DataRobot
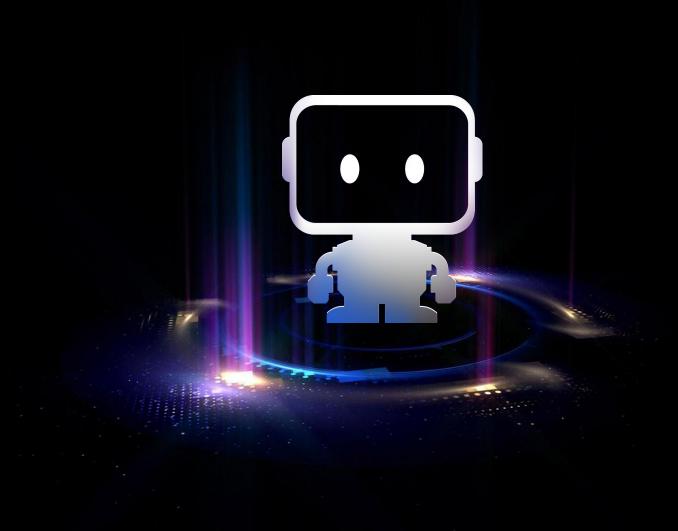
Massive Business Impact from
Extreme Automation

Speed & Scale

$1Bn Invested

Most comprehensive end-to-end AI platform on the market

750 + Data Scientists and Engineers

Applied ML experience

Extensible Platform

Models from Code, GUI or Converted from Legacy, Deployed and Operated Anywhere

For Everyone

From Business Users, to IT From novice to expert

Clear and Trusted

Business outcomes driven by understandable AI
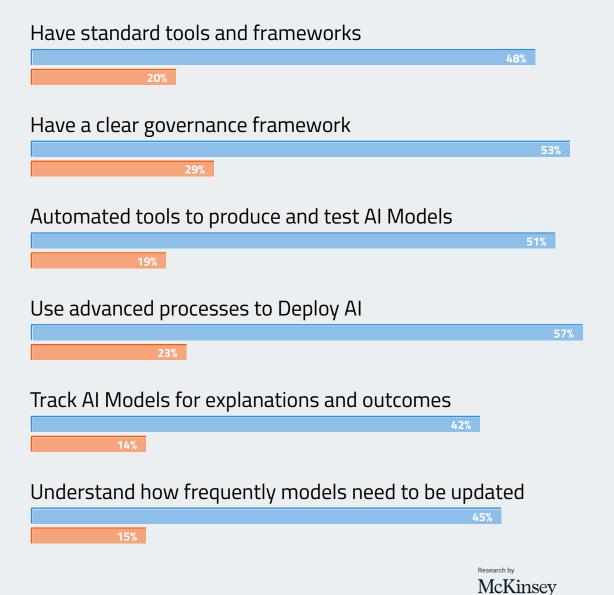
Scale

