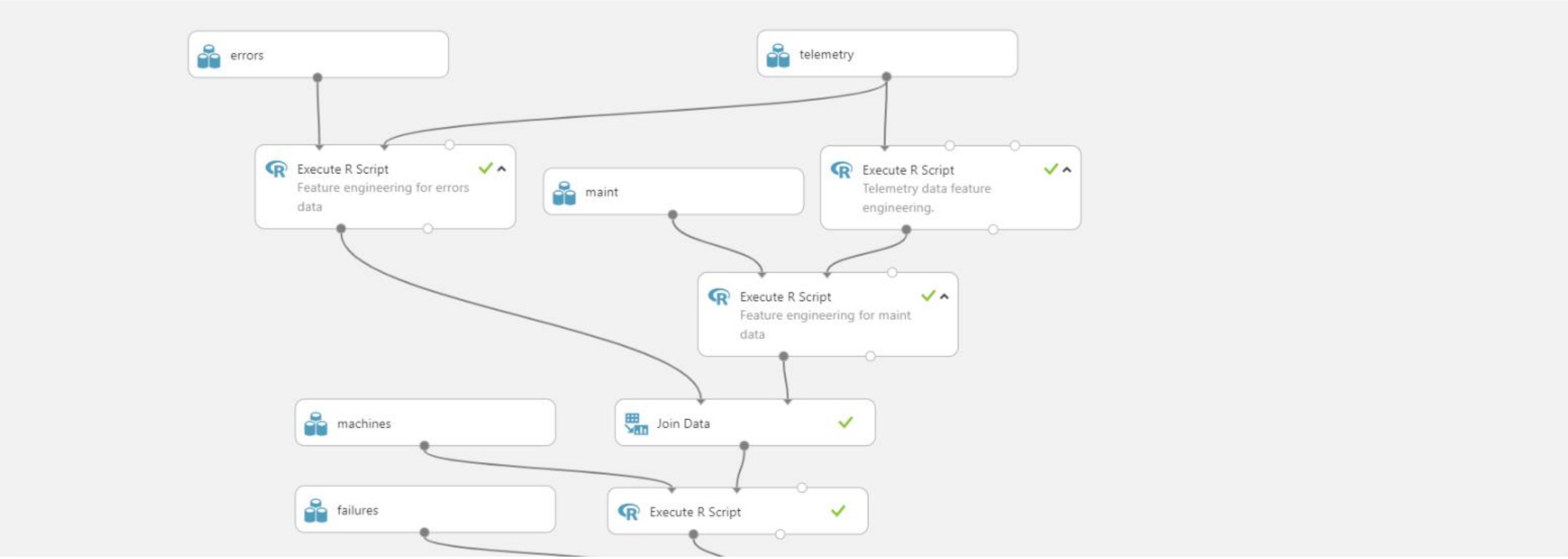
New Experiment




Predictive Modeling › Evaluate Model › Evaluation results
ROC
PRECISION/RECALL
LIFT
Precision
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Scored dataset
Score Model
Evaluate Model
1:1

Predictive Maintenance on ML Studio





Data Sources

- Failure History
- Maintenance History
- Sensor data
- Features – (Machine model, engine size etc)
- Metadata – Telemetry
- Error labels

Telemetry

rows
876100

columns
6

view as

datetime	machineID	volt	rotate	pressure	vibration
2015-01-01T06:00:00	1	176.217853	418.504078	113.077935	45.087686
2015-01-01T07:00:00	1	162.879223	402.74749	95.460525	43.413973
2015-01-01T08:00:00	1	170.989902	527.349825	75.237905	34.178847
2015-01-01T09:00:00	1	162.462833	346.149335	109.248561	41.122144
2015-01-01T10:00:00	1	157.610021	435.376873	111.886648	25.990511
2015-01-01T11:00:00	1	172.504839	430.323362	95.927042	35.655017
2015-01-01T12:00:00	1	156.556031	499.071623	111.755684	42.75392
2015-01-01T13:00:00	1	172.522781	409.624717	101.001083	35.482009
2015-01-01T14:00:00	1	175.324524	398.648781	110.624361	45.482287
2015-01-01T15:00:00	1	169.218423	460.85067	104.84823	39.901735
2015-01-01T16:00:00	1	167.060981	382.483543	103.780663	42.6758
2015-01-01T17:00:00	1	160.263954	448.084256	96.480976	38.543681
2015-01-01T18:00:00	1	153.353492	490.672921	86.01244	44.108554

Statistics

Visualizations

To view, select a column in the table.

