

DataRobot

User Guide

Contents

Introduction.....	4
QuickStart.....	5
How To.....	11
Create a Project.....	11
Select a Target Feature.....	12
Select an Optimization Metric.....	13
Set Advanced Modeling Parameters.....	14
Start Modeling.....	17
View the Model Leaderboard.....	22
View Model Blueprints.....	23
View a Model Lift Chart.....	25
View a Model ROC Curve.....	26
Predict Against a Model.....	30
Managing Projects.....	34
Create a Project.....	34
Rename a Project.....	35
Switch to a Project.....	37
Copy a Project.....	38
Share a Project.....	39
Unlock Holdout.....	42
Working with Datasets.....	45
Viewing Data.....	45
Dataset Guidelines.....	47
Working with Feature Lists.....	48
Creating Derived Features.....	50
Running Models.....	52
Select a Target Feature.....	52
Select an Optimization Metric.....	52
Set Advanced Modeling Parameters.....	53
Start Modeling.....	56
Add a New Model.....	61
Create a Blended Model.....	63
Run Selected Models.....	65
DataRobot Prime.....	67
Delete Models.....	68
Evaluating Models.....	70
The Model Leaderboard.....	70

Blueprint.....	71
Lift Chart.....	73
Model X-Ray.....	74
Model Info.....	75
Model Log.....	76
ROC Curve.....	77
Grid Search.....	81
Deploy Model.....	82
Predict Against a Model.....	83
Learning Curves.....	86
Speed Vs. Accuracy.....	87
Comparison.....	88
Insights.....	89
Variable Importance.....	89
Variable Effects.....	90
Text Mining.....	91
Word Cloud.....	93
Hotspots.....	94
RStudio IDE.....	97
Python IDE.....	98
Repository.....	99

Introduction

Overview

The DataRobot platform enables data scientists to build highly accurate predictive models, and to build them orders of magnitude faster than using traditional methods.

Building an accurate predictive model generally requires evaluating and selecting numerous data transformations, features, algorithms, and tuning parameters. DataRobot simplifies the model development process by performing a parallel heuristic search for the best model or combination of models for the applicable dataset and prediction target.

By cost-effectively evaluating thousands of models in parallel across a large cluster of servers, DataRobot delivers the best predictive model in the shortest amount of time.

About this Guide

Title: DataRobot User Guide

Version: 2.0

Publication date: 7-27-15

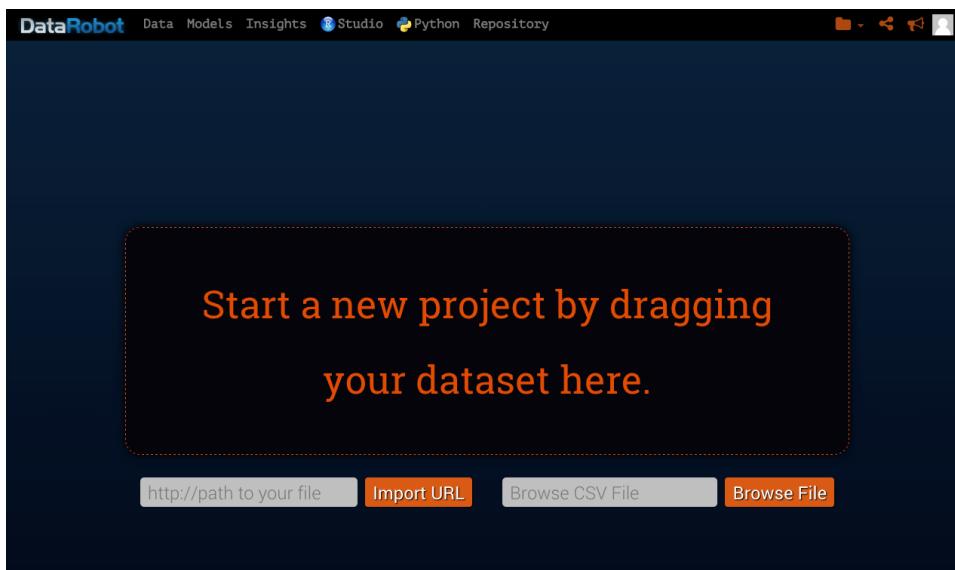
© 2015 DataRobot. All Rights Reserved.

QuickStart

In order to build models in DataRobot, you must create a project with a dataset. After you create a project, you can select a target feature and start the modeling process. A DataRobot project contains all of the models you build for the project dataset.

Use the following steps to begin modeling data with the DataRobot API.

1. You can use any of the following methods to create a new DataRobot project:
 - Drag a .csv data file to the orange box on the DataRobot platform home page.
 - Click **Browse File**, then use the file browser to select a .csv data file.
 - Type a URL in the box labeled "http://path to your file", then click **Import URL**.



2. A progress indicator is displayed while the file is being processed.

The screenshot shows the DataRobot interface with a dark theme. At the top, there's a navigation bar with links like Data, Models, Insights, Studio, Python, and Repository. Below it, a summary section asks "Tell us what you'd like to predict:" with a text input field containing "Enter Name of Target". To the right is a large, prominent "Push to Start" button. In the center, a progress bar indicates the modeling process: Step 1: Uploading Data (4.987 sec.) ✓, Step 2: Reading raw data (11.203 sec.) ✓, Step 3: Converting file (2.452 sec.) ✓, Step 4: Sampling data 0% Complete, Step 5: Saving files and Setting up EDA workers, and Step 6: Exploratory Data Analysis. On the left, a sidebar lists features: chlorpropamide, race, age, weight, admission_type_id, discharge_disposition_id, admission_source_id, time_in_hospital, and payer_code. At the bottom, it says "Creating Project" and "Current Feature List: 0 Active Features".

- To begin modeling, type in the name of the target feature. The target feature is the name of the column in the dataset you would like to predict.

DataRobot provides a preselected optimization metric based on the modeling task represented by the target feature (regression or classification). To select a different optimization metric, click **Or Select from Advanced Metrics**.

After specifying the target feature, click **Push to Start** to start modeling. The modeling process will run in Automatic mode by default.

This screenshot shows the DataRobot interface after selecting the target feature "readmitted". The "Push to Start" button is circled in red. The "Select Metric to Optimize" section shows "Recommended : LogLoss (Accuracy)" with the note "Measures Inaccuracy of Predicted Probabilities". Below it is a "Bar Chart" showing the frequency of "Yes" and "No" values. The "Show Advanced Options" dropdown is open, showing "All Features" and "Select a Feature List". The main feature list table shows the following columns: Feature Name, Var Type, Unique, Missing, Mean, SD, Median, Min, and Max. The listed features include diag_1_desc, diag_2_desc, diag_3_desc, race, gender, age, weight, admission_type_id, and discharge_disposition_id. At the bottom, it says "10k_diabetes.xlsx Total features: 51, Datapoints: 10,000 Target: readmitted".

4. The status of running models is displayed in the Worker Queue at the right of the screen. Depending on the size of the dataset, it may take several minutes to complete the modeling process.

The screenshot shows the DataRobot interface with the 'Models' tab selected. On the left, a table lists features from the '10k_diabetes.xlsx' file, including columns for Feature Name, Importance, Var Type, Unique, Missing, Mean, SD, Median, and Min. On the right, a sidebar titled 'Workers: 002' displays the 'Processing (2)' and 'Queue (5)' sections. Each entry in the queue includes a thumbnail, progress bar, and resource usage (CPU and RAM).

5. The results of the modeling process are displayed in the model Leaderboard, with the best models (based on the chosen performance metric) at the top of the list.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. The table lists various models along with their descriptions, feature counts, validation metrics, and holdout status. A sidebar on the right provides options like 'Run Autopilot On A Different Feature List', 'Unlock Holdout', and 'View Summary'. A 'Datarobot Prime' icon is also present.

6. Click a model to display the model blueprint, lift chart, and other information.

The screenshot shows the DataRobot interface with the "Autopilot has finished" section visible on the right. The main area displays a complex model blueprint consisting of multiple parallel processing paths. Each path involves various preprocessing steps (e.g., Model 1A, Model 1B, Model 1C) leading to different classifiers (e.g., Logistic Regression, Random Forest, SVM). The final output is a "Blender" block. Below the blueprint, several model cards are listed:

- ENET Blender (6+14+17+20+24+2...) (93)
- Advanced AVG Blender (6+14+17+20+24+2...) (92)
- AVG Blender (20+26+27) (90)
- GLM Blender (20+26+27) (89)

At the bottom left, there is a note: "10k_diabetes.xlsx Total features: 51, Datapoints: 10,000 Target: readmitmed, Metric: LogLoss".

- To predict against the model, click **Predict**, then select a dataset.

The screenshot shows the DataRobot interface with the "Predict" button highlighted in red. Below it, a large dashed box labeled "Drag Additional Datasets Here." is present. At the bottom of the interface, there are four upload buttons: "http://path to your file", "Import URL", "Browse CSV File", and "Browse File".

The main area lists the same models as the previous screenshot:

- ENET Blender (6+14+17+20+24+2...) (93)
- Advanced AVG Blender (6+14+17+20+24+2...) (92)
- AVG Blender (20+26+27) (90)
- GLM Blender (20+26+27) (89)
- Nystroem Kernel SVM Classifier (20)

At the bottom left, there is a note: "10k_diabetes.xlsx Total features: 51, Datapoints: 10,000 Target: readmitmed, Metric: LogLoss".

- After the dataset has been uploaded, click **Compute Prediction** to generate predictions for the dataset.

The screenshot shows the DataRobot interface with the 'Models' tab selected. In the center, there's a table of models. Below it, a section titled 'Compute & Download Model Predictions:' lists datasets. The 'Diabetes_12.xlsx' dataset is selected, and its 'Compute Prediction' button is highlighted with a red box.

9. The **Compute Prediction** button changes to **In Queue** for the selected dataset, and the job status appears under Processing in the Worker queue.

The screenshot shows the DataRobot interface with the 'Models' tab selected. The right sidebar shows the 'Worker' queue with a section titled 'Processing (1)'. It lists a job for 'ENET Blender...' with a progress bar at 8% completion. Other workers are shown as available.

10. When the prediction has finished running, click **Download Prediction** to view the results in a .csv file. Prediction data is also retained in the project.

The screenshot shows the DataRobot interface with the 'Models' tab selected. The main area displays a list of trained models, each with its name, description, feature list, sample size, validation metrics, and deployment status. A specific model, 'ENET Blender (6+14+17+20+24+2...) (93)', is highlighted. Below the model list, there's a section titled 'Compute & Download Model Predictions:' which lists datasets and their creation times. One dataset, 'Diabetes_500.xlsx', has a 'Download Prediction' button highlighted with a red box. The right sidebar shows the status 'Autopilot has finished' and provides options like 'Run Autopilot On A Different Feature List', 'Unlock Holdout', and 'View Summary'. A 'Datarobot Prime' icon is also present.

Model Name and Description	Feature List	Sample Size	Validation	Cross Validation	Hold Out
ENET Blender (6+14+17+20+24+2...) (93)	Informative Features	64.0 %	0.6072	0.6048	Locked
Advanced AVG Blender (6+14+17+20+24+2...) (92)	Informative Features	64.0 %	0.6075	0.6054	Locked
AVG Blender (20+26+27) (90)	Informative Features	64.0 %	0.6081	0.6064	Locked
GLM Blender (20+26+27) (89)	Informative Features	64.0 %	0.6088	0.6053	Locked
Nystroem Kernel SVM Classifier (20)	Informative Features	64.0 %	0.6094	0.6106	Locked
Advanced GLM Blender (6+14+17+20+24+2...) (91)	Informative Features	64.0 %	0.6100	0.6049	Locked
Nystroem Kernel SVM Classifier (27)	Informative Features	64.0 %	0.6110	0.6097	Locked

Compute & Download Model Predictions:

Dataset	Created	Action
Informative Features	2015-06-23 18:48:35	Compute Prediction
Diabetes_500.xlsx	2015-06-24 16:35:27	Download Prediction

Autopilot has finished

- Run Autopilot On A Different Feature List
- Unlock Holdout
- View Summary

Datarobot Prime

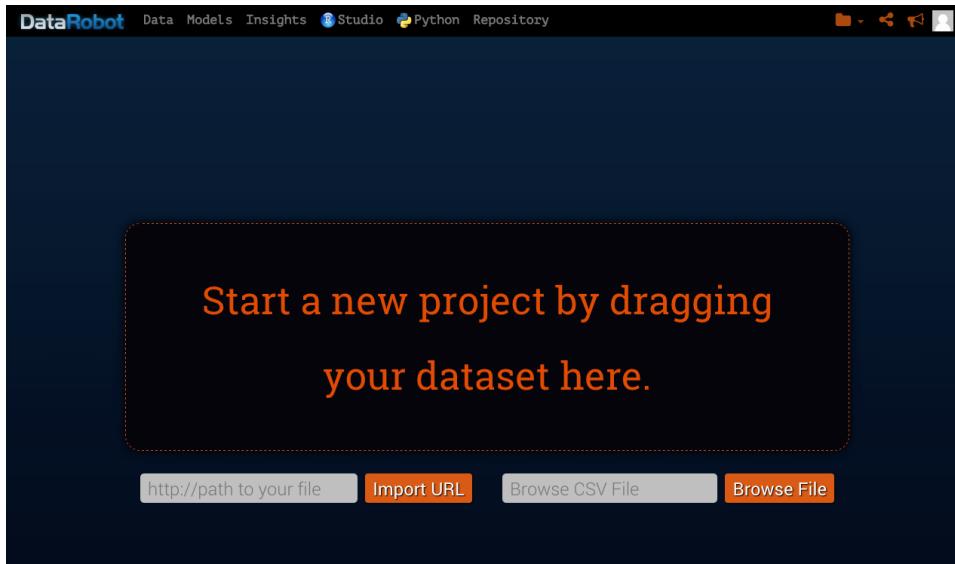
How To...

Create a Project

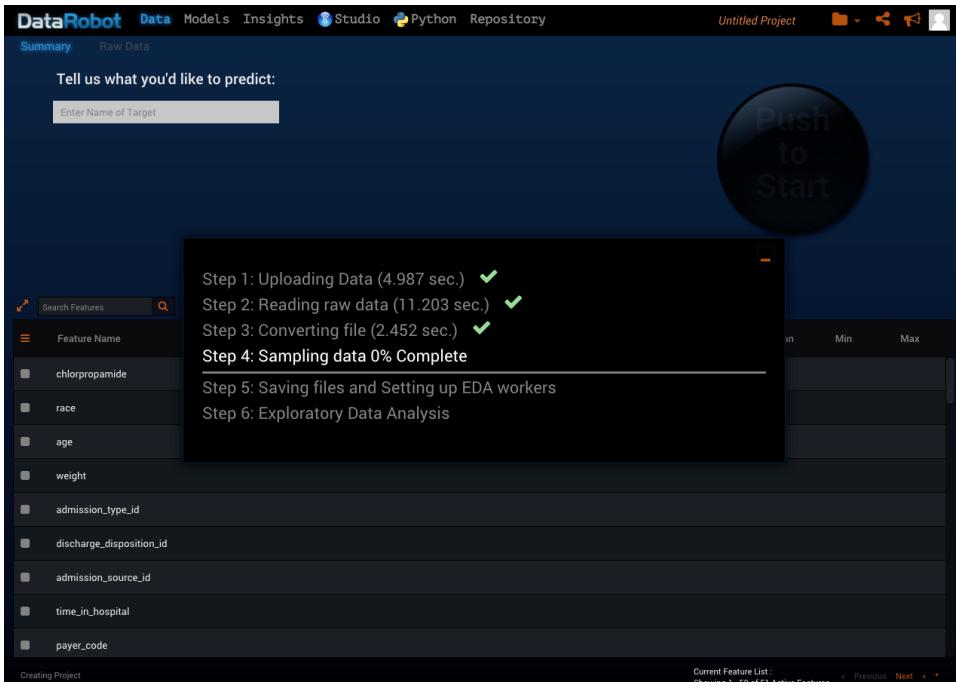
In order to build models in DataRobot, you must create a project with a dataset. After you create a project, you can select a target feature and start the modeling process. A DataRobot project contains all of the models you build for the project dataset.

Use the following steps to begin modeling data with the DataRobot API.