Over one million projects and greater than one trillion predictions

+ Category Creator of AutoML and MLOps

Realizing the benefits from AI

HIGH PERFORMERS

EVERYONE ELSE

Have standard tools and frameworks
- 48%
- 20%

Have a clear governance framework
- 53%
- 29%

Automated tools to produce and test AI Models
- 51%
- 19%

Use advanced processes to Deploy AI
- 57%
- 23%

Track AI Models for explanations and outcomes
- 42%
- 14%

Understand how frequently models need to be updated
- 45%
- 15%
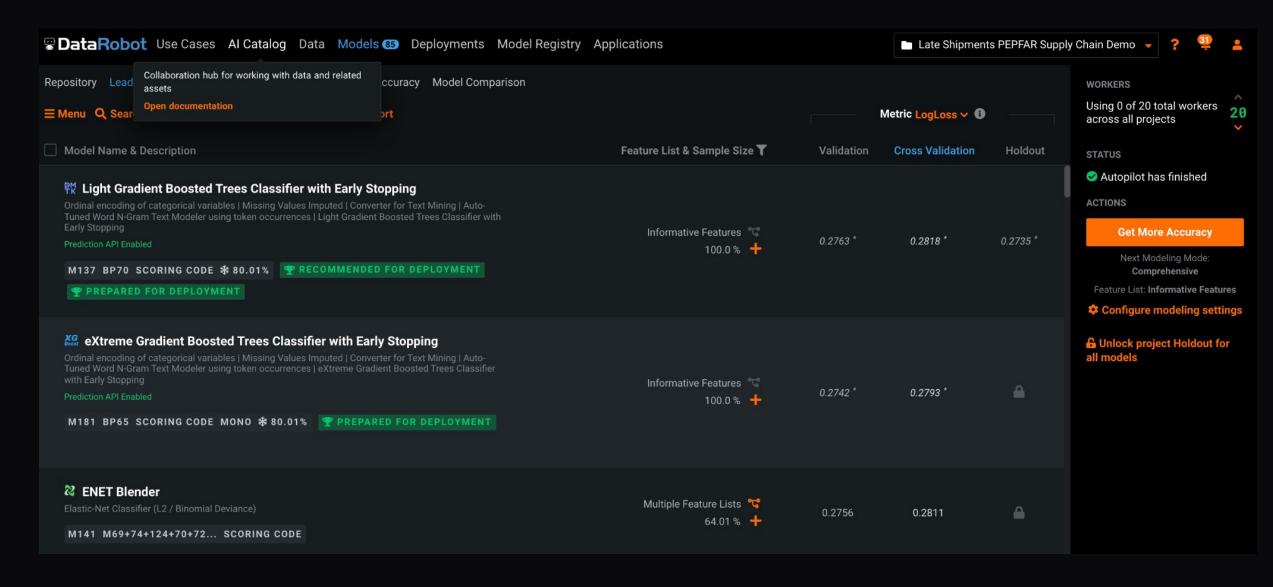
Research by
McKinsey & Company

+ DEMONSTRATION

Six Use Cases & Over 200 Models in an Afternoon

1. Predict late delivery of critical medical aid shipments - Binary Classification

2. Predict home prices in Utah - Regression with Multi-model Data

3. Classify leaf images as healthy or one of 10 plant diseases - Multiclass Classification

4. Detect anomalies in network intrusion monitoring systems - Anomaly Detection

5. Predict Sales at 10 stores for the next 7 days - Time Series Forecasting with derivation of features

6. Predict banana purchases - Binary Classification with derivation of features
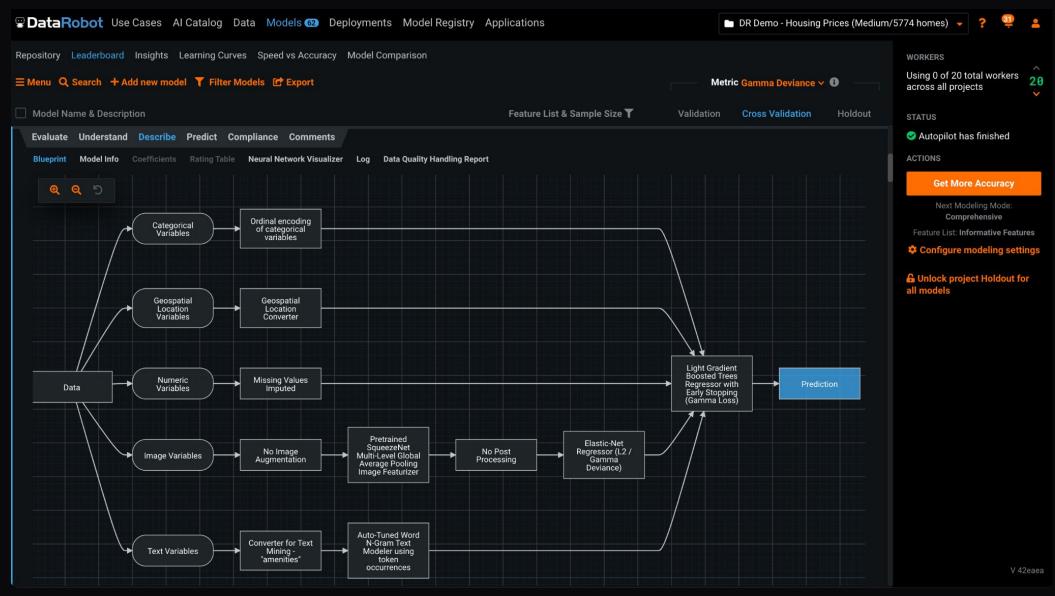
Late Shipments for Medical Aid