Error Logs

- Non failures
- Non breaking errors
- Machine is still operational

Maintenance

- Scheduled as well as unscheduled maintenance
- Record is created if component is replaced (Inspection / Breakdown)
- Breakdown records are called failures

Predictive Maintenance Modelling Regression › maint › dataset

rows	columns
3286	3

view as

datetime	machineID	comp
2014-06-01T06:00:00	1	comp2
2014-07-16T06:00:00	1	comp4
2014-07-31T06:00:00	1	comp3
2014-12-13T06:00:00	1	comp1
2015-01-05T06:00:00	1	comp4
2015-01-05T06:00:00	1	comp1
2015-01-20T06:00:00	1	comp3
2015-01-20T06:00:00	1	comp1
2015-02-04T06:00:00	1	comp4
2015-02-04T06:00:00	1	comp3
2015-02-19T06:00:00	1	comp3
2015-03-06T06:00:00	1	comp1
2015-03-21T06:00:00	1	comp1

Statistics

Visualizations

To view, select a column in the table.

Predictive Maintenance Modelling Regression › failures › dataset

rows 761

columns 3

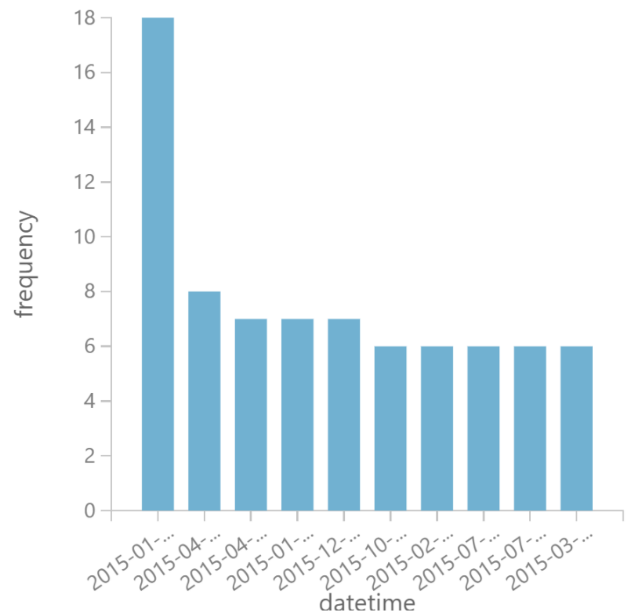
view as

datetime	machineID	failure
2015-01-05T06:00:00	1	comp4
2015-03-06T06:00:00	1	comp1
2015-04-20T06:00:00	1	comp2
2015-06-19T06:00:00	1	comp4
2015-09-02T06:00:00	1	comp4
2015-10-17T06:00:00	1	comp2
2015-12-16T06:00:00	1	comp4
2015-03-19T06:00:00	2	comp1
2015-03-19T06:00:00	2	comp2
2015-04-18T06:00:00	2	comp2
2015-12-29T06:00:00	2	comp2
2015-01-07T06:00:00	3	comp2

datetime

Histogram

compare to None



Feature Engineering

- Bringing all data-sources together
- Create combined feature dataset that best describes machine's health condition

Properties Project

Execute R Script

R Script

```
library("scales", quietly = TRUE)

#install.packages("data.table", verbose = FALSE)
library("data.table", quietly = TRUE)

library("dplyr", quietly = TRUE)
library("zoo", quietly = TRUE)

##------------------------------------------------------------
# format datetime and comp fields
maint$datetime <- as.POSIXct(maint$datetime,
                             format = "%m/%d/%Y %I:%M:%S %
                             tz = "UTC")
maint$comp <- as.factor(maint$comp)

##------------------------------------------------------------
## Sanity check plot
theme_set(theme_bw())
```

Random Seed

R Version

CRAN R 3.1.0

START TIME	2/5/2019 3:59:56 PM
END TIME	2/5/2019 4:00:31 PM
ELAPSED TIME	0:00:35.078
STATUS CODE	Finished
STATUS DETAILS	None

View output log

rows 291669

columns 30

view as

datetime	machineID	error1count	error2count	error3count	error4count	error5count	voltmean_24hrs	rotatemean_24hrs	pressuremean_24hrs	vibrationmean_24
2015-01-02T06:00:00Z	1	0	0	0	0	0	170.614862	446.364859	96.849785	39.736826
2015-01-02T09:00:00Z	1	0	0	0	0	0	169.893965	447.009407	97.7156	39.498374
2015-01-02T12:00:00Z	1	0	0	0	0	0	171.243444	444.233563	96.66606	40.22937
2015-01-02T15:00:00Z	1	0	0	0	0	0	170.792486	448.440437	95.766838	40.055214
2015-01-02T18:00:00Z	1	0	0	0	0	0	170.556674	452.267095	98.06586	40.033247
2015-01-02T21:00:00Z	1	0	0	0	0	0	168.460525	451.031783	99.273286	38.903462
2015-01-03T00:00:00Z	1	0	0	0	0	0	169.772951	447.502464	99.005946	39.389725
2015-01-03T03:00:00Z	1	0	0	0	0	0	170.900562	453.864597	100.877342	38.696225
2015-01-03T06:00:00Z	1	0	0	0	0	0	169.533156	454.785072	100.050567	39.449734
2015-01-03T09:00:00Z	1	1	0	0	0	0	170.866013	463.871291	99.360632	40.766639
2015-01-03T12:00:00Z	1	1	0	0	0	0	171.041651	463.701291	98.965877	42.39685
2015-01-03T15:00:00Z	1	1	0	0	0	0	171.244533	464.320613	98.853189	44.608814
2015-01-03T18:00:00Z	1	1	0	0	0	0	171.385039	459.937314	97.292157	45.284751
2015-01-03T21:00:00Z	1	1	0	1	0	0	171.880633	461.437128	96.786742	47.311018
2015-01-										