1. You can use any of the following methods to create a new DataRobot project:
 - Drag a .csv data file to the orange box on the DataRobot platform home page.
 - Click **Browse File**, then use the file browser to select a .csv data file.
 - Type a URL in the box labeled "http://path to your file", then click **Import URL**.



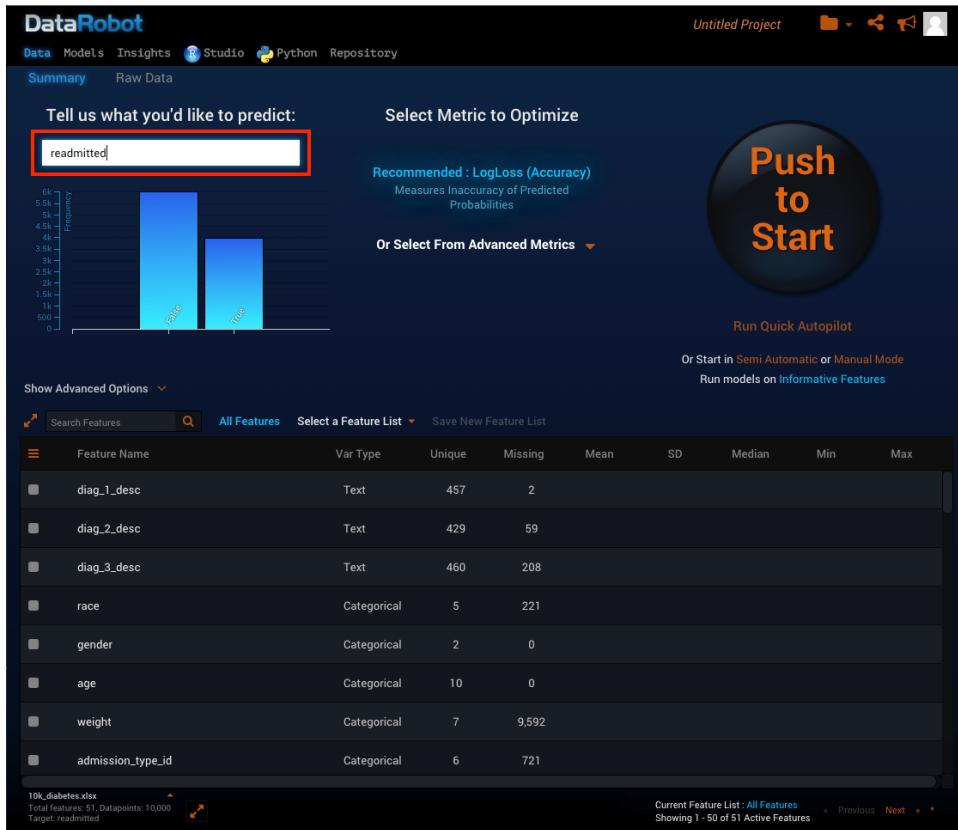
2. A progress indicator is displayed while the file is being processed.



Select a Target Feature

After you create a project, you can select a target feature and an optimization metric and start modeling. The target feature is the name of the column in the dataset you would like to predict. The optimization metric is the metric software used to optimize the models.

On the Data Summary page, type in the name of the target feature you would like to predict.



Select an Optimization Metric

After you create a project, you can select a target feature and an optimization metric and start modeling. The target feature is the name of the column in the dataset you would like to predict. The optimization metric is the metric software used to optimize the models.

After you type in a target feature, a recommended optimization metric is automatically selected based on the modeling task.

- If the selected target has only two unique values, the assumption is made that the task is a classification, and a classification metric will be recommended. Examples of recommended classification methods include LogLoss (if it is necessary to calculate a probability for each class), and Gini and AUC when it is necessary to sort records in order of ranking.
- Otherwise the assumption is made that the selected target represents a regression task. The most popular metrics for regression are RMSE (Root Mean Square Error) and MAD (Mean Absolute Deviation).

If you would like to override the recommended optimization metric, click **Select from Advanced Metrics**, then select an alternate metric.

The screenshot shows the DataRobot interface for a project titled "Untitled Project". On the left, there's a summary section with a bar chart for the target variable "readmitted". The chart shows two bars: one for "False" (approx. 5.5k) and one for "True" (approx. 3.8k). Below the chart is a list of features with their types and statistics. A dropdown menu is open under "Select Metric to Optimize", showing options like LogLoss (Accuracy), AUC (Suggested), Gini Norm (Suggested), RMSE (Suggested), FVE Binomial, Rate@Top10%, and Rate@Top5%. A large orange button on the right says "Push to Start".

Feature Name	Type	SD	Median	Min	Max
diag_1_desc	Text	429	59		
diag_2_desc	Text	460	208		
diag_3_desc	Categorical	5	221		
race	Categorical	2	0		
gender	Categorical	10	0		
age	Categorical	7	9,592		
weight	Categorical	6	721		
admission_type_id	Categorical				

Set Advanced Modeling Parameters

To set advanced modeling parameters, click **Show Advanced Options** on the Data Summary page.

The screenshot shows the DataRobot interface for an 'Untitled Project'. At the top, there are tabs for Data, Models, Insights, R Studio, Python, and Repository. Below that, 'Summary' and 'Raw Data' are selected. On the left, a bar chart titled 'Tell us what you'd like to predict:' shows 'readmitted' with a value of 5.3k. On the right, a section titled 'Select Metric to Optimize' shows 'Recommended : LogLoss (Accuracy)' and a large orange button labeled 'Push to Start'. Below this are sections for 'Run Quick Autopilot' and 'Or Start in Semi Automatic or Manual Mode'. A table below lists features: diag_1_desc, diag_2_desc, diag_3_desc, race, gender, age, weight, and admission_type_id. A red box highlights the 'Show Advanced Options' dropdown.

The advanced modeling options appear.

This screenshot shows the same DataRobot interface as above, but with the 'Advanced Options' expanded. It includes a 'Select Partitioning Method' section with 'Stratified' selected, showing a stratified sampling diagram. Other options like 'Cross Validation' and 'Train/Validate/Holdout Split' are also shown. To the right, there are sections for 'Recommender', 'Additional Parameters', and 'Run Quick Autopilot'.

You can specify the following advanced options:

Partitioning Method

Random

Description: Observations are randomly assigned to the training, validation, and holdout sets.

Modeling Options:

- Cross-Validation -- specify the number of folds and holdout percentage.
- Training/Validation/Holdout Split -- specify the percentages for training, validation, and holdout.
- Random Seed -- specify a positive integer value for the random seed.

Partition Column

Description: A column in the data file is used to either specify a train/validate/holdout split or the folds/holdout to be used for cross-validation.

Modeling Options:

- Cross-Validation -- you must specify a value from the selected partition column that will specify the holdout set.
- Training/Validation/Holdout Split -- you must specify training, validation, and holdout set values from the selected partition column.

Group

Description: One or more columns can be selected, and each combination of these values is guaranteed to be in the same training or test set as other matching values.

Modeling Options:

- Cross-Validation -- specify the number of folds and holdout percentage.
- Training/Validation/Holdout Split -- specify the percentages for training, validation, and holdout.

Date

Description: Observations in the holdout set come after observations in the validation set chronologically, and observations in the validation set come after those in the training set. For this method you specify the Date/Time column in the dataset, which is used to create a Train/Validate/Holdout split with all rows in the test set occurring later than all rows in the training set, and all rows in holdout occurring later than the test set.

Modeling Options:

- Validation Percentage -- percentage of data allocated to the validation set.
- Holdout Percentage -- percentage of data allocated to the holdout set.

Stratified

Description: Observations are randomly assigned to training, validation, and holdout sets, preserving the same ratio of positive to negative cases as in the original data.

Modeling Options:

- Cross-Validation -- specify the number of folds and holdout percentage.
- Training/Validation/Holdout Split -- specify the percentages for training, validation, and holdout.

Recommender

Select the **Is this a Recommendation Problem?** check box if you would like to predict the rating or preference that a user would give to an item.

If you select this option, you must specify the two columns that contain the user and item identifiers.

A typical example for recommender systems is a product recommendation on an e-commerce site where you would like to predict how highly a user might rate a product. Predicted ratings are then used for content discovery on the site.

For recommendation problems, DataRobot runs models that use content-based and collaborative filtering techniques.

- Content-based models take into account any available user or item features and create a profile for each user (key words, price range, product types, brands, etc.).
- Collaborative filtering techniques are content-agnostic -- they look solely at the latent structure in the user-item ratings in order to make predictions.

DataRobot also runs hybrid models that combine both content-based and collaborative filtering techniques, as well as combinations of models (blended models).

Additional Parameters

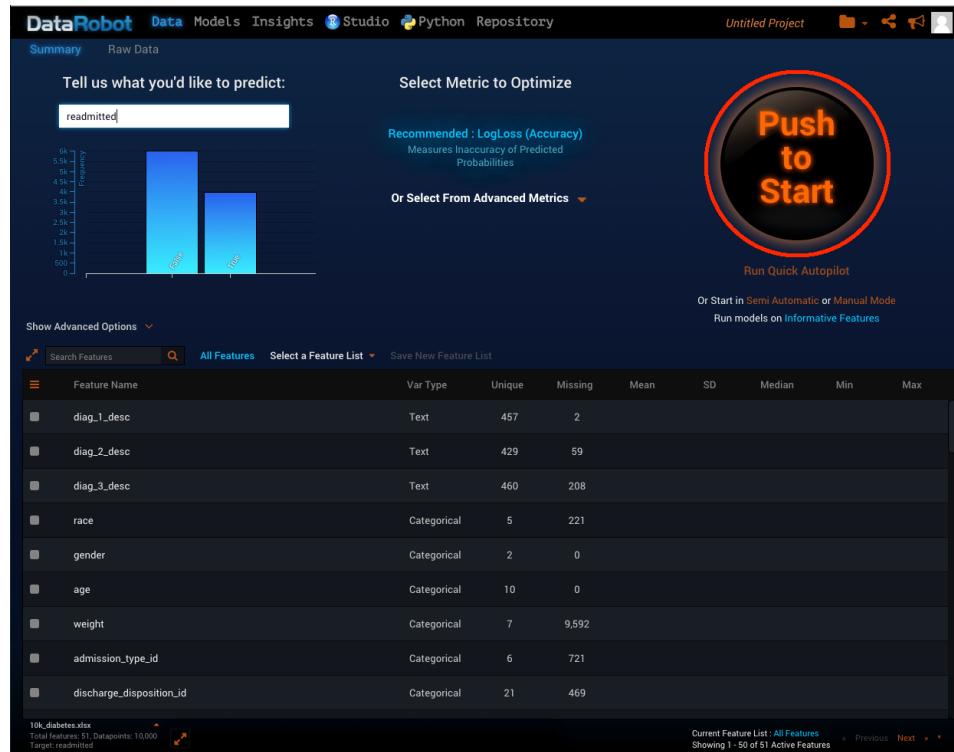
- Weight -- must be the name of a column in the dataset.
- Exposure -- must be the name of a column in the dataset.
- Upper Bound Running Time -- models that take longer to execute than this value (hrs) are excluded in subsequent autopilot runs.
- Response Cap -- The Response Cap limits the maximum value of the response (target) to a percentage of the original values. The value must be between 0.5 and 1.0 (50-100%).

Start Modeling

Start Modeling

After you have selected a target feature and an optimization metric, you can start modeling. The modeling process finds the best predictive models for the target feature.

To start modeling, click the **Push to Start** button.



The screenshot shows the DataRobot interface for a project titled "Untitled Project". On the left, there's a summary section with a bar chart showing the frequency of values for the target feature "readmitted". The chart has two bars: one for "No" (frequency ~4.5k) and one for "Yes" (frequency ~3.5k). Below the chart, there's a section to "Tell us what you'd like to predict:" with the target feature "readmitted" selected. To the right, there's a section to "Select Metric to Optimize" with "LogLoss (Accuracy)" recommended. A large button labeled "Push to Start" is prominently displayed, circled in red. Below it, there's a link to "Run Quick Autopilot". At the bottom, there's a table of features with their statistics, and a footer showing the file "10k_diabetes.xlsx" with 51 features and 10,000 datapoints, and the target feature "readmitted".

Feature Name	Var Type	Unique	Missing	Mean	SD	Median	Min	Max
diag_1_desc	Text	457	2					
diag_2_desc	Text	429	59					
diag_3_desc	Text	460	208					
race	Categorical	5	221					
gender	Categorical	2	0					
age	Categorical	10	0					
weight	Categorical	7	9.592					
admission_type_id	Categorical	6	721					
discharge_disposition_id	Categorical	21	469					

Autopilot Modes

The modeling process runs in Autopilot (fully automatic) mode by default, which means that the DataRobot platform will automatically select the best predictive models for the specified target feature.

For more control over which models to run, you can select **Run Quick Autopilot**, **Semi Automatic**, or **Manual Mode** under the **Push to Start** button.

The screenshot shows the DataRobot interface for a project titled "Untitled Project". At the top, there are tabs for Data, Models, Insights, Studio, Python, and Repository. Below the tabs, there's a summary section with a bar chart showing the frequency of the target variable "readmitted" (0 and 1). To the right of the chart is a large orange button labeled "Push to Start". Above the button, it says "Select Metric to Optimize" and "Recommended : LogLoss (Accuracy) Measures Inaccuracy of Predicted Probabilities". Below the button, there are options for "Run Quick Autopilot", "Semi Automatic", and "Manual Mode". A tooltip indicates "Run models on Informative Features". At the bottom, there's a table of features with columns for Feature Name, Var Type, Unique, Missing, Mean, SD, Median, Min, and Max. The table includes features like diag_1_desc, diag_2_desc, diag_3_desc, race, gender, age, weight, admission_type_id, and discharge_disposition_id. The footer shows the file path "10K.diabetes.xlsx", total features (51), datapoints (10,000), and the target variable "readmitted".

- **Run Quick Autopilot** -- The Quick Autopilot runs a small subset of models consisting of the best models based on the specified target feature and performance metric.
- **Semi Automatic** -- In semi-automatic mode, the autopilot will pause after the first cross-validation set before submitting additional models to the job queue. This allows you to inspect the models and choose to either add additional models or delete models.
- **Manual Mode** -- Starting the autopilot in manual mode gives you full control over which models to execute. For example, you can choose a specific model from the Task Repository rather than running the models selected by default.

The Worker Queue

After you start the modeling process, running and pending jobs appear in the Worker Queue at the right of the page. When a model is running, a graphical display of its CPU and RAM use will be displayed in the Worker Queue. The queue will execute a maximum fixed number of workers per project at any one time.

The screenshot shows the DataRobot interface with the following details:

- Top Bar:** DataRobot, Data, Models, Insights, Studio, Python Repository, Untitled Project.
- Left Panel:** Summary, Raw Data, Search Features, All Features, Select a Feature List, Save New Feature List.
- Feature List Table:**

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	Target	Boolean	2	0	0.40	0.49	0	
discharge_disposition_id	Categorical	21	469					
number_diagnoses	Numeric	9	0	7.03	2.02	7		
number_inpatient	Numeric	11	0	0.39	0.85	0		
medical_specialty	Categorical	52	4,100					
admission_source_id	Categorical	10	936					
diag_3	Categorical	460	208					
diag_2	Categorical	429	59					
num_medications	Numeric	68	0	15.56	8.39	14		
num_lab_procedures	Numeric	108	0	43.08	19.45	44		
diag_2_desc	Text	429	59					
weight	Categorical	7	9,592					
age	Categorical	10	0					
diag_1_desc	Text	457	2					
payer_code	Categorical	15	5,341					
diag_3_desc	Text	460	208					
time_in_hospital	Numeric	14	0	4.43	3.02	4		
- Bottom Left:** 10K_diabetes.xlsx, Total features: 51, Datapoints: 10,000, Target: readmitted, Metric: LogLoss.
- Bottom Right:** Current Feature List: All Features, Showing 1 - 50 of 51 Active Features.
- Right Panel:** Workers: 002, Processing (2) tasks, Queue (5) tasks.

Adjusting the Number of Workers

You can adjust the maximum number of simultaneous workers by clicking the orange arrows at the right of the green numbers in the Workers field at the top of the Worker Queue.