Shipment Scenario Planner - No Code AI App

DataRobot

Simulate | **Compare (3)**

| Scenario **"Late_delivery"**	Scenario 2 **No (81%)**	Scenario 3 **Yes (24%)**	Scenario 5 **No (95%)**
	+++ Brand	+++ Brand	+++ Brand
	+++ Dosage	+++ Dosage	+++ Dosage
	+++ Vendor INCO Term	+++ Vendor INCO Term	+++ Vendor INCO Term
	+++ Pack Price	+++ Pack Price	+++ Pack Price
	+++ Unit Price	+++ Shipment Mode	––– Vendor
Inputs			
Vendor	Aurobindo Pharma Limited	Aurobindo Pharma Limited	HETERO LABS LIMITED
Item Description	Zidovudine 300mg	Zidovudine 300mg	Zidovudine 300mg
Molecule/Test Type	Zidovudine	Zidovudine	Zidovudine
Pack Price	7	7	7
Line Item Quantity	500	5000	5000
Brand	Generic	Generic	Generic
Dosage	300mg	300mg	300mg
Line Item Insurance (USD)	100	1000	1000
Product Group	ARV	ARV	ARV
Sub Classification	Adult	Adult	Adult
Managed By	PMO - US	PMO - US	PMO - US
First Line Designation	Yes	Yes	Yes
Manufacturing Site	Micro labs, Verna, Goa, India	Micro labs, Verna, Goa, India	Hetero Unit III Hyderabad IN
Dosage Form	Tablet	Tablet	Tablet
Weight (Kilograms)	500	5000	5000
Country	Congo, DRC	Congo, DRC	Congo, DRC
Shipment Mode	Air	Air	Air
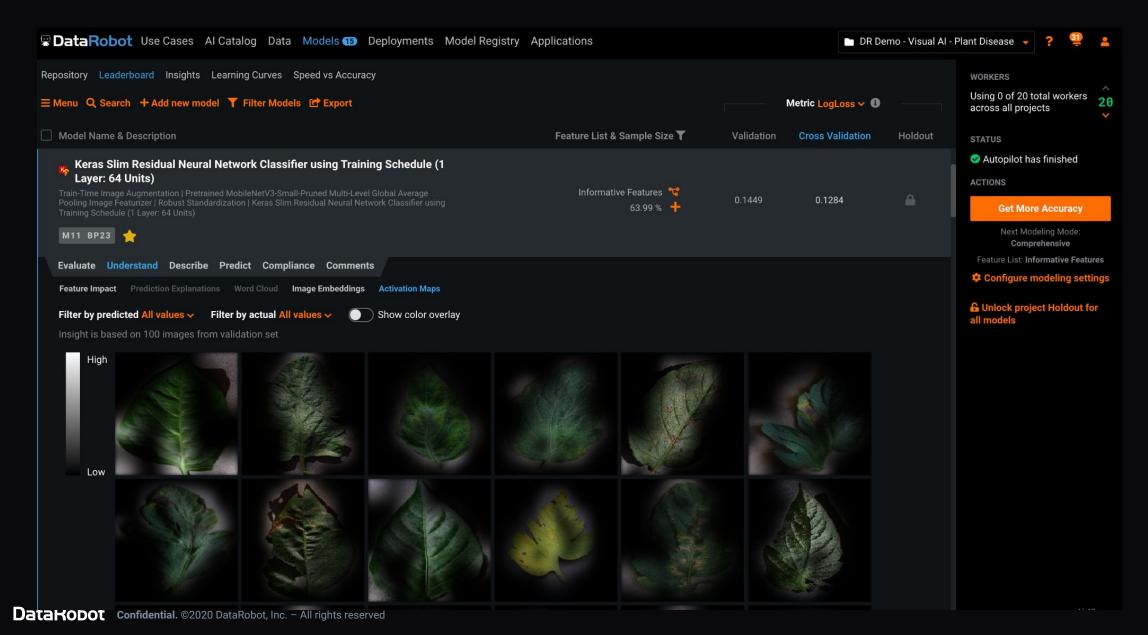
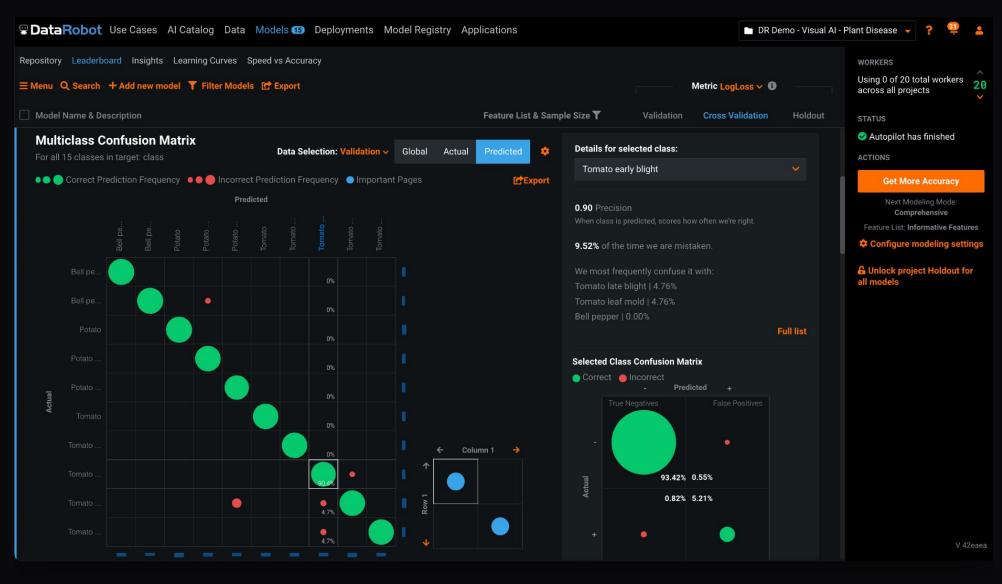# Home Prices using Text, Images and Geospatial Inputs

