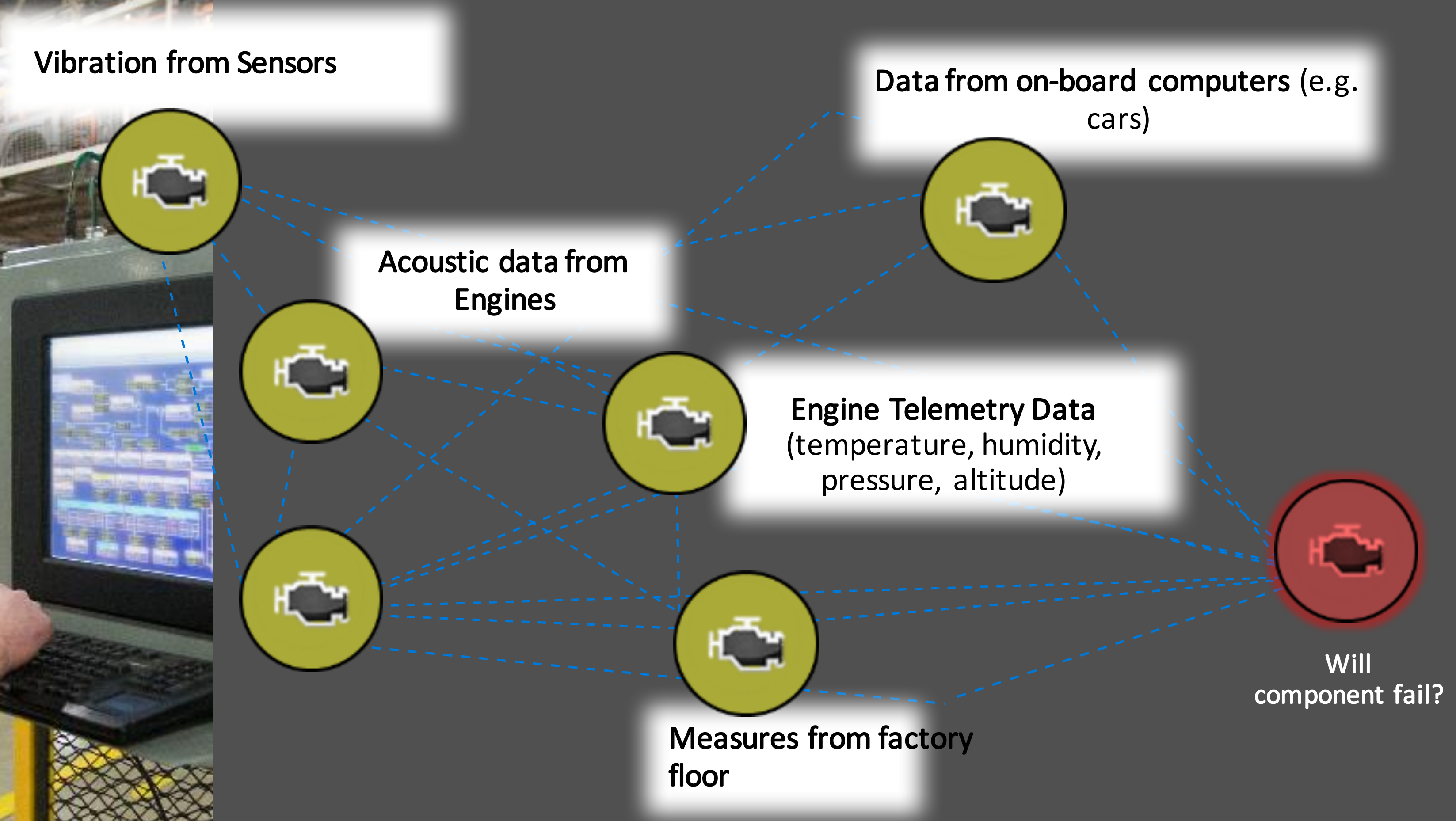
What is Predictive maintenance?

Techniques that monitor equipment in-service to predict the best time for maintenance.