The screenshot shows the DataRobot interface with the following details:

- Top Bar:** DataRobot, Data, Models, Insights, Studio, Python Repository, Untitled Project.
- Left Panel:** Summary, Raw Data, Search Features, All Features, Select a Feature List, Save New Feature List.
- Feature List Table:**

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	Target	Boolean	2	0	0.40	0.49	0	
discharge_disposition_id	Categorical	21	469					
number_diagnoses	Numeric	9	0	7.03	2.02	7		
number_inpatient	Numeric	11	0	0.39	0.85	0		
medical_specialty	Categorical	52	4,100					
admission_source_id	Categorical	10	936					
diag_3	Categorical	460	208					
diag_2	Categorical	429	59					
num_medications	Numeric	68	0	15.56	8.39	14		
num_lab_procedures	Numeric	108	0	43.08	19.45	44		
diag_2_desc	Text	429	59					
weight	Categorical	7	9,592					
age	Categorical	10	0					
diag_1_desc	Text	457	2					
payer_code	Categorical	15	5,341					
diag_3_desc	Text	460	208					
time_in_hospital	Numeric	14	0	4.43	3.02	4		
- Bottom Left:** 10K_diabetes.xlsx, Total features: 51, Datapoints: 10,000, Target: readmitted, Metric: LogLoss.
- Bottom Right:** Current Feature List: All Features, Showing 1 - 50 of 51 Active Features.
- Right Panel:** Workers: 002, Processing (2) tasks, Queue (5) tasks.

Viewing Worker Queue Details

To view more details about the running jobs, click the orange arrow at the bottom of the Worker Queue. Click the arrow again to hide queue details.

The screenshot shows the DataRobot interface with the 'Worker Queue' details expanded. The queue contains 13 items, each with a progress bar and some descriptive text. At the bottom right of the queue area, there is an orange-outlined arrow pointing to the right, which serves as a button to collapse the queue details.

Pausing the Worker Queue

To pause the Worker Queue, click the Pause symbol at the top of the Worker Queue. After you pause the queue, the Pause symbol changes to a Play symbol (arrow). To resume running models, click the Play arrow.

The screenshot shows the DataRobot interface with the following details:

- Feature List:** A table showing 51 features. Key columns include Feature Name, Importance, Var Type, Unique, Missing, Mean, SD, Median, and Min.
- Workers:** A section titled "Workers: 002" showing two active workers: "ExtraTrees Classifier (Gini) (26)" and "Nystroem Kernel SVM Classifier...".
- Queue:** A section titled "Queue (5)" listing five waiting models: "Gradient Boosted Trees Clas...", "eXtreme Gradient Boosted Tr...", "Gradient Boosted Greedy Tre...", "Auto-tuned K-Nearest Neighb...", and "Elastic-Net Classifier (mix...)".
- Bottom Status:** Shows "10K_diabetes.xlsx", "Total features: 51, Datapoints: 10,000", "Target: readmitted, Metric: LogLoss", "Current Feature List: All Features", "Showing 1 - 50 of 51 Active Features", and navigation buttons for "Previous" and "Next".

Killing Workers

You can kill a worker by clicking the X next to the job name in the Worker Queue. If a worker fails for any reason, it will be listed under Errors at the bottom of the Worker Queue.

This screenshot is identical to the one above, except the "ExtraTrees Classifier (Gini) (26)" worker in the "Processing" section has been killed, as indicated by a red box around its X button.

After modeling has started, you can click **Models** in the top navigation menu to view the Leaderboard. The Leaderboard is a list of models ranked by the chosen performance metric, with the best models at the top

of the list. The rankings keep changing until all models have finished running. You cannot view full details about a model until it finishes running.

When all workers have finished running, the Worker Queue displays a message at the top of the queue and provide options to rerun the models a different feature list, or to unlock the holdout dataset and continue running workers.

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	Target	Boolean	2	0	0.40	0.49	0	
discharge_disposition_id		Categorical	21	469				
number_diagnoses		Numeric	9	0	7.03	2.02	7	
number_inpatient		Numeric	11	0	0.39	0.85	0	
medical_specialty		Categorical	52	4,100				
admission_source_id		Categorical	10	936				
diag_3		Categorical	460	208				
diag_2		Categorical	429	59				
num_medications		Numeric	68	0	15.56	8.39	14	
num_lab_procedures		Numeric	108	0	43.08	19.45	44	
diag_2_desc		Text	429	59				
weight		Categorical	7	9,592				
age		Categorical	10	0				
diag_1_desc		Text	457	2				
payer_code		Categorical	15	5,341				

10k_diabetes.xlsx
Total features: 51, Datapoints: 10,000
Target: readmitted, Metric: LogLoss

Current Feature List: All Features
Showing 1 - 50 of 51 Active Features

Workers: 002

Autopilot has finished

- Run Autopilot On A Different Feature List
- Unlock Holdout
- View Summary
- Datarobot Prime

The Model Leaderboard

Click **Models** in the top navigation menu to view the Leaderboard. The Leaderboard is a list of models ranked by the chosen performance metric, with the best models at the top of the list. You cannot view full details about a model until it finishes running.

Click a model name to view details about the model. Model details are discussed in the following sections.

Blueprint

Blueprints provide a graphical representation of the many steps involved in transforming input predictors and targets into a model. A blueprint represents the high-level end-to-end procedure for fitting the model, including any pre-processing steps, algorithms, and post-processing. Each box in a blueprint may represent multiple steps.

To view a graphical representation of a blueprint, click a model in the leaderboard. You can use the navigation toolbar to zoom in on the blueprint.

The screenshot shows the DataRobot interface with the 'Blueprint' visualization for the ENET Blender model. The visualization is a directed graph where nodes represent different data processing steps and models. The graph starts with a 'Data' node on the left, which branches into multiple 'Model Processing' nodes (Model 1, Model 14, Model 17, Model 21, Model 23, Model 26, Model 27, Model 28, Model 29, Model 30). These nodes then feed into a 'Stacked Ensemble' node, which finally leads to a 'Predictor' node on the right. The 'Stacked Ensemble' node contains sub-nodes labeled 'Stacked Ensemble (Model 14)' and 'Stacked Ensemble (Model 27)'. The entire graph is enclosed in a large rounded rectangle. Below the graph, there is a summary table for the ENET Blender model:

Model Name and Description	Informative Features	Sample Size	Validation	Cross Validation	Hold Out
ENET Blender (6+14+17+20+24+2...) (93)	Informative Features	64.0 %	0.6072	0.6048	

Below this table, there are other models listed in the leaderboard:

- Advanced AVG Blender (6+14+17+20+24+2...) (92)
- AVG Blender (20+26+27) (90)
- GLM Blender (20+26+27) (89)

At the bottom of the visualization area, there is a footer with the text: "10K_diabetes.xlsx Total features: 81, Datapoints: 10,000 Target: readmitted, Metric: LogLoss".

Input Types

Different columns in the dataset require different types of preparation and transformation. For example, some algorithms recommend subtracting the mean and dividing by the standard deviation of the input data -- but this would not make sense for text input data. The first step in the execution of a blueprint is to identify data types that belong together so they can be processed separately.

Transformers and Models

The other nodes in the blueprint are other types of data transformations or models. Click a blueprint node to display additional information. Many of the models use DataRobot proprietary approaches to data pre-processing.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. The 'Blueprint' section is highlighted, showing a detailed flowchart of the model's architecture. The flowchart starts with 'Data' and branches through various preprocessing steps (Model 12, Model 13, Model 14, Model 15, Model 16, Model 17, Model 18, Model 19, Model 20, Model 21, Model 22, Model 23, Model 24, Model 25, Model 26, Model 27) before reaching the 'ElasticNet Classifier (mixing alpha=0.5 / Binomial Deviance)' stage. This stage is described as 'Elastonet Classifier. Based on lightning CDDClassifier'. The Blueprint also includes links to 'documentation' and 'ROC Curve', 'Grid Search', 'Deploy Model', and 'Predict' options.

Lift Chart

The lift chart depicts how effective a model is at predicting the target. The chart is sorted by predicted values, so you can see how well the model performs for different ranges of values of the target variable.

To view a lift chart, click a model in the Leaderboard list, then click **Lift Chart**.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. The 'Lift Chart' button in the Blueprint section is highlighted. The lift chart for the ENET Blender model is displayed, showing the relationship between 'Value' (Y-axis, ranging from 0.11 to 0.70) and 'Sorted Prediction' (X-axis, ranging from 1 to 10). The chart compares 'Predicted' values (blue line with '+' markers) and 'Actual' values (orange line with circle markers). The chart shows a strong positive correlation, indicating that the model is effective at predicting the target variable. Other models listed in the Leaderboard include Advanced AVG Blender, AVG Blender, GLM Blender, and Nystroem Kernel SVM Classifier.

Display Options

- Data Source -- select validation, cross-validation, or holdout.
- Number of Bins -- use this drop-down to adjust the granularity of the displayed values.
- Enable Drill Down -- click this link to compute and download cross-validation predictions.
- Download Lift Table -- download a .csv file with the lift table data.

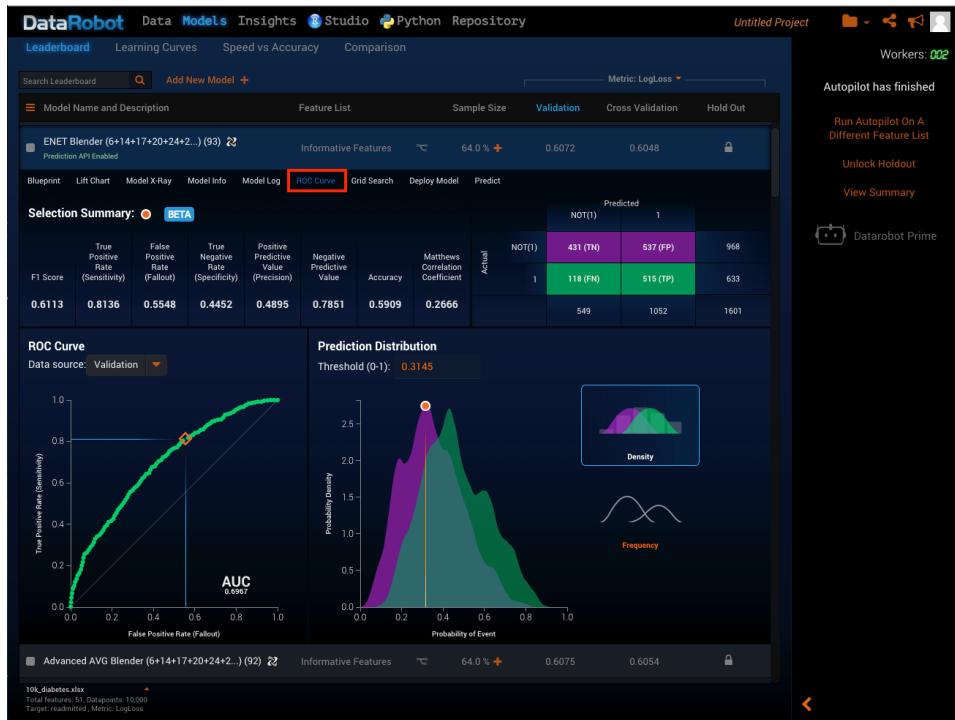
ROC Curve

You can use the ROC Curve page to assess model quality. To view the ROC Curve page, click a model in the Leaderboard list, then click **ROC Curve**.



Note: The ROC curve is displayed only for models created for a binary classification target, i.e., a target with two unique values.

The ROC Curve page contains a set of interactive graphical displays that includes the ROC Curve, a selection summary, the prediction distribution, and a confusion matrix.



ROC Curve

The ROC Curve plots the true positive rate against the false positive rate. It has two important characteristics: the area under the curve, and the shape of the curve.

Area Under the ROC Curve

The Area Under the Curve (AUC) is displayed at the lower right of the ROC Curve. The goal is for the AUC to be as large as possible.

- If the area = 0.5, it means that the predictions based on this model are no better than a random model.
- If the area = 1.0, it would mean that predictions based on the model are perfect.

A perfect model is not realistic -- if it does happen, there is likely something wrong with the model (it may contain variables that depend on the response and should be excluded). Therefore the goal is to achieve an area as large as possible between 0.5 and 1.0.

ROC Curve Shape

Another informative feature of the ROC Curve is its shape. It is better when the curve grows quickly for small X-values, and slowly for values of X closer to 1. An ideal ROC curve would hug the top left corner, indicating a high true positive rate and a low false positive rate.

Display Options

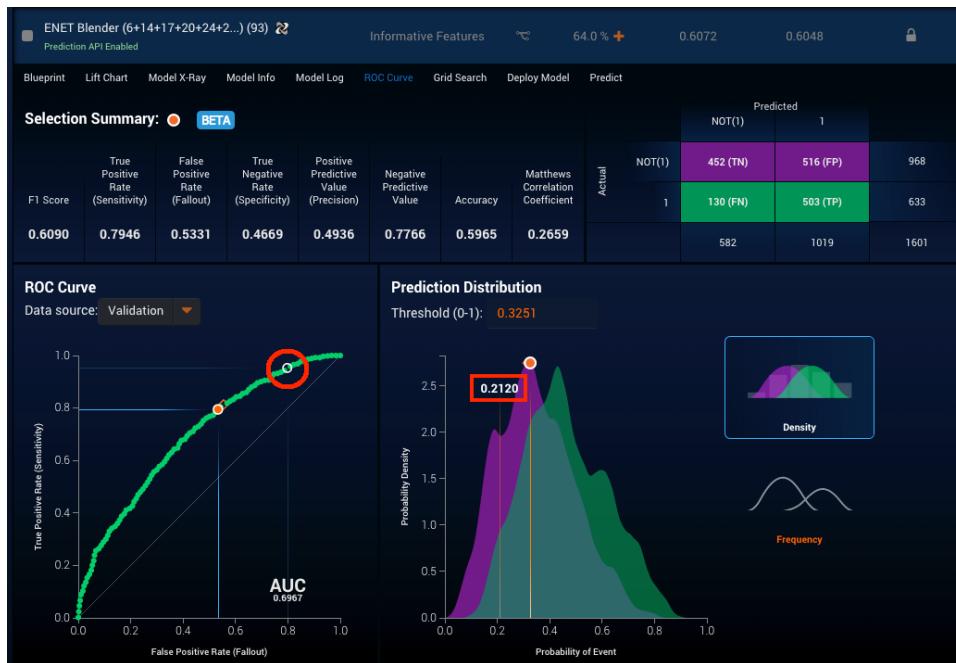
You can use the Data Source drop-down to display validation or cross-validation data.

Prediction Distribution

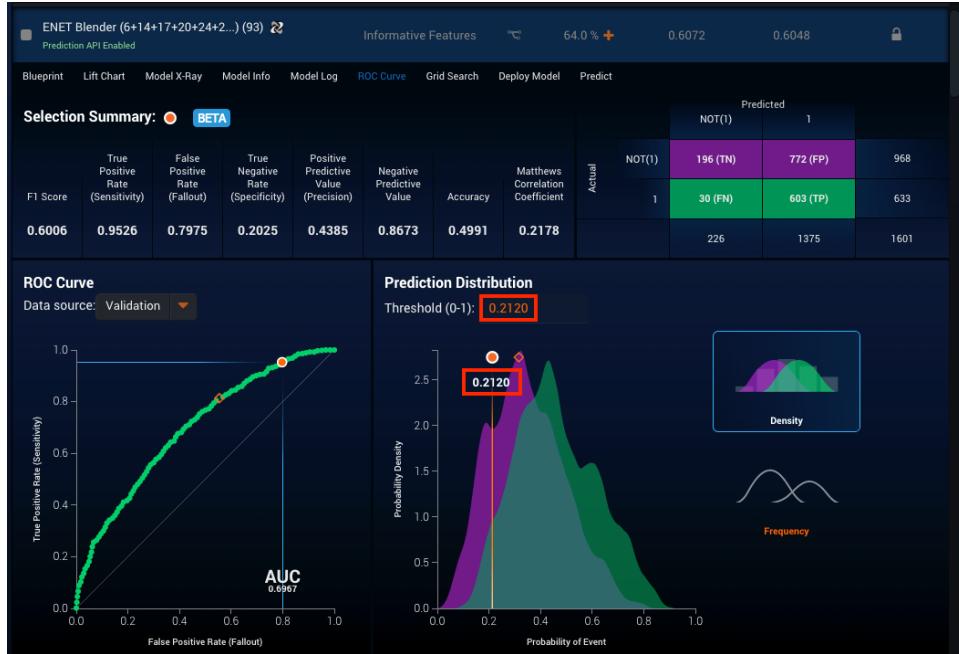
The Prediction Distribution chart expresses the performance of the model on the validation, cross-validation, or holdout dataset. You can use the Prediction Distribution chart to change the threshold value. When you change the threshold value, the ROC Curve, the Selection Summary table, and the Confusion Matrix are also updated to reflect the new value.