Classifying Fifteen Plant Diseases from Leaf Images
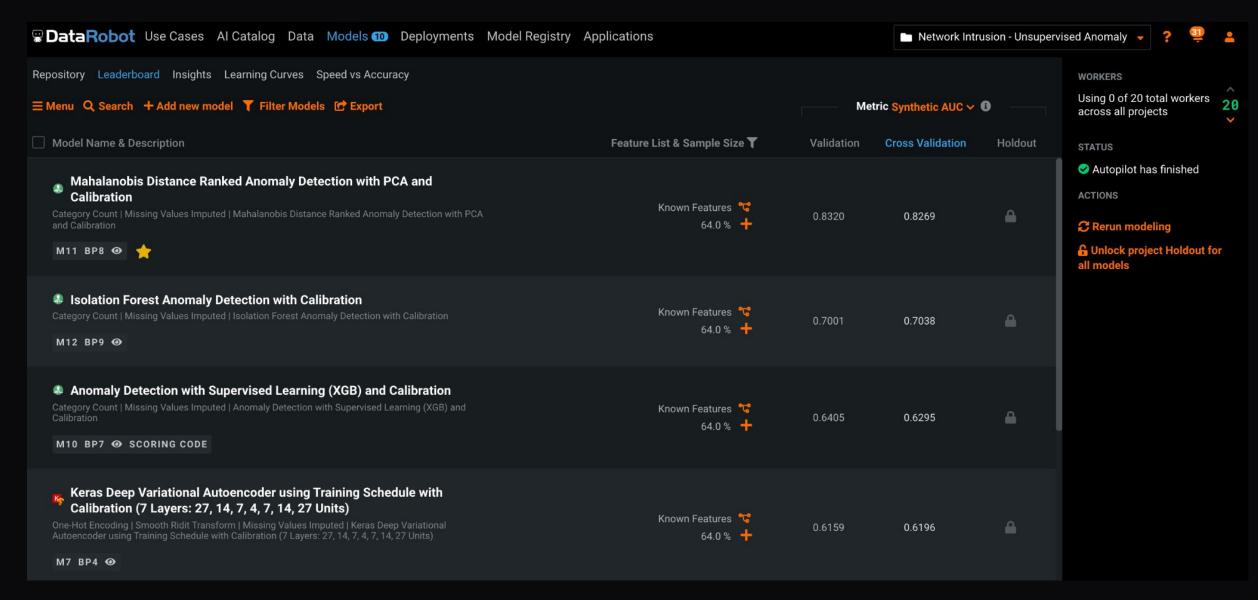
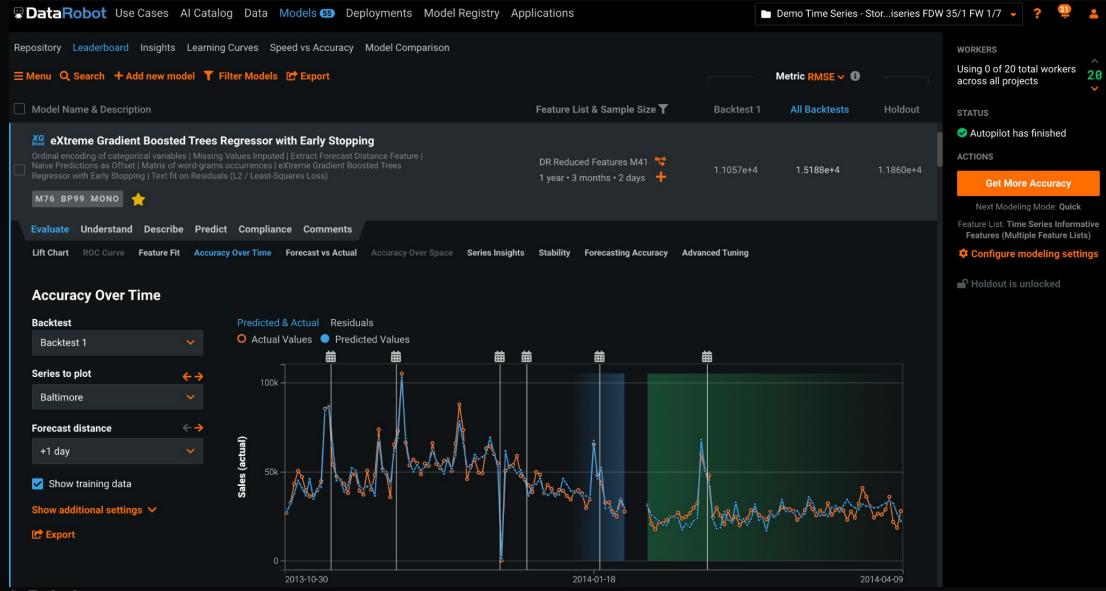# Classifying