Statistics

Visualizations

To view, select a colu
table.

Modelling

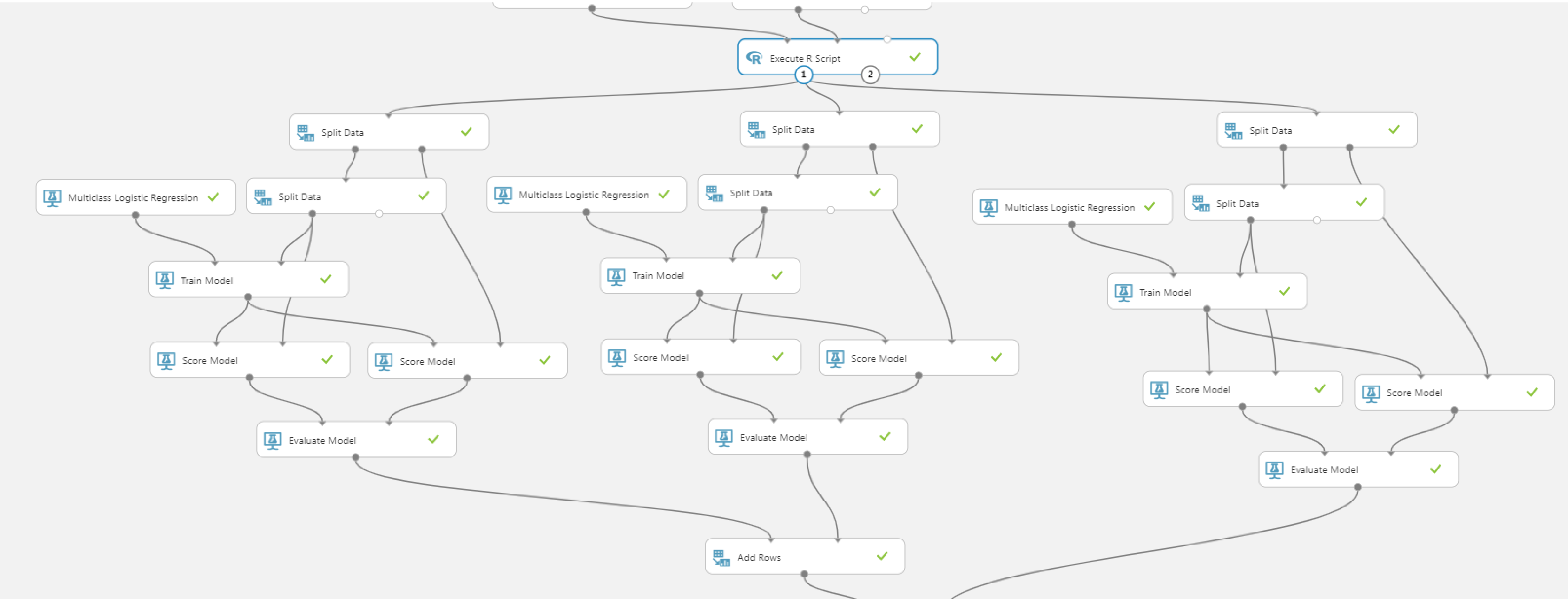
- Training
- Validation
- Testing

Modelling

- Splitting data into training validation and test sets
- Prevention of overestimating
- Trained on historical data and validated and tested on future data
- Avoid random sampling
- Time dependent splitting strategy

Evaluation

- Machine failures are rare occurrences
- Imbalance in label distribution
- Classify majority class examples better at the expense of minority class examples
- Because total misclassification error is much improved when majority class is labeled correctly.