Use the following steps to set a new threshold value.

1. Pass the cursor over the Prediction Distribution chart. The threshold value will be displayed in white text as you move the cursor over the chart, and guidelines for the new values will also be dynamically displayed on both the Prediction Distribution chart and the ROC Curve.



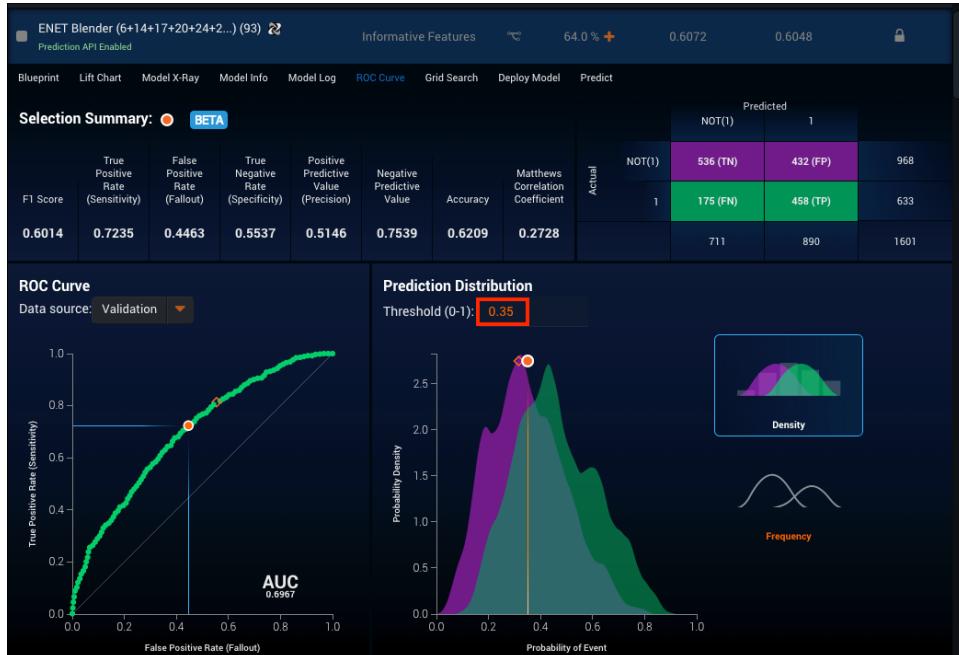
2. Click to select a new threshold value. The new value will appear in orange text in the Threshold field, and the orange dots and guidelines on the Prediction Distribution graph and ROC Curve will move to reflect the new threshold value. The information in the Selection Summary table and the Confusion Matrix will also be updated to reflect the new value.



3. You can also change the threshold setting by typing a new value in the Threshold field. Click the orange Threshold text, then type in a new value.

Note: "Invalid value" will appear next to the field if the number entered is not between 0 and 1. If a non-numeric character is entered, "NaN" (Not a Number) will appear in the Threshold field.

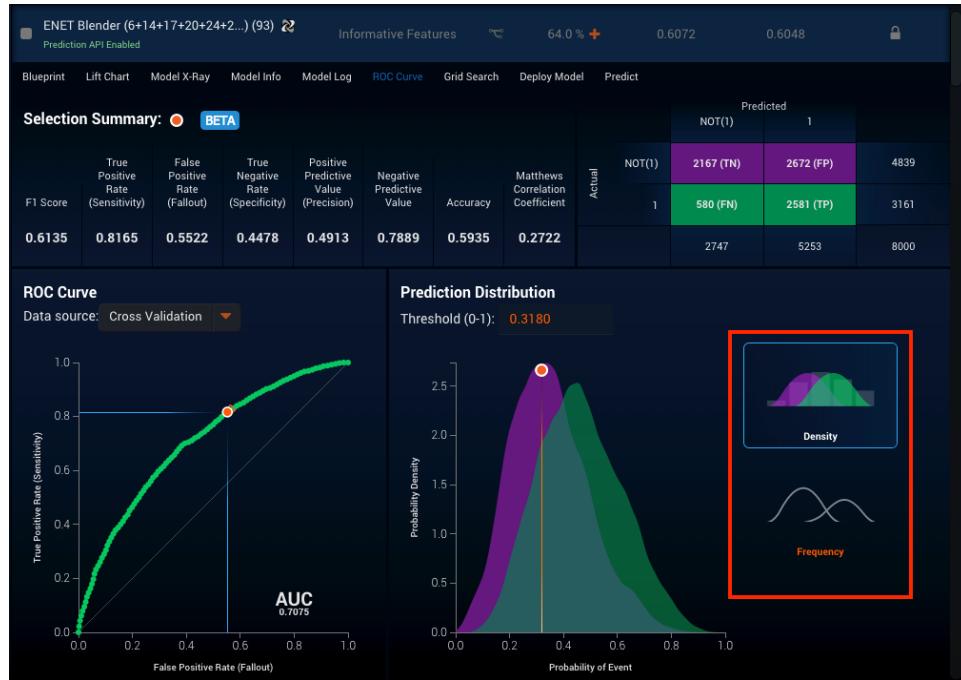
The orange dots and guidelines on the Prediction Distribution graph and ROC Curve will move to reflect the new threshold value, and the information in the Selection Summary table and the Confusion Matrix will also be updated to reflect the new value.



4. Click the **Density** and **Frequency** symbols at the right of the Prediction Distribution chart to display the Predictions distribution as a density or frequency curve:

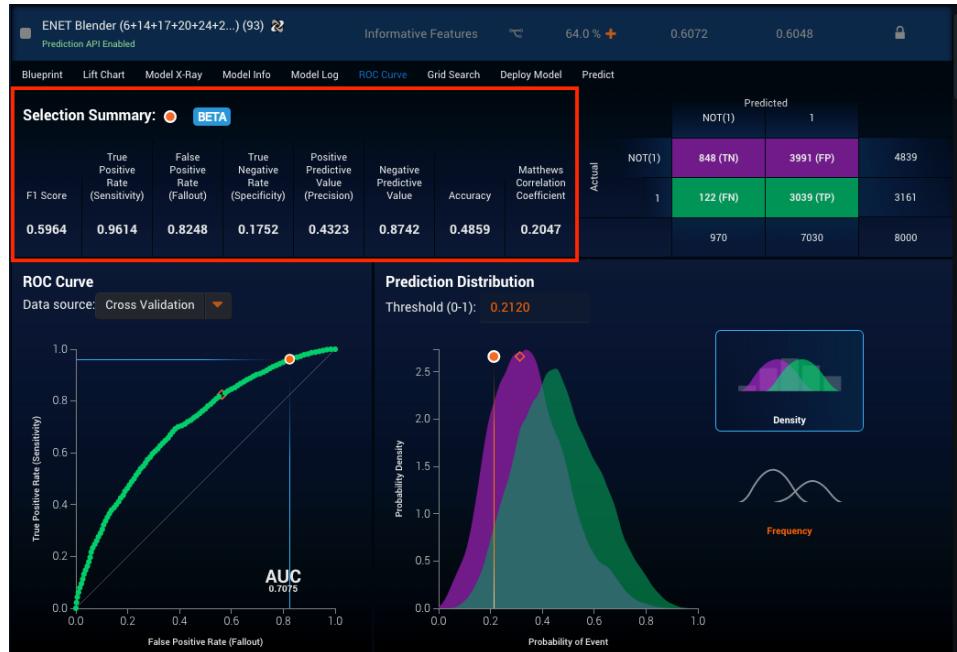
- Density curve -- There is an equal area underneath both the positive and negative curves.

- Frequency curve -- The area underneath each curve varies, and is determined by the number of observations in each class.



Selection Summary

The Selection Summary table contains a number of statistics describing model performance at the selected threshold, including the F1 Score (the normalized accuracy), sensitivity, fallout, specificity, precision, negative predictive value, accuracy, and the Matthews Correlation Coefficient.

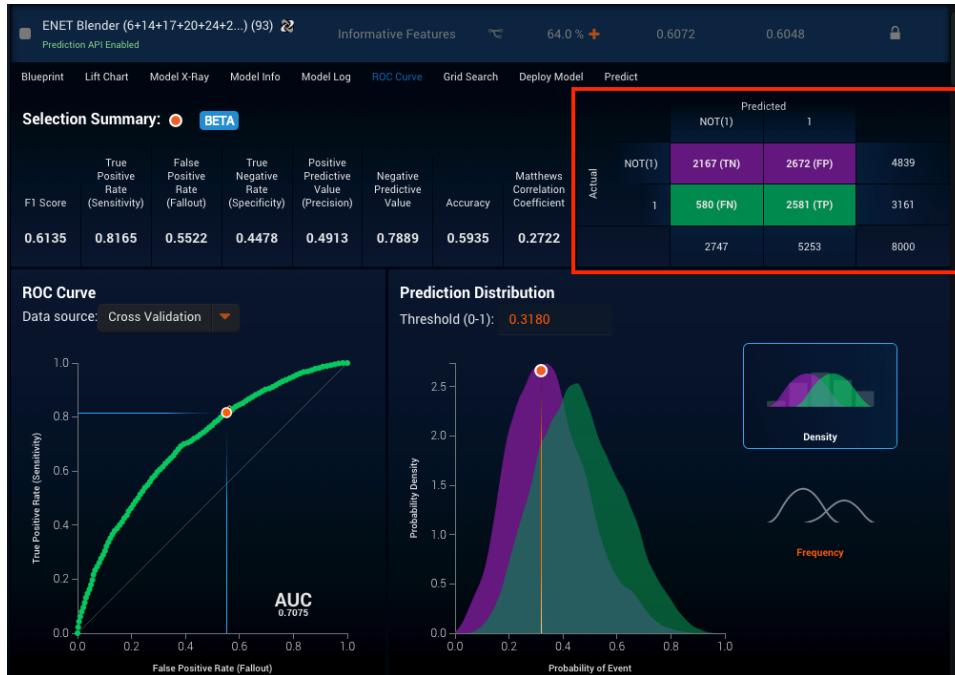


Confusion Matrix

The Confusion Matrix is a table with two rows and two columns that reports the number of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

The name "confusion matrix" refers to the fact that the matrix makes it easy to see if the model is confusing two classes (consistently mislabeling one class as another class).

The information in this table facilitates more detailed analysis than relying on accuracy alone. Accuracy is not always a reliable metric for the real performance of a classifier, because it will yield misleading results if the data set is unbalanced (that is, when the number of samples in different classes varies greatly).



Predict Against a Model

Use the following steps to generate predictions on a new dataset.

- Click a model in the Leaderboard list, then click **Predict**. Use one of the following methods to select a dataset to predict with the model.
 - Drag a .csv data file to the orange box on the DataRobot platform home page.
 - Click Browse File, then use the file browser to select a .csv data file.
 - Type a URL in the box labeled "http://path to your file", then click Import URL.

The screenshot shows the DataRobot interface with the 'Autopilot has finished' status. At the top, there's a large orange box containing the text "Drag Additional Datasets Here.". Below this, a file named "10k_diabetes.xlsx" is listed with the following details:

- Total Features: 91
- Data points: 10,000
- Target: readmitted Metric: LogLoss

- After the file is uploaded, click **Compute Prediction** next to the file.

The screenshot shows the DataRobot interface with the 'Compute & Download Model Predictions' section. A dataset named "Diabetes_500.xlsx" is selected, and its "Compute Prediction" button is highlighted with a red box.

- The **Compute Prediction** button changes to **In Queue** for the selected dataset, and the job status appears under Processing in the Worker queue.

The screenshot shows the DataRobot interface with the following details:

- Header:** DataRobot, Data, Models, Insights, Studio, Python, Repository, Untitled Project.
- Search Bar:** Search Leaderboard, Add New Model.
- Model List:**
 - ENET Blender (6+14+17+20+24+2...) (93) - Informative Features, Sample Size: 64.0%, Validation: 0.6072, Cross Validation: 0.6048, Hold Out: 0.6048.
 - Advanced AVG Blender (6+14+17+20+24+2...) (92) - Informative Features, Sample Size: 64.0%, Validation: 0.6075, Cross Validation: 0.6054, Hold Out: 0.6054.
 - AVG Blender (20+26+27) (90) - Informative Features, Sample Size: 64.0%, Validation: 0.6081, Cross Validation: 0.6064, Hold Out: 0.6064.
 - GLM Blender (20+26+27) (89) - Informative Features, Sample Size: 64.0%, Validation: 0.6088, Cross Validation: 0.6053, Hold Out: 0.6053.
 - Nystroem Kernel SVM Classifier (20) - Regularized Linear Model Preprocessing v4, Informative Features, Sample Size: 64.0%, Validation: 0.6094, Cross Validation: 0.6106, Hold Out: 0.6106.
 - Advanced GLM Blender (6+14+17+20+24+2...) (91) - Informative Features, Sample Size: 64.0%, Validation: 0.6100, Cross Validation: 0.6049, Hold Out: 0.6049.
 - Nystroem Kernel SVM Classifier (27) - Regularized Linear Model Preprocessing v5, Informative Features, Sample Size: 64.0%, Validation: 0.6110, Cross Validation: 0.6097, Hold Out: 0.6097.
- Compute & Download Model Predictions:**
 - Informative Features (Created 2015-06-23 18:48:35) - Compute Prediction button.
 - Diabetes_500.xlsx (Created 2015-06-24 16:35:27) - In Queue button.
- Processing Queue:**
 - ENET Blender... (64.0% sample, CV 45, 2%, 1.9 GB RAM)
- Footer:** 10k_diabetes.xlsx, Total features: 81, Datapoints: 10,000, Target: readmitted, Metric: LogLoss.

- When the prediction has finished running, click **Download Prediction** to view the results in a .csv file.

The screenshot shows the DataRobot interface with the following details:

- Header:** DataRobot, Data, Models, Insights, Studio, Python, Repository, Untitled Project.
- Search Bar:** Search Leaderboard, Add New Model.
- Model List:**
 - ENET Blender (6+14+17+20+24+2...) (93) - Informative Features, Sample Size: 64.0%, Validation: 0.6072, Cross Validation: 0.6048, Hold Out: 0.6048.
 - Advanced AVG Blender (6+14+17+20+24+2...) (92) - Informative Features, Sample Size: 64.0%, Validation: 0.6075, Cross Validation: 0.6054, Hold Out: 0.6054.
 - AVG Blender (20+26+27) (90) - Informative Features, Sample Size: 64.0%, Validation: 0.6081, Cross Validation: 0.6064, Hold Out: 0.6064.
 - GLM Blender (20+26+27) (89) - Informative Features, Sample Size: 64.0%, Validation: 0.6088, Cross Validation: 0.6053, Hold Out: 0.6053.
 - Nystroem Kernel SVM Classifier (20) - Regularized Linear Model Preprocessing v4, Informative Features, Sample Size: 64.0%, Validation: 0.6094, Cross Validation: 0.6106, Hold Out: 0.6106.
 - Advanced GLM Blender (6+14+17+20+24+2...) (91) - Informative Features, Sample Size: 64.0%, Validation: 0.6100, Cross Validation: 0.6049, Hold Out: 0.6049.
 - Nystroem Kernel SVM Classifier (27) - Regularized Linear Model Preprocessing v5, Informative Features, Sample Size: 64.0%, Validation: 0.6110, Cross Validation: 0.6097, Hold Out: 0.6097.
- Compute & Download Model Predictions:**
 - Informative Features (Created 2015-06-23 18:48:35) - Compute Prediction button.
 - Diabetes_500.xlsx (Created 2015-06-24 16:35:27) - Download Prediction button (highlighted).
- Processing Queue:**
 - Autopilot has finished.
 - ENET Blender... (64.0% sample, CV 45, 2%, 1.9 GB RAM)
- Footer:** 10k_diabetes.xlsx, Total features: 81, Datapoints: 10,000, Target: readmitted, Metric: LogLoss.

- To upload and run predictions on additional datasets, click **Upload Additional Datasets**.