Fifteen Plant Diseases from Leaf Images
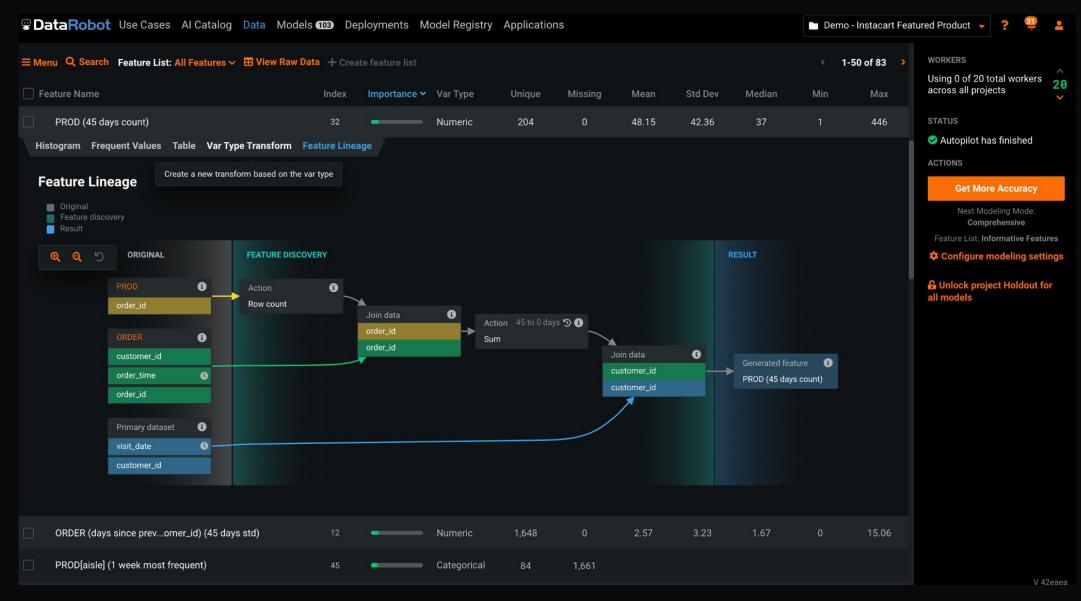
Network Intrusion Anomaly Detection