Common Scenarios for Predictive Maintenance

- Predicting *if* a given component will fail or not before it happens
- Predicting the most likely *causes* of failure
- Predicting *when* a component will fail
- Predicting the *yield failure* on a manufacturing plant



Features for **Predictive Maintenance**

Business Problem

- Build a model that predicts yield failure on semiconductor manufacturing plant, and

- Through your analysis determine the factors that lead to yield failures in the process

The Data

- SECOM data set University of California at Irvine's Machine Learning Repository
- This dataset contains 1,567 examples, each with 591 features
- Of the 1567 examples, 104 of them represent yield failures
- The columns represent sensor readings from 591 points in the manufacturing process.
- Plus the yield result (i.e. a simple pass or fail) for each example

Model_PdM2 › SECOM_Dataset_AllFeatures.csv › dataset

rows 1576

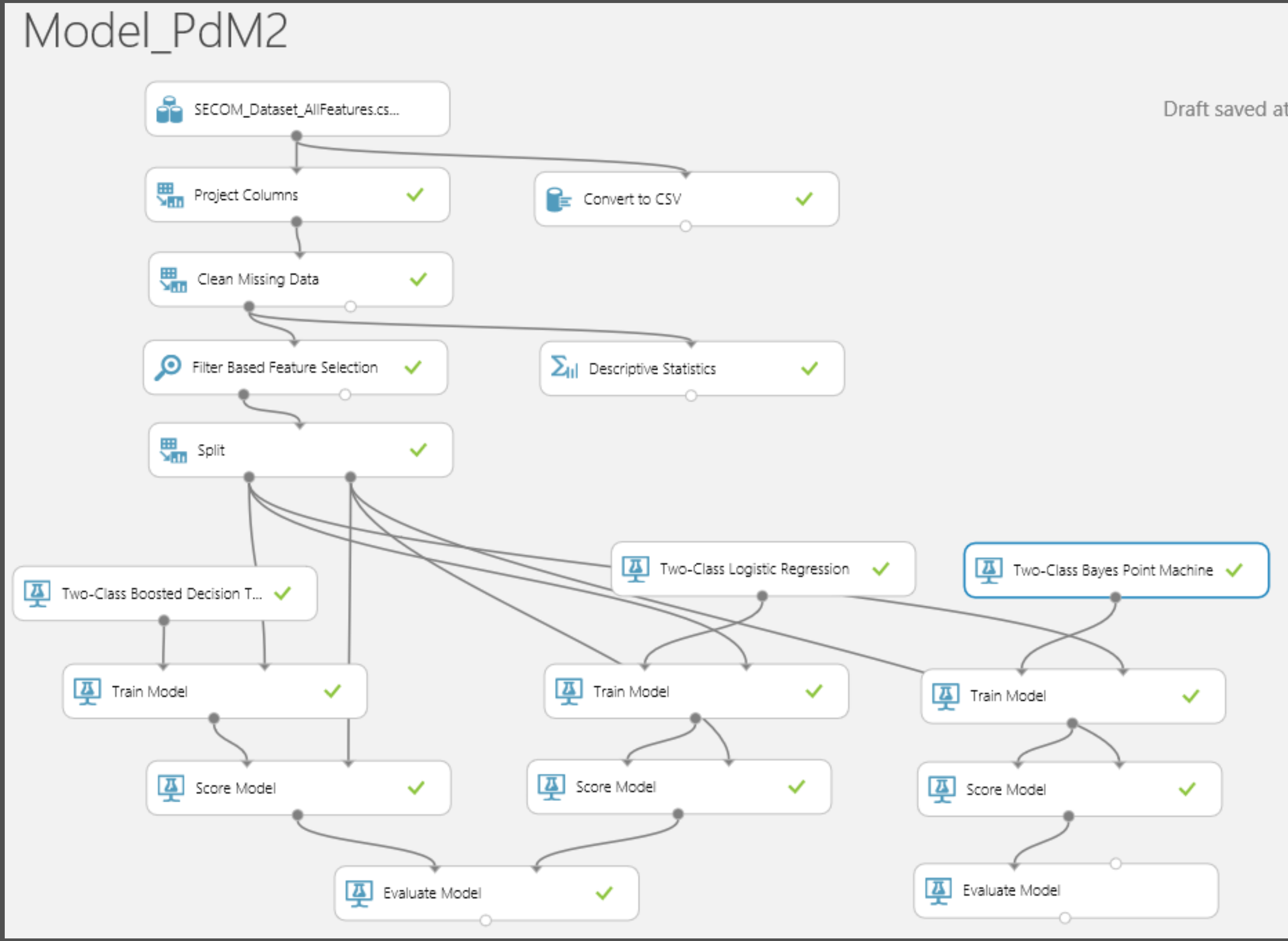
columns 592

view as

Sensor#1	Sensor#2	Sensor#3	Sensor#4	Sensor#5	Sensor#6	Sensor#7	Sensor#8
3030.93	2564	2187.7333	1411.1265	1.3602	100	97.6133	0.1242
3095.78	2465.14	2230.4222	1463.6606	0.8294	100	102.3433	0.1247
2932.61	2559.94	2186.4111	1698.0172	1.5102	100	95.4878	0.1241
2988.72	2479.9	2199.0333	909.7926	1.3204	100	104.2367	0.1217
3032.24	2502.87	2233.3667	1326.52	1.5334	100	100.3967	0.1235
2946.25	2432.84	2233.3667	1326.52	1.5334	100	100.3967	0.1235
3030.27	2430.12	2230.4222	1463.6606	0.8294	100	102.3433	0.1247
3058.88	2690.15	2248.9	1004.4692	0.7884	100	106.24	0.1185
2967.68	2600.47	2248.9	1004.4692	0.7884	100	106.24	0.1185
3016.11	2428.37	2248.9	1004.4692	0.7884	100	106.24	0.1185
2994.05	2548.21	2195.1222	1046.1468	1.3204	100	103.34	0.1223