DataRobot Data Models Insights Studio Python Repository Untitled Project

Leaderboard Learning Curves Speed vs Accuracy Comparison Metric: LogLoss

Search Leaderboard Add New Model +

Model Name and Description Feature List Sample Size Validation Cross Validation Hold Out

ENET Blender (6+14+17+20+24+2...) (93) 22 Informative Features 64.0 % 0.6072 0.6048

Blueprint Lift Chart Model X-Ray Model Info Model Log ROC Curve Grid Search Deploy Model Predict

Compute & Download Model Predictions: Upload Additional Datasets +

Dataset	Created	Action
Informative Features	2015-06-25 16:50:25	Compute Prediction
Diabetes_500.xlsx	2015-07-01 18:38:21	Download Prediction

Autopilot has finished

Run Autopilot On A Different Feature List

Unlock Holdout

View Summary

Datarobot Prime

Advanced AVG Blender (6+14+17+20+24+2...) (92) 22 Informative Features 64.0 % 0.6075 0.6054

Stochastic Gradient Descent Classifier (33) 22 Regularized Linear Model Preprocessing V2 Informative Features 80.0 % 0.6079 0.6119 *

AVG Blender (20+26+27) (90) 22 Informative Features 64.0 % 0.6081 0.6064

GLM Blender (20+26+27) (89) 22 Regularized Linear Model Preprocessing V4 Informative Features 64.0 % 0.6088 0.6053

Nystroem Kernel SVM Classifier (20) 22 Regularized Linear Model Preprocessing V4 Informative Features 64.0 % 0.6094 0.6106

Advanced GLM Blender (6+14+17+20+24+2...) (91) 22 Informative Features 64.0 % 0.6100 0.6049

Multinomial Kernel SVM Classifier (97) 22 10k_diabetes.xlsx Total features: 81, Datapoints: 10,000 Target: readmit, Metric: LogLoss

The screenshot shows the DataRobot web application interface. At the top, there's a navigation bar with links for Data, Models, Insights, Studio, Python, and Repository, and a project title 'Untitled Project'. Below the navigation is a search bar and a button to 'Add New Model'. The main area displays a 'Leaderboard' of models, each with its name, description, feature list, sample size, validation and cross-validation metrics, and a lock icon. One model, 'ENET Blender', is highlighted. Below the leaderboard is a section for 'Compute & Download Model Predictions' with a table showing datasets and their creation times, along with 'Compute Prediction' and 'Download Prediction' buttons. To the right, there's a sidebar with options like 'Autopilot has finished', 'Run Autopilot On A Different Feature List', 'Unlock Holdout', 'View Summary', and 'Datarobot Prime'. At the bottom, there's a detailed view of the 'Advanced AVG Blender' model, including its configuration, metrics, and a note about the 'Multinomial Kernel SVM Classifier'.

Managing Projects

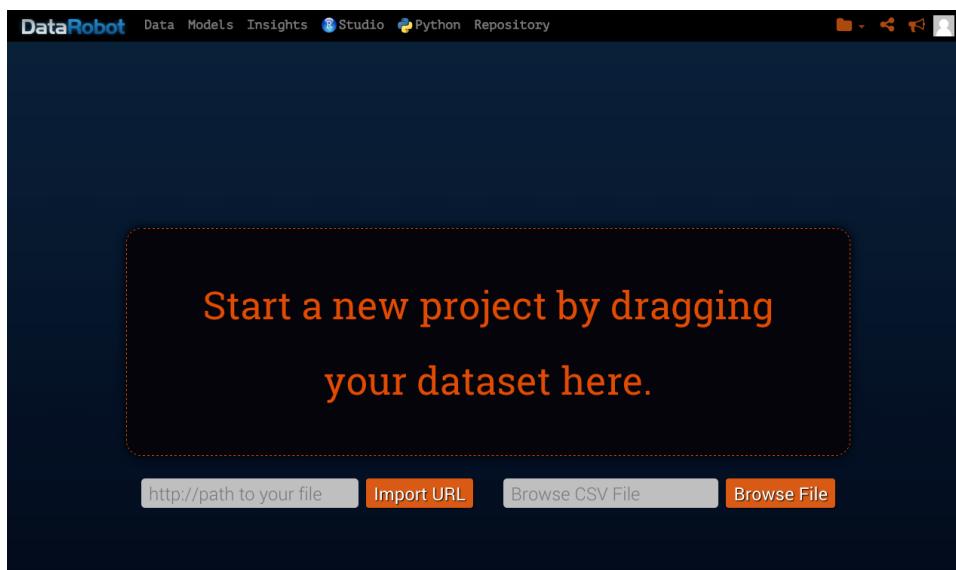
In order to build models in DataRobot, you must create a project. Each project has one dataset that is used as the source from which to train models.

Create a Project

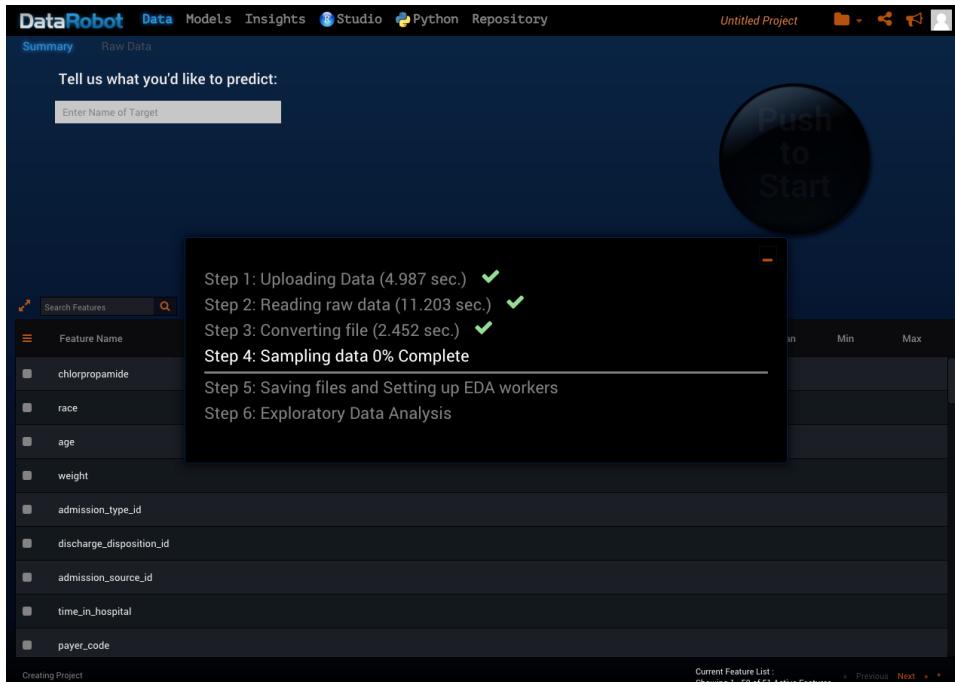
In order to build models in DataRobot, you must create a project with a dataset. After you create a project, you can select a target feature and start the modeling process. A DataRobot project contains all of the models you build for the project dataset.

Use the following steps to begin modeling data with the DataRobot API.

1. You can use any of the following methods to create a new DataRobot project:
 - Drag a .csv data file to the orange box on the DataRobot platform home page.
 - Click **Browse File**, then use the file browser to select a .csv data file.
 - Type a URL in the box labeled "http://path to your file", then click **Import URL**.

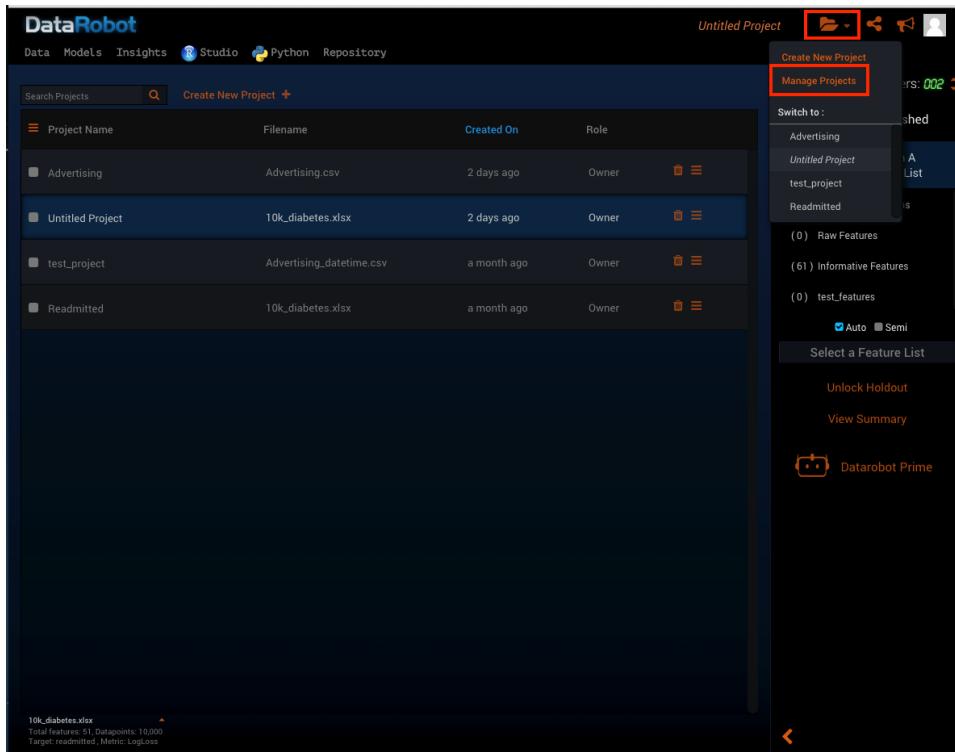


2. A progress indicator is displayed while the file is being processed.



Rename a Project

1. Click the folder icon in the menu at the top right of the page, then select **Manage Projects**.



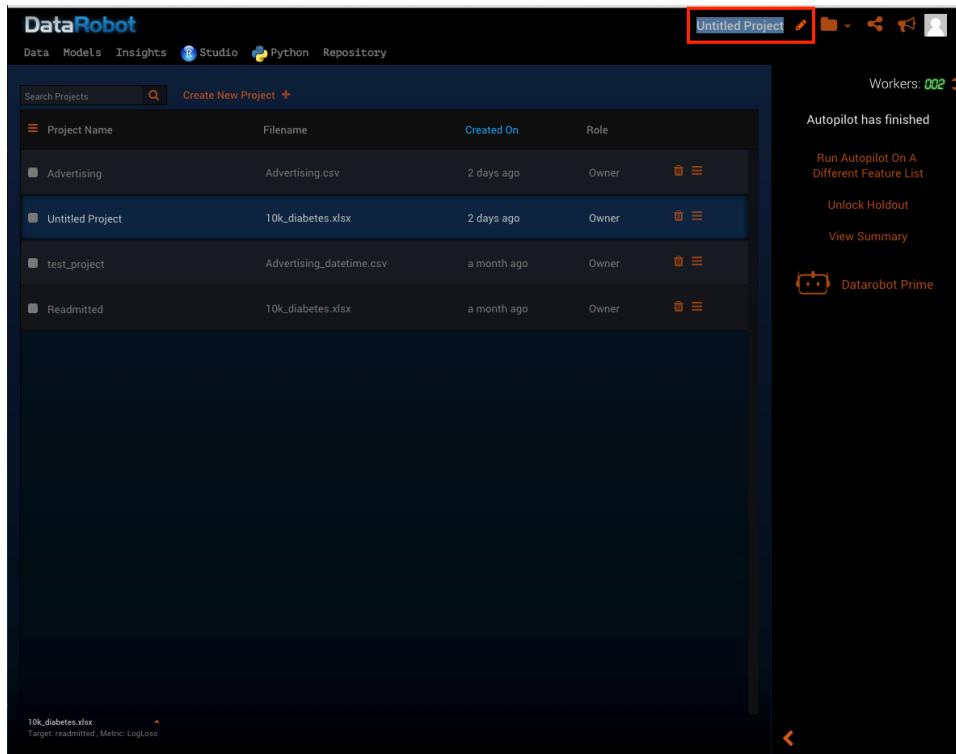
2. Click the project menu icon, then select **Rename Project**.

The screenshot shows the DataRobot interface with the 'Untitled Project' selected. A context menu is open, and the 'Rename Project' option is highlighted with a red box.

- Type in a new project name.

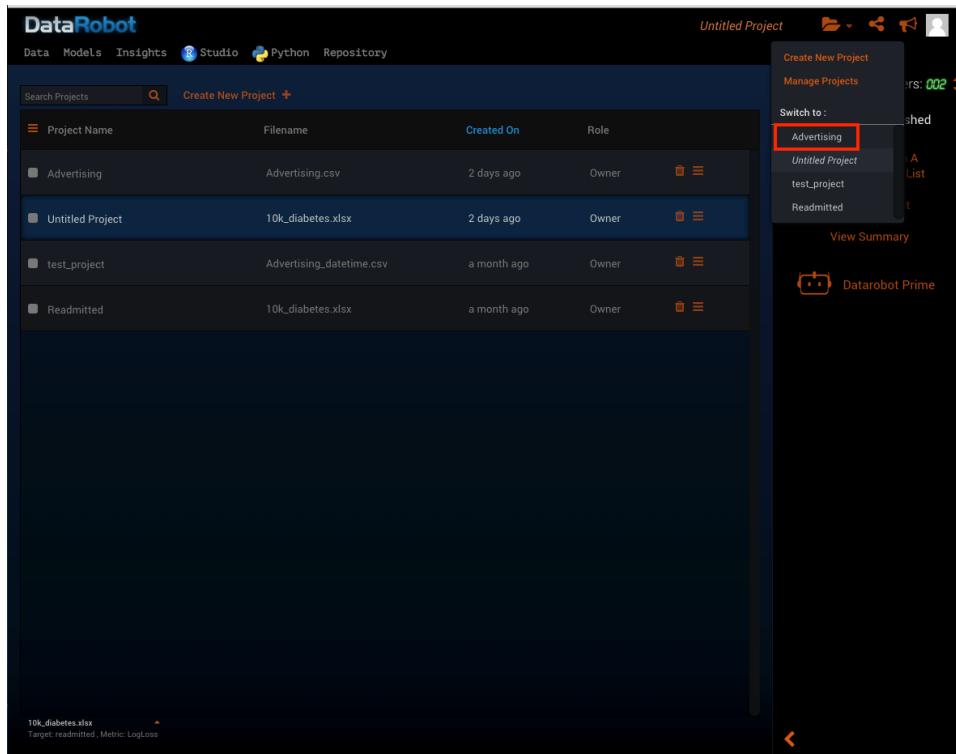
The screenshot shows the DataRobot interface with the 'Untitled Project' row selected. The project name field is highlighted with a red box and contains the text 'Untitled Project'. Below the input field, it says '84 characters remaining'.

- You can also rename a project by clicking the edit icon or the project name at the top of the page and entering a new project name.



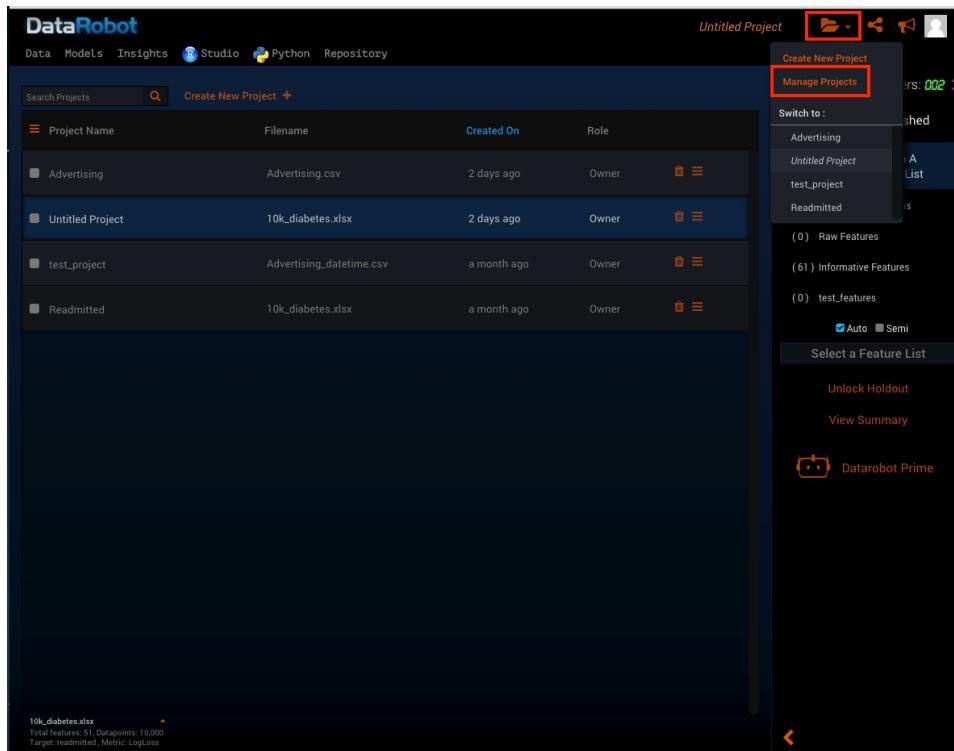
Switch to a Project

To switch to a different project, click the folder icon in the menu at the top right of the page, then select one of the projects listed under **Switch to:**.