Store Sales at 10 Locations for Next 7 Days

Grocery Purchase Propensity from Transaction Tables

+ MORE SCALE AND SPEED

Model Factory Set-up

```
In [14]:  project_prefix = 'TS Fctry'
          date_col = 'Date'
          series_id = 'Store'
          target = 'Sales'
          metric = 'RMSE'

In [17]:  def run_project(dataID, derive_start, derive_end, forecast_start, forecast_end):

              # create a project
              project = dr.Project.create_from_dataset(project_name=f'{project_prefix} Derive {derive_start}/{derive
                                                       dataset_id=dataID)

              # define time-series project specs
              ts_specs = dr.DatetimePartitioningSpecification(
                  datetime_partition_column=date_col,
                  use_time_series=True,
                  multiseries_id_columns=[series_id],

                  feature_derivation_window_start=derive_start,
                  feature_derivation_window_end=derive_end,
                  forecast_window_start=forecast_start,
                  forecast_window_end=forecast_end,

                  validation_duration='P0Y0M60DT0H0M0S',   #backtesting period of 60 days
                  number_of_backtests = 2)

              # kickoff autopilot
              project.set_target(
                  target=target,
                  metric=metric,
                  partitioning_method=ts_specs,
                  worker_count=2)                           #2 workers per project

              return project
```
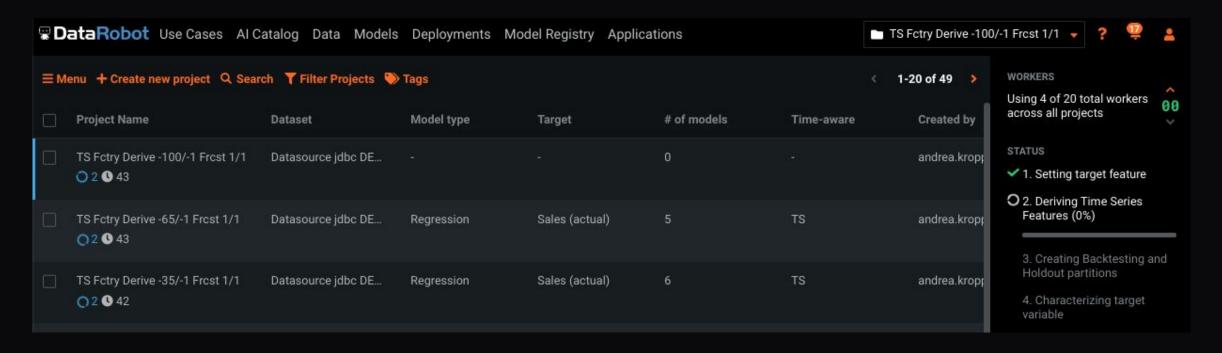
Function that takes derivation windows and forecast distances as inputs.

Same dataset and backtest period for all experiments.

Two DataRobot workers allocated per project.