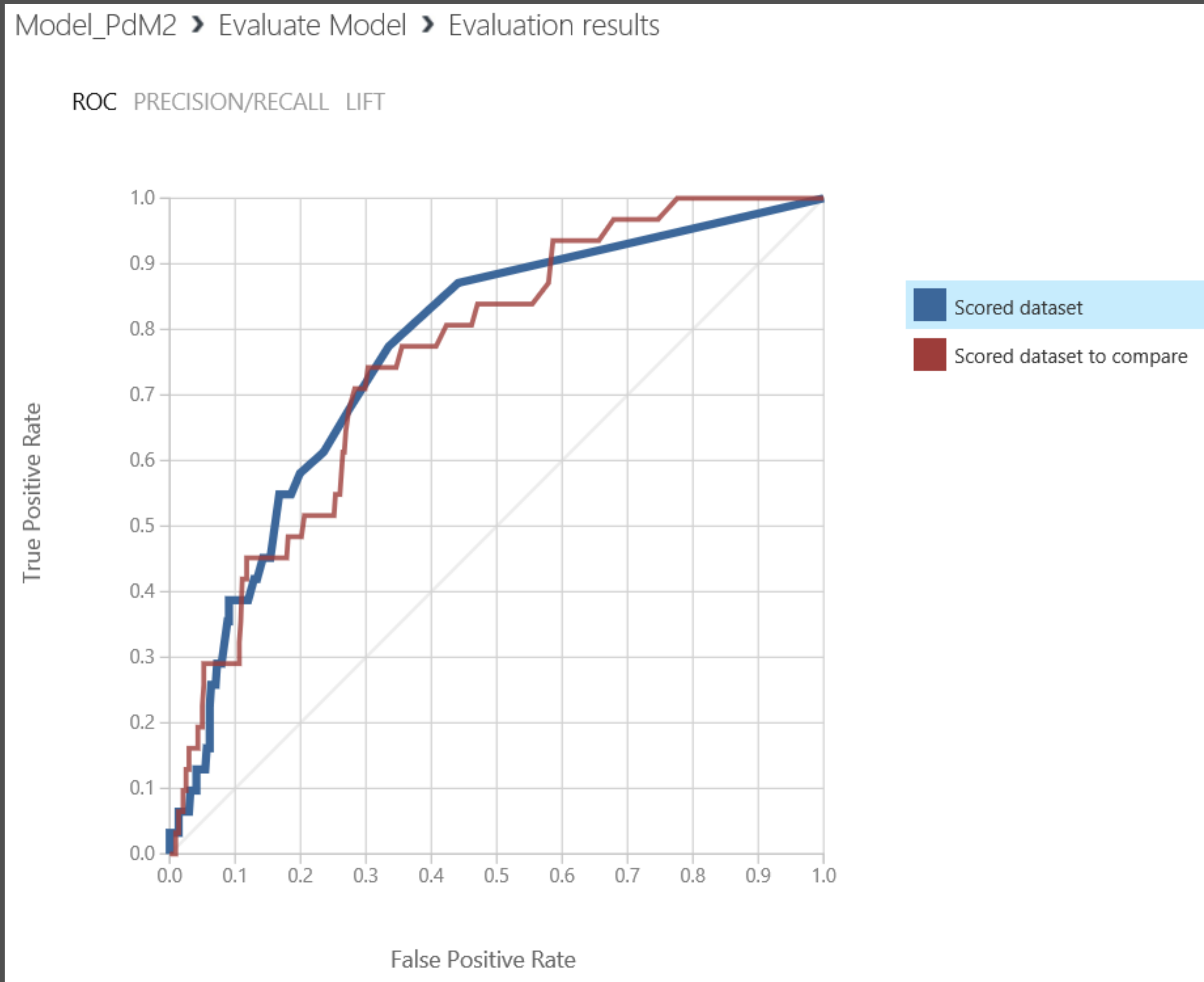
Potentials to Deeply Engaged Models

- Input dataset: SECOM data
- Final dataset had 591 input variables
- Each input is a sensor reading from manufacturing plant
- Three models were built with
  - Boosted Decision Trees in Azure ML
  - Logistic Regression in Azure ML and
  - Bayes Point Machine in Azure ML



Results

- Boosted Decision Tree was the most predictive with
  - AUC of 77.6%
  - Accuracy = 93%
  - Precision = 25%
  - Recall = 3.2% and
  - F1 Score = 5.7%



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
1	30	0.930	0.250	0.5	0.776
False Positive	True Negative	Recall	F1 Score		
3	439	0.032	0.057		

Demo

Building the Predictive Model

How Good is the Model?

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
1	30	0.930	0.250	0.5	0.776
False Positive	**True Negative**	**Recall**	**F1 Score**		
3	439	0.032	0.057		

- Accuracy is very high: 93%
- *BUT:*
  - Precision = 25%
  - Recall = 3.2% and
  - F1 Score = 5.7%
- The model is a good start, but not great!

Why?

- Class imbalance!
- The SECOM data only has 6.6% of yield failures
- It is much harder to train algorithms to learn the minority class with serious class imbalance
- Accuracy is not a good measure of success with class imbalance
  - Because the ML algorithm can show high accuracy by simply learning majority class e.g.
  - A naïve model that predicts “pass” for all cases will have accuracy of 93.4%!
- So 93% accuracy from our model is not great!