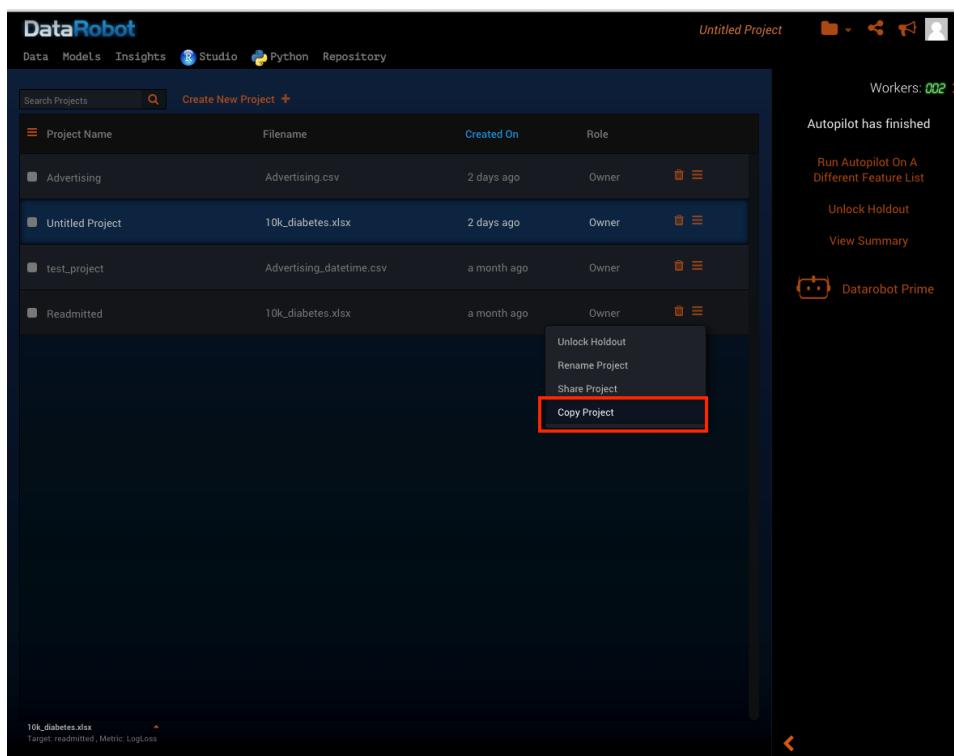


Copy a Project

1. Click the folder icon in the menu at the top right of the page, then select **Manage Projects**.



2. Click the project menu icon, then select **Copy Project**.



3. A new untitled project will be created using the dataset from the project, and the Data Summary page will be displayed.

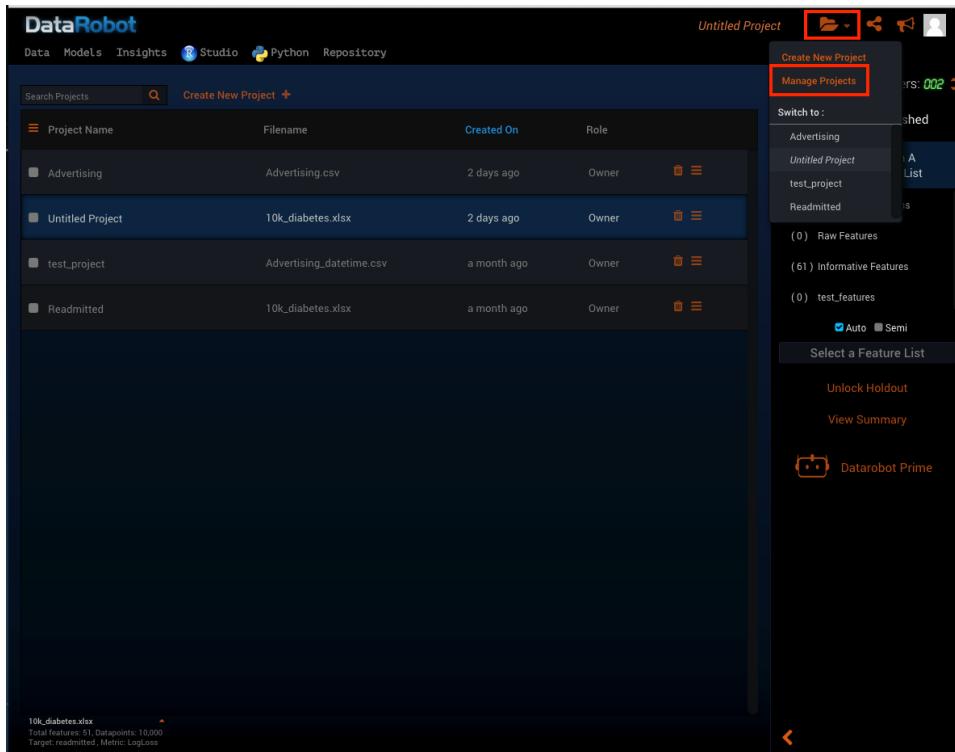
The screenshot shows the DataRobot interface with the 'Data' tab selected. At the top, there's a search bar labeled 'Enter Name of Target'. To the right is a large, dark circular button with the text 'Push to Start' in white. Below the search bar is a table titled 'Search Features' with columns: Feature Name, Var Type, Unique, Missing, Mean, SD, Median, Min, and Max. The table lists several features from a dataset named '10k_diabetes.xlsx':

Feature Name	Var Type	Unique	Missing	Mean	SD	Median	Min	Max
diag_1_desc	Text	457	2					
diag_2_desc	Text	429	59					
diag_3_desc	Text	460	208					
race	Categorical	5	221					
gender	Categorical	2	0					
age	Categorical	10	0					
weight	Categorical	7	9,592					
admission_type_id	Categorical	6	721					

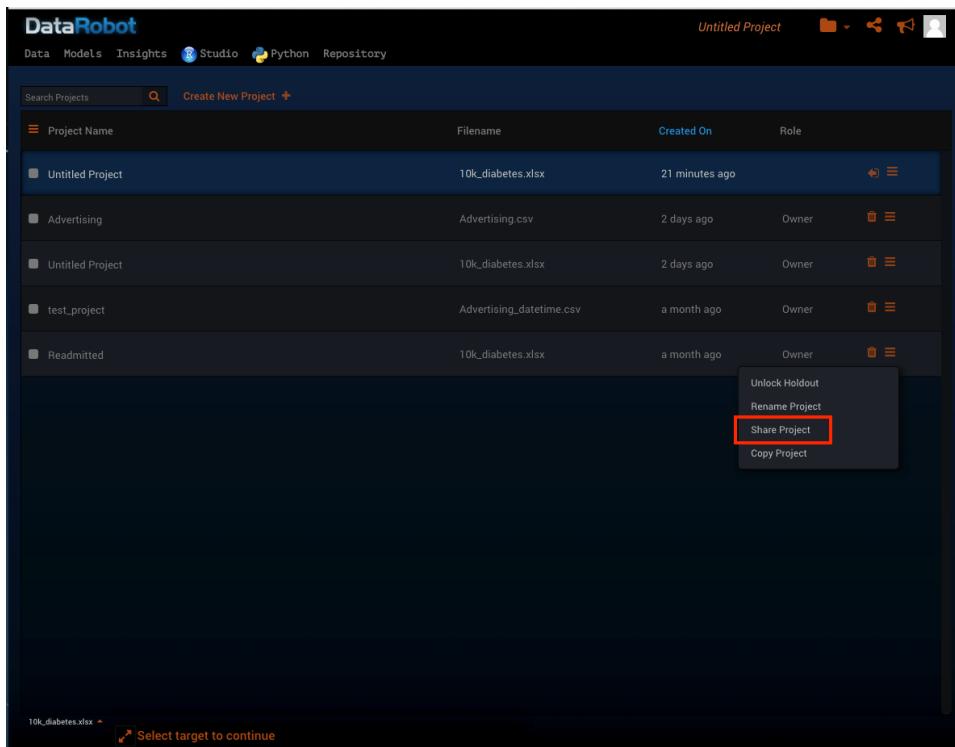
At the bottom left, there's a note: 'Select target to continue'. On the right, it says 'Current Feature List: All Features' and 'Showing 1 - 50 of 51 Active Features'.

Share a Project

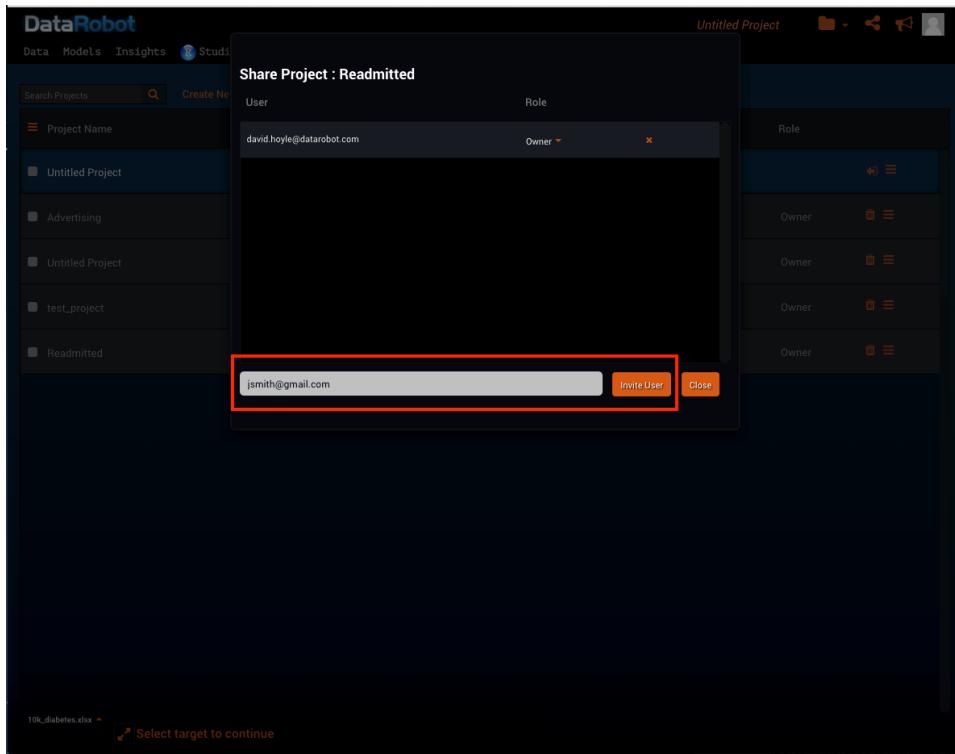
1. Click the folder icon in the menu at the top right of the page, then select **Manage Projects**.



2. Click the project menu icon, then select **Share Project**.



3. On the Share Project pop-up, type in the email address of the person you would like to share the project with, then click **Invite User**.



4. After you add the new user, they will receive an email invitation to join the project. You can use the drop-down in the Role column to change their role.

The available roles are:

- Observer -- can view the project, but cannot edit the project.
- Data Scientist --
- Admin --
- Owner --

5. You can also share a project by clicking the Share icon in the top menu.

The screenshot shows the DataRobot web application. At the top, there's a navigation bar with links for Data, Models, Insights, R Studio, Python, and Repository. Below the navigation is a search bar labeled "Search Projects" and a "Create New Project" button. The main area displays a list of projects:

Project Name	Filename	Created On	Role
Untitled Project	10k_diabetes.xlsx	17 hours ago	Owner
Advertising	Advertising.csv	3 days ago	Owner
Untitled Project	10k_diabetes.xlsx	3 days ago	Owner
test_project	Advertising_datetime.csv	a month ago	Owner
Readmitted	10k_diabetes.xlsx	a month ago	Owner

A context menu is open on the right side of the screen, with the "Unlock Holdout" option highlighted. Other visible options in the menu include "Run Autopilot On A Different Feature List", "View Summary", and "Datarobot Prime".

Unlock Holdout

1. Click the folder icon in the menu at the top right of the page, then select **Manage Projects**.

This screenshot is similar to the one above, showing the DataRobot interface with the project list. The context menu is open again, and the "Manage Projects" option is now highlighted with a red box.

2. Click the project menu icon, then select **Unlock Holdout**.

The screenshot shows the DataRobot interface with a list of projects. A context menu is open over the third project in the list, which is named 'Untitled Project'. The menu options are: Unlock Holdout (highlighted with a red box), Rename Project, Share Project, and Copy Project.

3. On the Confirm pop-up, click **Unlock**.

Note: Once holdout has been unlocked for a project, it cannot be re-locked.

The screenshot shows the DataRobot interface with a 'Confirm' dialog box overlaid. The dialog asks, "Are you sure you want to unlock the holdout for this project?" It contains two buttons: "Unlock" (highlighted with a red box) and "Cancel".

4. Holdout is unlocked and the label on the project menu changes to "Holdout is Unlocked".

Server has recently updated. Click to reload the page.

DataRobot Data Models Insights Studio Python Repository Untitled Project

Search Projects Create New Project +

Project Name	Filename	Created On	Role
Untitled Project	10k_diabetes.xlsx	17 hours ago	Owner
Advertising	Advertising.csv	3 days ago	Owner
Untitled Project	10k_diabetes.xlsx	3 days ago	Owner
test_project	Advertising_datetime.csv	a month ago	
Readmitted	10k_diabetes.xlsx	a month ago	

Workers: 0/2

Autopilot has finished

Run Autopilot On A Different Feature List

Holdout is Unlocked

View Summary

Datarobot Prime

Holdout is Unlocked

Rename Project

Share Project

Copy Project

10k_diabetes.xlsx
Total features: 81, Datapoints: 10,000
Target readmitted, Metric: LogLoss

The screenshot shows the DataRobot web interface. At the top, there's a navigation bar with links for Data, Models, Insights, Studio, Python, and Repository, along with a 'Untitled Project' section. Below the navigation is a search bar and a 'Create New Project' button. The main area displays a table of projects with columns for Project Name, Filename, Created On, and Role. One row in the table is highlighted with a red box around the 'Role' column, which shows 'Owner'. A context menu is open over this row, listing options: 'Holdout is Unlocked', 'Rename Project', 'Share Project', and 'Copy Project'. To the right of the table, there's a sidebar with status messages like 'Autopilot has finished' and 'Holdout is Unlocked', along with links to 'View Summary' and 'Datarobot Prime'. At the bottom left, there's a preview for a file named '10k_diabetes.xlsx' with details about its features and target metric.