Model Factory Run

```
In [18]:  #Experiment A - To forecast the next day very accurately is it better to use        eatures?

          run_project(sourcedata, -35, -1, 1, 1)
          run_project(sourcedata, -65, -1, 1, 1)
          run_project(sourcedata, -100, -1, 1, 1)

Out[18]:  Project(TS Fctry Derive -100/-1 Frcst 1/1)
```

```
In [19]:  #Experiment B - To forecast 7 to 13 days out, is it better to use 35, 65 or 100 days of lagged features?

          #Forecasting 7 to 13 days out is useful to set staffing levels and provide employee schedules a week in advance.

          projB35 = run_project(sourcedata, -35, -1, 7, 13)
          projB65 = run_project(sourcedata, -65, -1, 7, 13)
          projB100 = run_project(sourcedata, -100, -1, 7, 13)
```

```
In [*]:   #Experiment C - Using no information from the last 7 days, how well can we forecast the upcoming week?
          #Is it better to forecast each day individually in a project or create a single project for the week

          #Forecasting without the most recent information is relevant if you receive batch data feeds from partners

          projC65_1 = run_project(sourcedata, -65, -7, 1, 1)
          projC65_2 = run_project(sourcedata, -65, -7, 2, 2)
          projC65_3 = run_project(sourcedata, -65, -7, 3, 3)
          projC65_123 = run_project(sourcedata, -65, -7, 1, 3)
```

Ten DataRobot projects initiated. Progress can be monitored in the UI. These will complete in about 2.5 hours with about 500 models in total.

Mass Model Automation

Mass Model Automation

Mass Model Automation

Aggregate Results across Experiments

Create a summary dataframe with all models across all projects (~500)

Create a summary dataframe with the lowest MAE in each project (10)

```python
In [6]: all_scores = pd.DataFrame()
        minimum_scores = pd.DataFrame()

        for proj in proj_list:

            minimum_project_score = pd.DataFrame()
            project_scores = pd.DataFrame()

            #get all models for a project; most of them have 50 models
            model_list = proj.get_datetime_models()

            for m in model_list:

                backtest_scores = pd.DataFrame(
                    [
                        {
                            'Project_Name': proj.project_name,
                            'Model_Type': m.model_type,
                            'Featurelist': m.featurelist_name,
                            'Backtest_MAE': m.metrics['MAE']['backtesting'],
                            'Backtest_MASE': m.metrics['MASE']['backtesting'],
                            'Backtest_RMSE': m.metrics['RMSE']['backtesting'],
                            'Model_ID': m.id,
                        }
                    ]
                )

                project_scores = project_scores.append(backtest_scores, sort=False).reset_index(drop=True)

            minimum_project_score = project_scores[project_scores.Backtest_MAE == project_scores.Backtest_MAE.min()]

            #gather the minimum MAE from each project
            minimum_scores = minimum_scores.append(minimum_project_score, sort=False).reset_index(drop=True)

            #gather all models across all projects
            all_scores = all_scores.append(project_scores, sort=False).reset_index(drop=True)
```

Loop across the ten completed projects and retrieve all the models into a dataframe called 'all' and the best model from each project into a dataframe called 'minimum'.

Examine Top Performing Model Types

```
In [7]:  #dataframe containing the lowest MAE model from each project
         minimum_scores
```
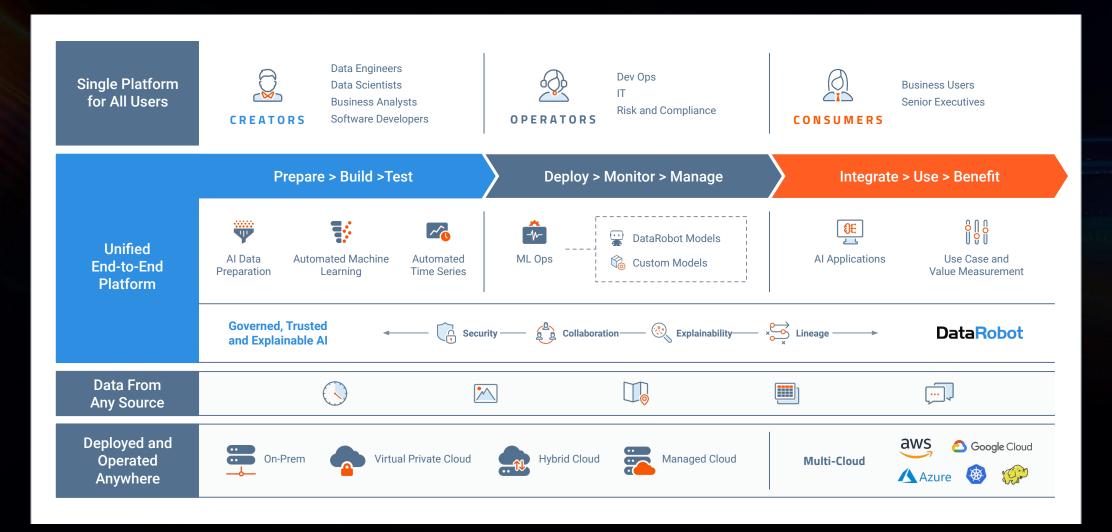
Out[7]:

	Project_Name	Model_Type	Featurelist	Backtest_MAE	Backtest_MASE	Backtest_RMSE	
0	TS Fctry Derive -65/-7 Frcst 1/3	AVG Blender	With Differencing (average baseline)	10820.38350	0.61737	20142.38791	
1	TS Fctry Derive -65/-7 Frcst 3/3	Keras DeepAR LSTM Regressor with Student's T P...	Baseline Only (latest)	12360.58726	0.67644	22856.17261	
2	TS Fctry Derive -65/-7 Frcst 2/2	Eureqa Generalized Additive Model (250 Generat...	With Differencing (average baseline)	12152.50138	0.68710	23456.95418	
3	TS Fctry Derive -65/-7 Frcst 1/1	AVG Blender	With Differencing (7 day)	11856.36516	0.70188	20970.97479	
4	TS Fctry Derive -100/-1 Frcst 7/13	AVG Blender	No Differencing	11003.92908	0.60493	20841.73936	
5	TS Fctry Derive -65/-1 Frcst 7/13	AVG Blender	With Differencing (nonzero average baseline)	10818.51987	0.58817	19717.89076	617612dc2aee95eb2281674a
6	TS Fctry Derive -35/-1 Frcst 7/13	Zero-Inflated eXtreme Gradient Boosted Trees R...	DR Reduced Features M46	9972.72674	0.54300	18493.85678	6176120934b94bd35a18b9bc
7	TS Fctry Derive -100/-1 Frcst 1/1	Performance Clustered eXtreme Gradient Boosted...	DR Reduced Features M17	9264.97918	0.60343	18864.50680	61760efa2193c115989ea5eb
8	TS Fctry Derive -65/-1 Frcst 1/1	AVG Blender	With Differencing (average baseline)	9358.14896	0.60745	18653.02359	61760db715cfc0e16418b988
9	TS Fctry Derive -35/-1 Frcst 1/1	AVG Blender	With Differencing (latest)	9552.11368	0.61496	18612.55056	617607ace86af0b9558fae28

Compare the best models (lowest MAE) from each project.

Notice that the top-performing model type and feature list varies greatly across experiments.

Notice that ensemble models (Blenders) are often the top performer.

PLATFORM OVERVIEW

AI Cloud Platform

Single Platform for All Users

CREATORS
Data Engineers
Data Scientists
Business Analysts
Software Developers

OPERATORS
Dev Ops
IT
Risk and Compliance

CONSUMERS
Business Users
Senior Executives

Prepare > Build > Test | **Deploy > Monitor > Manage** | **Integrate > Use > Benefit**

Unified End-to-End Platform

AI Data Preparation
Automated Machine Learning
Automated Time Series
ML Ops
DataRobot Models
Custom Models
AI Applications
Use Case and Value Measurement

Governed, Trusted and Explainable AI
Security — Collaboration — Explainability — Lineage
DataRobot

Data From Any Source

Deployed and Operated Anywhere
On-Prem
Virtual Private Cloud
Hybrid Cloud
Managed Cloud
Multi-Cloud
aws
Google Cloud
Azure