The Solution

1. Up-sampling
   - Increase proportion of minority class in your sample

2. Down-sampling
   - Decrease proportion of majority class in your sample

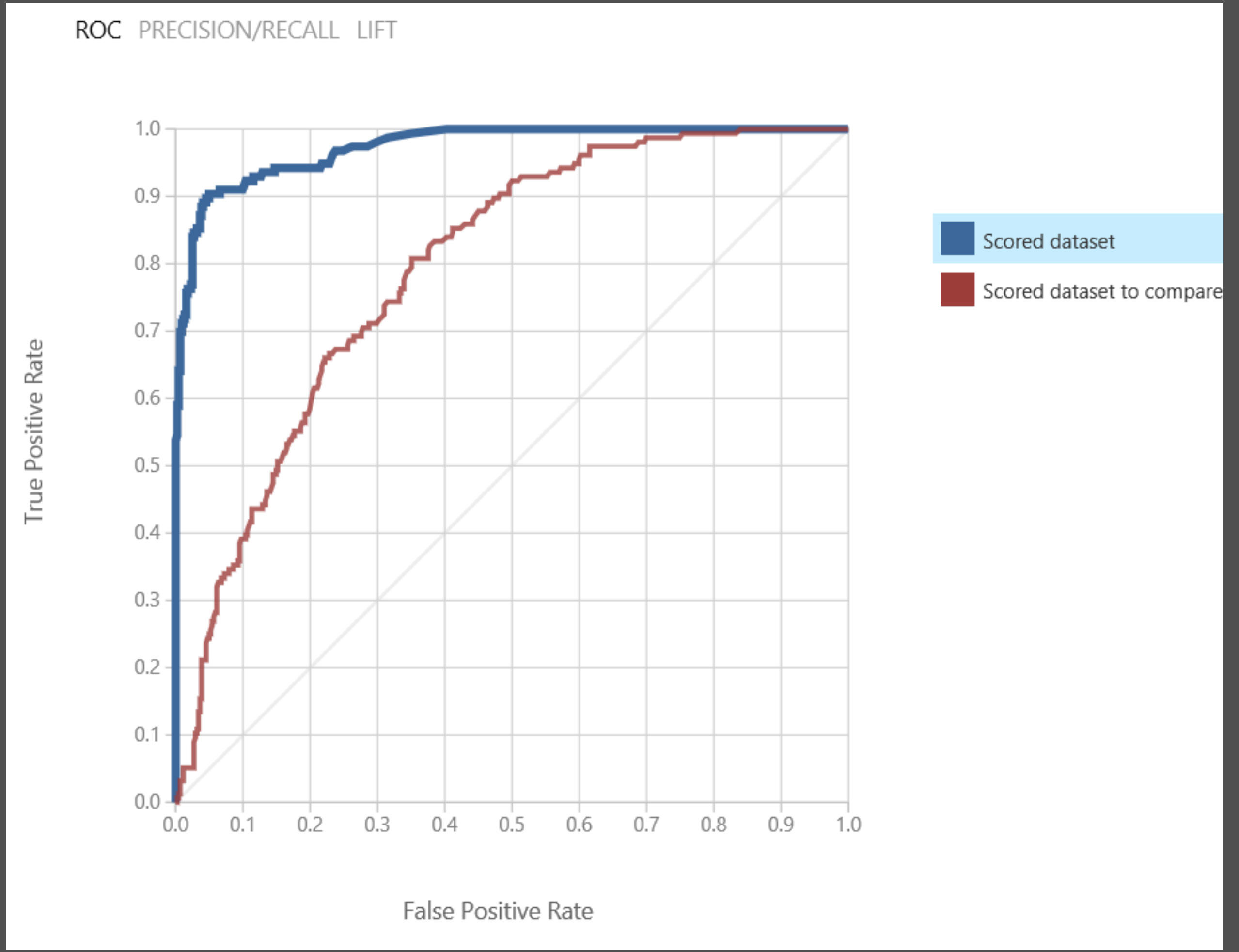
Demo

Improving Predictive Models with Up or Down-sampling

Results: Up-Sampling with SMOTE Algorithm

- Boosted Decision Tree was the most predictive model

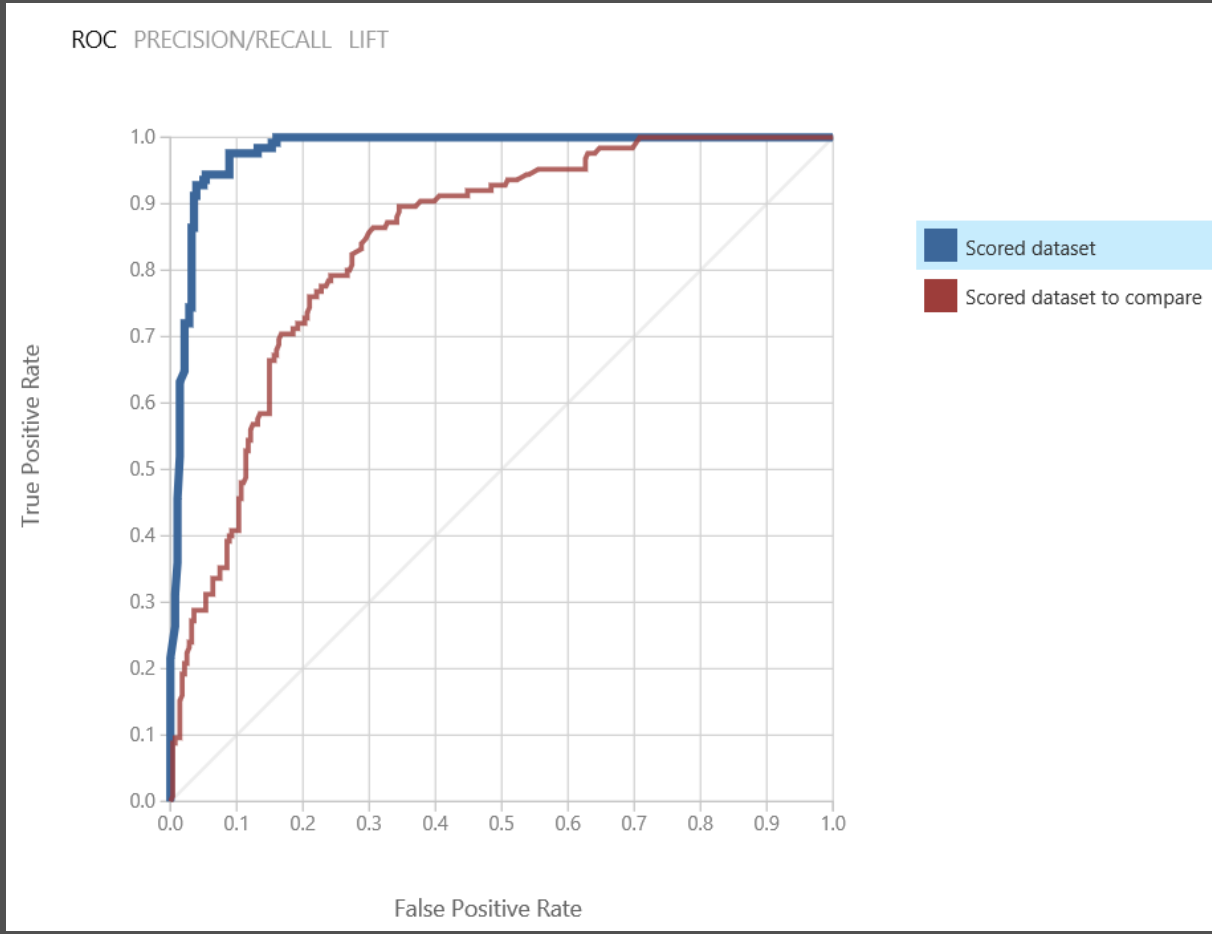
Metric	Before Resampling	After Resampling
Area under Curve (AUC)	77.6%	97.4%
Accuracy	93%	94%
Precision	25%	92.3%
Recall	3.2%	84%
F1 Score	5.7%	87.9%



Up and Down-Sampling with SMOTE in R

- Boosted Decision Tree was the most predictive model
- Also had higher recall rates and F1 score than simple SMOTE model

Metric	Before Resampling	After Resampling
Area under Curve (AUC)	77.6%	98.7%
Accuracy	93%	94.6%
Precision	25%	89.9%
Recall	3.2%	92.8%
F1 Score	5.7%	91.3%



Summary

1. Built models for predicting yield failure in semiconductor manufacturing process

2. Initial model was good (with AUC of 77.%), but impacted by class imbalance

3. Imbalance problem was addressed by
   - Up-sampling with SMOTE algorithm in Azure ML
   - Further improvements through up- and down-sampling with SMOTE Algorithm in R

4. Final model had significantly better performance after up- and down-sampling