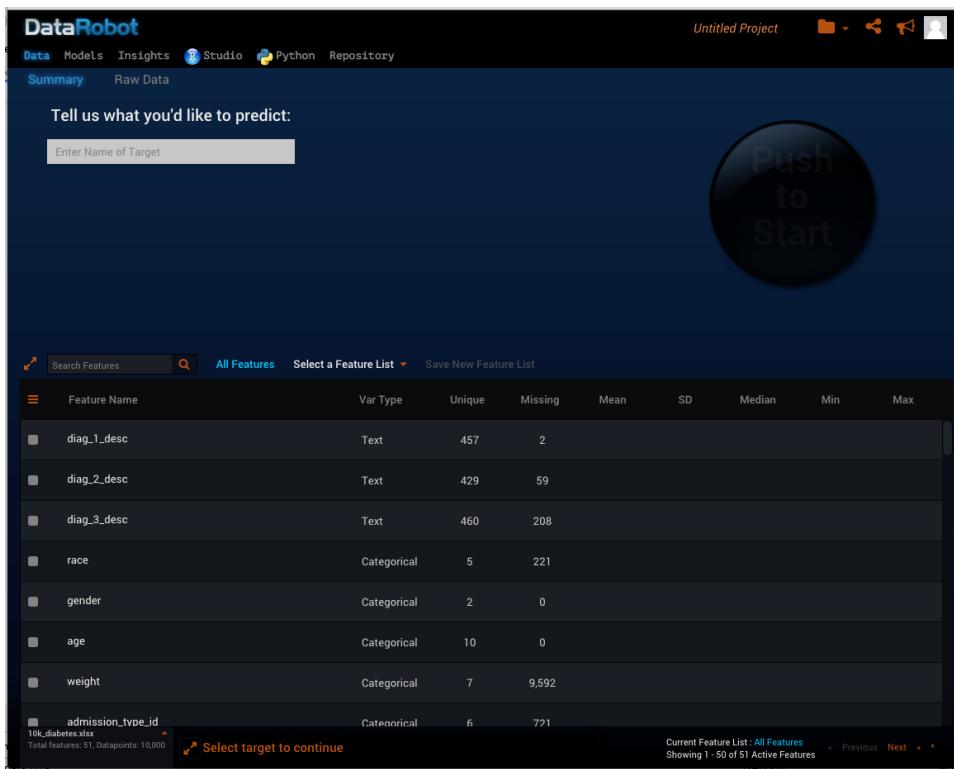
Working with Datasets

Each DataRobot project has one dataset that is used as the source from which to train models. You can use the Data page to view dataset features, create feature lists containing subsets of the dataset, and create derived features.

Viewing Data

You can use the Data page to view dataset features and create feature lists, and then select a target feature and begin modeling.

The Summary page appears after you create a DataRobot project.



The screenshot shows the DataRobot interface with the 'Summary' tab selected. At the top, there's a search bar labeled 'Enter Name of Target'. Below it is a large button labeled 'Push to Start'. The main area displays a table of features:

Feature Name	Var Type	Unique	Missing	Mean	SD	Median	Min	Max
diag_1_desc	Text	457	2					
diag_2_desc	Text	429	59					
diag_3_desc	Text	460	208					
race	Categorical	5	221					
gender	Categorical	2	0					
age	Categorical	10	0					
weight	Categorical	7	9,592					
admission_type_id	Categorical	6	721					

At the bottom, it says '10K_diabetes.xlsx Total features: 51, Datapoints: 10,000' and 'Select target to continue'. A note at the bottom right says 'Current Feature List: All Features Showing 1 - 50 of 51 Active Features'.



Note:

After you select a target feature and begin modeling, DataRobot analyzes the data and presents this information on the Data Summary page. Until then, the information displayed on the Summary page is somewhat limited. The following information assumes that you have selected a target feature and started the modeling process.

After you select a target feature and begin modeling, DataRobot analyzes the data and sorts the features on the Summary page in order of their importance in predicting the target variable.

The green bar next to each feature indicates its relative importance. Importance is calculated using an algorithm that measures the information content of the variable, and this calculation is done independently for each feature in the dataset. Importance is scaled such that 1 is equivalent to a perfect model, and 0 is equivalent to a null model, so it can be interpreted in the same manner as an R-squared score.

The following image shows the Data Summary page with features rated by importance. The page also lists unique and missing values, mean, median, standard deviation, and minimum and maximum values.

10k_diabetes.xlsx
Total features: 51, Datapoints: 10,000
Target: readmitted, Metric: LogLoss

Current Feature List: Informative Features
Showing 1 - 40 of 40 Active Features

Click a feature to display a graph of the mean target response plotted against the frequency distribution of the variable. To display a histogram of frequent values, click **Frequent Values**.

10k_diabetes.xlsx
Total features: 51, Datapoints: 10,000
Target: readmitted, Metric: LogLoss

Current Feature List: Informative Features
Showing 1 - 40 of 40 Active Features

Click **Raw Data** to view the raw data table.

race	gender	age	weight	admission_type_id	discharge_dispositi...	admission_source...	time_in_hospital	payer_code	medical_specialty	num_lab_proc
Caucasian	Male	[40-50)		Elective	Discharged to ho...	Physician Referral	1		Cardiology	26
Caucasian	Female	[80-90)		Urgent	Discharged to ho...	Emergency Room	2	MC	Emergency/Trauma	6
Caucasian	Male	[70-80)		Elective	Discharged to ho...	Physician Referral	1		Surgery-Neuro	35
AfricanAmerican	Male	[50-60)		Emergency	Discharged to ho...	Emergency Room	1		InternalMedicine	40
AfricanAmerican	Female	[50-60)		Emergency	Discharged to ho...	Emergency Room	5		Psychiatry	31
Caucasian	Female	[70-80)		Elective	Discharged/transf...	Physician Referral	6	SP	InternalMedicine	1
Caucasian	Female	[60-70)		Elective	Expired	Physician Referral	6	MC	InternalMedicine	46
Caucasian	Female	[50-60)		Emergency	Discharged to ho...	Emergency Room	2			49
Caucasian	Female	[80-90)		Emergency	Discharged to ho...	Emergency Room	3		InternalMedicine	46
AfricanAmerican	Female	[50-60)		Emergency	Discharged to ho...	Emergency Room	5	BC	InternalMedicine	59
Hispanic	Female	[30-40)			Discharged to ho...	Physician Referral	3		ObstetricsandGyn...	25
Caucasian	Female	[70-80)		Elective	Discharged/transf...	Physician Referral	4	MC	Radiologist	35
Caucasian	Male	[80-90)		Emergency	Discharged to ho...	Emergency Room	6			21
Caucasian	Male	[80-90)		Not Mapped	Discharged to ho...	Physician Referral	1		Surgery-Vascular	22
Caucasian	Female	[60-70)	[75-100)	Elective	Discharged to ho...	Physician Referral	4	MC	InternalMedicine	43
Caucasian	Female	[90-100)		Urgent	Discharged to ho...	Physician Referral	2		Cardiology	12
AfricanAmerican	Male	[30-40)		Emergency	Discharged/transf...	Emergency Room	10			66
Caucasian	Male	[60-70)		Emergency	Discharged to ho...	Emergency Room	6		InternalMedicine	59
AfricanAmerican	Female	[90-100)		Emergency	Discharged/transf...	Emergency Room	5			37
AfricanAmerican	Female	[90-100)		Emergency	Discharged/transf...	Emergency Room	14		Cardiology	47
AfricanAmerican	Female	[60-70)		Emergency	Discharged to ho...	Emergency Room	2		InternalMedicine	44
Caucasian	Female	[60-70)		Elective	Discharged to ho...	Transfer from a h...	4			45
AfricanAmerican	Female	[70-80)		Emergency	Discharged to ho...	Emergency Room	6		Family/GeneralPr...	74
Caucasian	Female	[60-70)		Urgent	Discharged to ho...	Emergency Room	1	BC	Emergency/Trauma	46
Asian	Male	[10-20)		Emergency	Discharged to ho...	Emergency Room	9	LOS		20

Page 1 of 27 ▾ 0 - 100 of 2576 Rows
10k_diabetes.xlsx Total features: 51, Datapoints: 10,000 Select target to continue

Dataset Guidelines

The following data file formats are supported by DataRobot.

- .csv
- .xls
- .xlsx

Those formats can also be compressed or archived with the following formats:

- .bz2
- .zip
- .gz
- .tar
- .t gz
- .tar.gz

The file must be a comma-separated file with a header that matches the number of data columns.

Each row must have the same number of fields, some of which may be blank. Ensure that the data is in ASCII, UTF-8, or WINDOWS-1252 CSV file format, and that the data file does not have any extraneous characters or escape sequences from URLs.



Note: At this time it is safer to upload files smaller than 524 MB, even if they are compressed.

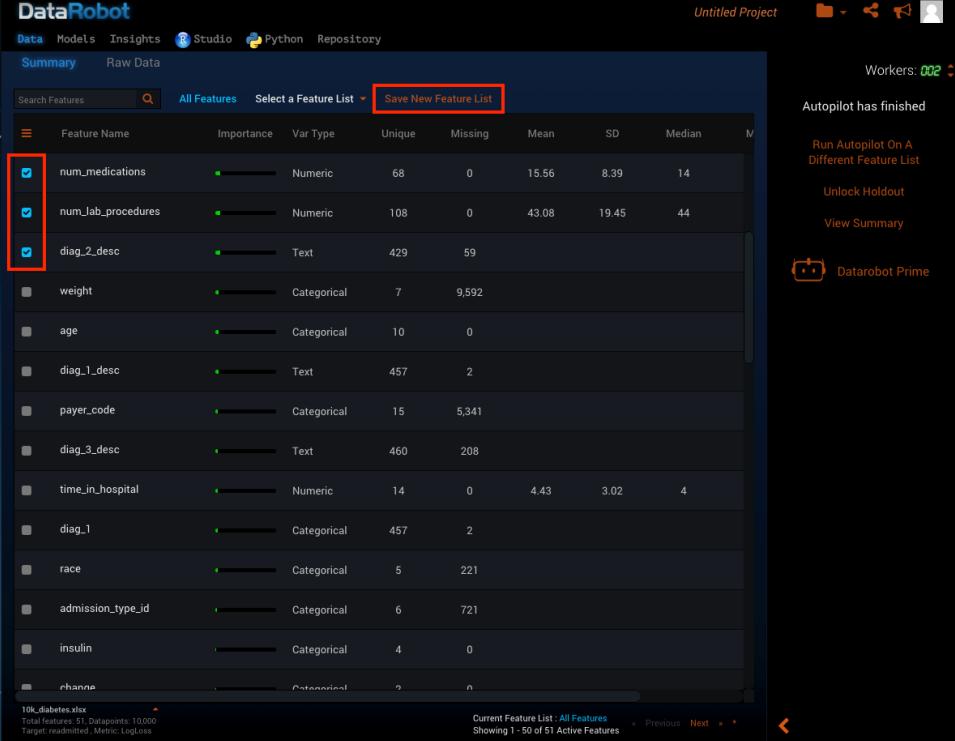
If you have a larger dataset, you can submit a subset of the data and contact DataRobot for information on working with large datasets.

Working with Feature Lists

By default DataRobot uses all of the features (also referred to as Raw Features) in the dataset for the prediction task. You can create a feature list that contains a subset of the features, and train the model on this feature set to see if it yields a better model.

Create a Featurelist

To create a new feature list, use the check boxes on the Data Summary page to select a set of features, then click **Save New Feature List**.



The screenshot shows the DataRobot interface with the 'Data' tab selected. In the top navigation bar, there are tabs for Data, Models, Insights, R Studio, Python, and Repository. Below the tabs, there are buttons for Summary and Raw Data. A search bar labeled 'Search Features' is followed by buttons for All Features, Select a Feature List (with a dropdown arrow), and Save New Feature List (which is highlighted with a red box). The main area displays a table of features with columns: Feature Name, Importance, Var Type, Unique, Missing, Mean, SD, Median, and M. Several checkboxes are checked for the first three features: num_medications, num_lab_procedures, and diag_2_desc. To the right of the table, a sidebar shows 'Workers: 002' and a message 'Autopilot has finished'. It also includes links for 'Run Autopilot On A Different Feature List', 'Unlock Holdout', 'View Summary', and a 'Datarobot Prime' section. At the bottom of the table, there is a note about the file '10k_diabetes.xlsx' and its metrics: Total Features 51, Datapoints 10000, Target: readmitted, Metric: LogLoss. The footer of the table indicates the current feature list is 'All Features' and shows 'Showing 1 - 50 of 51 Active Features'.

The new feature list appears on the Data Summary page, and its name replaces the Select a Feature List label.

The screenshot shows the DataRobot interface with the 'Summary' tab selected. In the top navigation bar, there is a dropdown menu labeled 'test_features' with a red box around it. To the right of the main dashboard, a sidebar titled 'Autopilot has finished' is open. This sidebar includes a 'Select a Feature List' dropdown set to 'Auto' (with 'Semi' as an option), a 'Unlock Holdout' button, and a 'View Summary' button. At the bottom of the sidebar, there is a 'Datarobot Prime' logo.

You can use the **Select a Feature List** menu to change the set of features used to train models. DataRobot provides the following feature lists:

- Raw Features -- All of the features (the default selection).
- Univariate Selections -- Features that are determined by the ACE function to have a relationship with the target.
- Informative Features -- Features that pass a reasonableness check that determines whether or not they contain useful information. For example, a column of ones or a duplicate of an existing column would be excluded from the informative features list.

including **Univariate Selections** and **Informative Features**. Informative Features are those features that have been sorted at the top of the Features list.

The screenshot shows the DataRobot interface with the 'Summary' tab selected. A modal dialog titled 'Select a Feature List' is open, listing three categories: 'Univariate Selections', 'Raw Features', and 'Informative Features'. The 'Informative Features' section contains one item: 'test_features'. The main table below shows various features like 'readmitted', 'discharge_disposition_id', and 'number_inpatient' with their respective statistics.

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	0.40	Numeric	9	0	0.49	0	0	0
discharge_disposition_id	-0.69	Categorical	52	4,100				
number_inpatient	0.0	Numeric	11	0	7.03	2.02	7	1
number_inpatient	0.39	Numeric	11	0	0.39	0.85	0	0
medical_specialty	-0.0	Categorical	52	4,100				
admission_source_id	-0.0	Categorical	10	936				
diag_3	-0.0	Categorical	460	208				
diag_2	-0.0	Categorical	429	59				
num_medications	-0.0	Numeric	68	0	15.56	8.39	14	1
num_lab_procedures	-0.0	Numeric	108	0	43.08	19.45	44	1
diag_2_desc	-0.0	Text	429	59				
weight	-0.0	Categorical	7	9,592				
age	-0.0	Categorical	10	0				
diag_1_desc	-0.0	Text	457	2				

Creating Derived Features

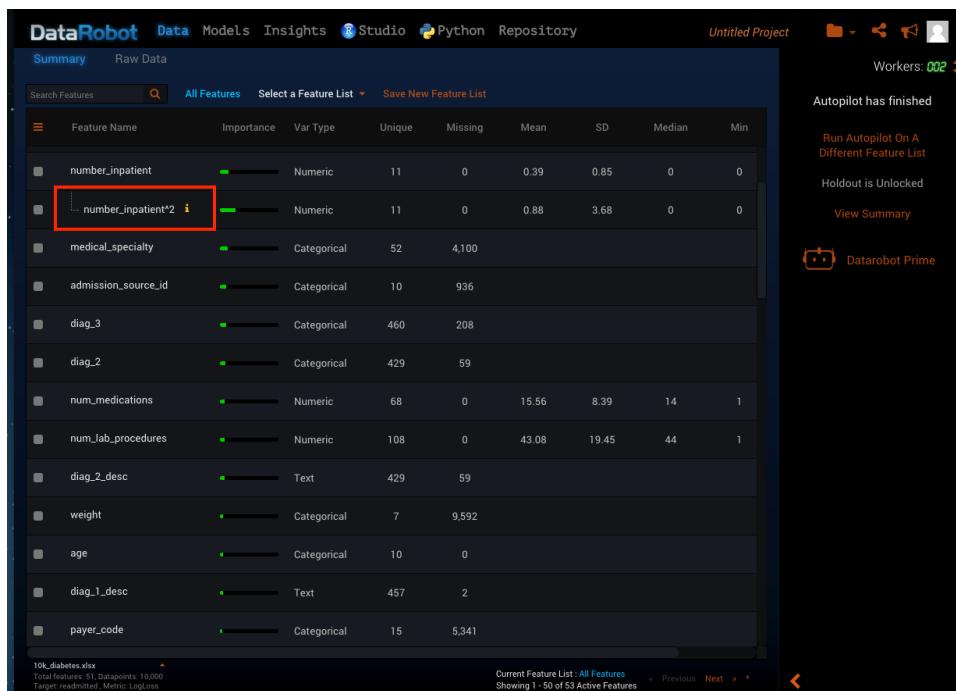
DataRobot provides access to several different transformations that you can apply to your data. Currently, that includes squaring and taking the natural logarithm of numeric data, when appropriate.

- To create a derived feature, click the orange arrow at the left of the feature name, then select a transformation.

The screenshot shows the DataRobot interface with the 'Summary' tab selected. A derived feature 'number_inpatient**2' has been created and is listed in the table. The 'Transformations' column for this feature shows 'log(number_inpatient)' and 'f(number_inpatient)'. The 'Var Type' column shows 'Categorical' for this specific transformation. Other features like 'diag_2' and 'num_medications' are also listed with their respective statistics.