Execute R Script
Split Data
Split Data
Split Data
Multiclass Logistic Regression
Split Data
Multiclass Logistic Regression
Split Data
Multiclass Logistic Regression
Split Data
Train Model
Train Model
Train Model
Score Model
Score Model
Score Model
Score Model
Score Model
Score Model
Evaluate Model
Evaluate Model
Evaluate Model
Add Rows

Evaluation Results

Predictive Maintenance Modelling Regression › Add Rows › Results dataset

rows 20
columns 9

Class	Predicted as "comp1"	Predicted as "comp2"	Predicted as "comp3"	Predicted as "comp4"	Predicted as "none"	Average Log Loss	Precision	Recall
comp1	804	30	7	15	177	0.741211	0.842767	0.778316
comp2	18	1097	8	30	141	0.513307	0.912646	0.847759
comp3	1	8	579	15	71	0.43048	0.899068	0.85905
comp4	10	26	10	772	100	0.556353	0.894554	0.840959
none	121	41	40	31	163146	0.006318	0.997012	0.998574
comp1	484	5	13	9	83	0.62822	0.743472	0.814815
comp2	20	843	1	28	109	0.645127	0.885504	0.842158
comp3	13	17	373	8	57	0.755748	0.934837	0.797009
comp4	17	28	1	542	69	0.593958	0.903333	0.824962
none	117	59	11	13	119851	0.006819	0.997354	0.998334
comp1	912	32	12	15	201	0.756639	0.823848	0.778157
comp2	24	1247	9	41	162	0.540444	0.901012	0.840863
comp3	13	7	666	14	80	0.428377	0.9	0.853846
comp4	10	40	11	895	115	0.545402	0.895896	0.835668
none	148	58	42	34	188164	0.006269	0.997043	0.998504
comp1	353	7	0	13	86	0.690683	0.767391	0.769063
comp2	10	704	1	9	88	0.607798	0.903723	0.866995
comp3	9	17	272	8	54	0.936588	0.967972	0.755556
comp4	13	16	1	424	52	0.55888	0.905983	0.837945
none	75	35	7	14	95642	0.005715	0.997081	0.998632

Confusion matrix

Predictive Maintenance Modelling Regression › Evaluate Model › Evaluation results

Metrics

Overall accuracy	0.994465
Average accuracy	0.997786
Micro-averaged precision	0.994465
Macro-averaged precision	0.90356
Micro-averaged recall	0.994465
Macro-averaged recall	0.861407

Confusion Matrix

Predicted Class (columns) / Actual Class (rows)

	comp1	comp2	comp3	comp4	none
comp1	77.8%	2.7%	1.0%	1.3%	17.2%
comp2	1.6%	84.1%	0.6%	2.8%	10.9%
comp3	1.7%	0.9%	85.4%	1.8%	10.3%
comp4	0.9%	3.7%	1.0%	83.6%	10.7%
none	0.1%	0.0%	0.0%	0.0%	99.9%

Metrics

Overall accuracy	0.99474
Average accuracy	0.997896
Micro-averaged precision	0.99474
Macro-averaged precision	0.90843
Micro-averaged recall	0.99474
Macro-averaged recall	0.845638

Confusion Matrix

Predicted Class (columns) / Actual Class (rows)

	comp1	comp2	comp3	comp4	none
comp1	76.9%	1.5%		2.8%	18.7%
comp2	1.2%	86.7%	0.1%	1.1%	10.8%
comp3	2.5%	4.7%	75.6%	2.2%	15.0%
comp4	2.6%	3.2%	0.2%	83.8%	10.3%
none	0.1%	0.0%	0.0%	0.0%	99.9%

Results using Neural Networks

Predictive Maintenance Modelling NN › Evaluate Model › Evaluation results

Metrics

Overall accuracy	0.995807
Average accuracy	0.998323
Micro-averaged precision	0.995807
Macro-averaged precision	0.893361
Micro-averaged recall	0.995807
Macro-averaged recall	0.922083

Confusion Matrix

Predicted Class (columns) / Actual Class (rows)

	comp1	comp2	comp3	comp4	none
comp1	86.6%	3.3%	2.1%	1.1%	6.8%
comp2	1.6%	88.2%	1.1%	3.9%	5.2%
comp3	0.3%	0.1%	95.4%	2.6%	1.7%
comp4	1.3%	2.9%	0.7%	91.0%	4.1%
none	0.1%	0.0%	0.1%	0.0%	99.8%

Metrics

Overall accuracy	0.992994
Average accuracy	0.997197
Micro-averaged precision	0.992994
Macro-averaged precision	0.812289
Micro-averaged recall	0.992994
Macro-averaged recall	0.89027

Confusion Matrix

Predicted Class (columns) / Actual Class (rows)

	comp1	comp2	comp3	comp4	none
comp1	83.2%	2.0%	0.2%	3.1%	11.5%
comp2	1.1%	89.0%	0.2%	2.1%	7.5%
comp3	2.5%	5.0%	84.2%	2.2%	6.1%
comp4	2.6%	2.4%	0.4%	89.1%	5.5%
none	0.1%	0.1%	0.1%	0.1%	99.6%

Q&A

https://developers.de/author/indraneel

https://www.linkedin.com/in/ipole

@indraneelpole