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
number_inpatient	0.39	Numeric	11	0	0.39	0.85	0	0
number_inpatient	-0.0	Categorical	52	4,100				
number_inpatient**2	-0.0	Categorical	10	936				
diag_2	-0.0	Categorical	429	59				
num_medications	-0.0	Numeric	68	0	15.56	8.39	14	1
num_lab_procedures	-0.0	Numeric	108	0	43.08	19.45	44	1
diag_2_desc	-0.0	Text	429	59				
weight	-0.0	Categorical	7	9,592				
age	-0.0	Categorical	10	0				
diag_1_desc	-0.0	Text	457	2				
payer_code	-0.0	Categorical	15	5,341				
diag_3_desc	-0.0	Text	460	208				

2. The transformed feature appears under the original feature. It can be included in any new feature lists, and can also be used for modeling.



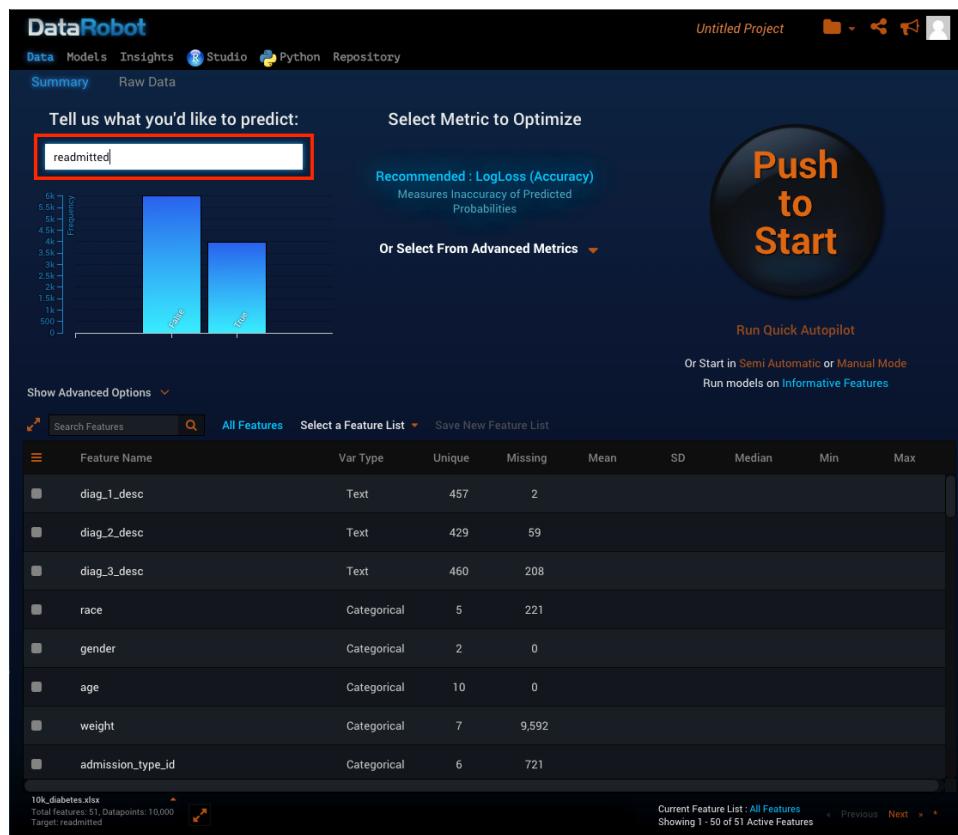
Running Models

After you create a DataRobot project, you can select a target feature and modeling options, and then start the modeling process. You can also add new models, create blended models, and run selected models.

Select a Target Feature

After you create a project, you can select a target feature and an optimization metric and start modeling. The target feature is the name of the column in the dataset you would like to predict. The optimization metric is the metric software used to optimize the models.

On the Data Summary page, type in the name of the target feature you would like to predict.



Select an Optimization Metric

After you create a project, you can select a target feature and an optimization metric and start modeling. The target feature is the name of the column in the dataset you would like to predict. The optimization metric is the metric software used to optimize the models.

After you type in a target feature, a recommended optimization metric is automatically selected based on the modeling task.

- If the selected target has only two unique values, the assumption is made that the task is a classification, and a classification metric will be recommended. Examples of recommended classification methods include LogLoss (if it is necessary to calculate a probability for each class), and Gini and AUC when it is necessary to sort records in order of ranking.

- Otherwise the assumption is made that the selected target represents a regression task. The most popular metrics for regression are RMSE (Root Mean Square Error) and MAD (Mean Absolute Deviation).

If you would like to override the recommended optimization metric, click **Select from Advanced Metrics**, then select an alternate metric.

The screenshot shows the DataRobot interface with the 'Data' tab selected. On the left, there's a histogram for the 'readmitted' feature. Below it is a list of features with their types and statistics. A dropdown menu titled 'Select Metric to Optimize' is open, listing several metrics:

- Recommended : LogLoss (Accuracy)**: Measures Inaccuracy of Predicted Probabilities
- AUC (Suggested)**: Measures ability to Distinguish the Ones from the Zeros
- Gini Norm (Suggested)**: Measures Ability to Rank
- RMSE (Suggested)**: Measures Inaccuracy of predicted mean values when the target is normally distributed
- FVE Binomial**: Fraction of variance explained for binomial deviance
- Rate@Top10%**: Response rate in top 10% highest predictions
- Rate@Top5%**

On the right, there's a large orange button labeled 'Push to Start' and a section for 'Run Quick Autopilot'.

Set Advanced Modeling Parameters

To set advanced modeling parameters, click **Show Advanced Options** on the Data Summary page.

The screenshot shows the DataRobot interface for an 'Untitled Project'. At the top, there are tabs for Data, Models, Insights, R Studio, Python, and Repository. Below that, 'Summary' and 'Raw Data' are selected. On the left, a bar chart titled 'Tell us what you'd like to predict:' shows two categories: 'readmitted' (approx. 5.5k) and 'not readmitted' (approx. 3.5k). To the right, a section titled 'Select Metric to Optimize' shows 'Recommended : LogLoss (Accuracy)' and 'Measures Inaccuracy of Predicted Probabilities'. A large orange button labeled 'Push to Start' is prominent. Below it, a link says 'Run Quick Autopilot' and 'Or Start in Semi Automatic or Manual Mode'. A table below lists features: diag_1_desc, diag_2_desc, diag_3_desc, race, gender, age, weight, and admission_type_id. At the bottom, it says '10k_diabetes.xlsx Total Features: 87, Datapoints: 10,000 Target: readmitted'.

The advanced modeling options appear.

This screenshot shows the same DataRobot interface as above, but with more advanced options visible. Under 'Tell us what you'd like to predict:', the 'Show Advanced Options' button is highlighted. The 'Select Metric to Optimize' section remains the same. The 'Push to Start' button is still present. The feature list table is identical. At the bottom, the file information is the same. The new content includes a 'Select Partitioning Method' section with 'Stratified' selected, showing a stratified sampling diagram. It also includes sections for 'Recommender', 'Additional Parameters', and 'Run models using:' with 'Cross Validation' selected. There are also sections for 'Holdout Percentage' (set to 20%), 'Random Seed' (set to 0), and 'Train/Validate/Holdout Split'.

You can specify the following advanced options:

Partitioning Method

Random

Description: Observations are randomly assigned to the training, validation, and holdout sets.

Modeling Options:

- Cross-Validation -- specify the number of folds and holdout percentage.
- Training/Validation/Holdout Split -- specify the percentages for training, validation, and holdout.
- Random Seed -- specify a positive integer value for the random seed.

Partition Column

Description: A column in the data file is used to either specify a train/validate/holdout split or the folds/holdout to be used for cross-validation.

Modeling Options:

- Cross-Validation -- you must specify a value from the selected partition column that will specify the holdout set.
- Training/Validation/Holdout Split -- you must specify training, validation, and holdout set values from the selected partition column.

Group

Description: One or more columns can be selected, and each combination of these values is guaranteed to be in the same training or test set as other matching values.

Modeling Options:

- Cross-Validation -- specify the number of folds and holdout percentage.
- Training/Validation/Holdout Split -- specify the percentages for training, validation, and holdout.

Date

Description: Observations in the holdout set come after observations in the validation set chronologically, and observations in the validation set come after those in the training set. For this method you specify the Date/Time column in the dataset, which is used to create a Train/Validate/Holdout split with all rows in the test set occurring later than all rows in the training set, and all rows in holdout occurring later than the test set.

Modeling Options:

- Validation Percentage -- percentage of data allocated to the validation set.
- Holdout Percentage -- percentage of data allocated to the holdout set.

Stratified

Description: Observations are randomly assigned to training, validation, and holdout sets, preserving the same ratio of positive to negative cases as in the original data.

Modeling Options:

- Cross-Validation -- specify the number of folds and holdout percentage.
- Training/Validation/Holdout Split -- specify the percentages for training, validation, and holdout.

Recommender

Select the **Is this a Recommendation Problem?** check box if you would like to predict the rating or preference that a user would give to an item.

If you select this option, you must specify the two columns that contain the user and item identifiers.

A typical example for recommender systems is a product recommendation on an e-commerce site where you would like to predict how highly a user might rate a product. Predicted ratings are then used for content discovery on the site.

For recommendation problems, DataRobot runs models that use content-based and collaborative filtering techniques.

- Content-based models take into account any available user or item features and create a profile for each user (key words, price range, product types, brands, etc.).
- Collaborative filtering techniques are content-agnostic -- they look solely at the latent structure in the user-item ratings in order to make predictions.

DataRobot also runs hybrid models that combine both content-based and collaborative filtering techniques, as well as combinations of models (blended models).

Additional Parameters

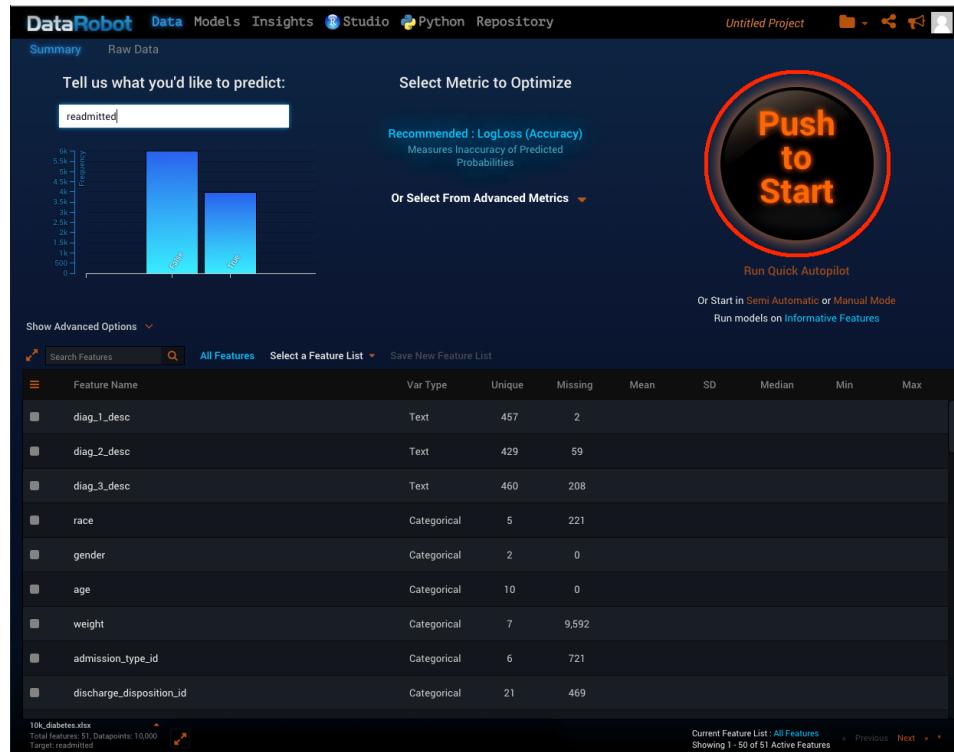
- Weight -- must be the name of a column in the dataset.
- Exposure -- must be the name of a column in the dataset.
- Upper Bound Running Time -- models that take longer to execute than this value (hrs) are excluded in subsequent autopilot runs.
- Response Cap -- The Response Cap limits the maximum value of the response (target) to a percentage of the original values. The value must be between 0.5 and 1.0 (50-100%).

Start Modeling

Start Modeling

After you have selected a target feature and an optimization metric, you can start modeling. The modeling process finds the best predictive models for the target feature.

To start modeling, click the **Push to Start** button.



The screenshot shows the DataRobot interface for a project titled "Untitled Project". The top navigation bar includes links for Data, Models, Insights, Studio, Python Repository, and a file menu. The main workspace is titled "Summary" and "Raw Data". On the left, a search bar contains the text "readmitted". A histogram shows the frequency distribution of this target variable. To the right, a section titled "Select Metric to Optimize" recommends "LogLoss (Accuracy)" as the "Measures Inaccuracy of Predicted Probabilities". Below this, there's a link to "Or Select From Advanced Metrics". A large, prominent button in the center-right is labeled "Push to Start" with a red circular highlight around it. Below the button, there's a link to "Run Quick Autopilot" and a note about starting in "Semi Automatic or Manual Mode". At the bottom, a table lists various features with their statistics: diag_1_desc (Text, 457 unique, 2 missing), diag_2_desc (Text, 429 unique, 59 missing), diag_3_desc (Text, 460 unique, 208 missing), race (Categorical, 5 unique, 221 missing), gender (Categorical, 2 unique, 0 missing), age (Categorical, 10 unique, 0 missing), weight (Categorical, 7 unique, 9.592 mean), admission_type_id (Categorical, 6 unique, 721 missing), and discharge_disposition_id (Categorical, 21 unique, 469 missing). The bottom status bar indicates "10k_diabetes.xlsx", "Total Features: 51, Datapoints: 10,000", "Target: readmitted", "Current Feature List: All Features", "Showing 1 - 50 of 51 Active Features", and navigation buttons for "Previous" and "Next".

Autopilot Modes

The modeling process runs in Autopilot (fully automatic) mode by default, which means that the DataRobot platform will automatically select the best predictive models for the specified target feature.

For more control over which models to run, you can select **Run Quick Autopilot**, **Semi Automatic**, or **Manual Mode** under the **Push to Start** button.

The screenshot shows the DataRobot interface for a project titled "Untitled Project". At the top, there are tabs for Data, Models, Insights, Studio, Python, and Repository. Below the tabs, there's a summary section with a bar chart showing the frequency of the target variable "readmitted" (0 and 1). To the right of the chart is a large orange button labeled "Push to Start". Above the button, it says "Select Metric to Optimize" and "Recommended : LogLoss (Accuracy) Measures Inaccuracy of Predicted Probabilities". Below the button are options for "Run Quick Autopilot", "Semi Automatic", and "Manual Mode". A note says "Run models on Informative Features".

Feature Name	Var Type	Unique	Missing	Mean	SD	Median	Min	Max
diag_1_desc	Text	457	2					
diag_2_desc	Text	429	59					
diag_3_desc	Text	460	208					
race	Categorical	5	221					
gender	Categorical	2	0					
age	Categorical	10	0					
weight	Categorical	7	9,592					
admission_type_id	Categorical	6	721					
discharge_disposition_id	Categorical	21	469					

10K.diabetes.xlsx
Total features: 51, Datapoints: 10,000
Target: readmitted

Current Feature List: All Features
Showing 1 - 50 of 51 Active Features

- **Run Quick Autopilot** -- The Quick Autopilot runs a small subset of models consisting of the best models based on the specified target feature and performance metric.
- **Semi Automatic** -- In semi-automatic mode, the autopilot will pause after the first cross-validation set before submitting additional models to the job queue. This allows you to inspect the models and choose to either add additional models or delete models.
- **Manual Mode** -- Starting the autopilot in manual mode gives you full control over which models to execute. For example, you can choose a specific model from the Task Repository rather than running the models selected by default.

The Worker Queue

After you start the modeling process, running and pending jobs appear in the Worker Queue at the right of the page. When a model is running, a graphical display of its CPU and RAM use will be displayed in the Worker Queue. The queue will execute a maximum fixed number of workers per project at any one time.

The screenshot shows the DataRobot interface with the following details:

- Top Navigation:** DataRobot, Data, Models, Insights, Studio, Python Repository, Untitled Project.
- Left Panel:** Summary, Raw Data, Search Features, All Features, Select a Feature List, Save New Feature List.
- Feature Importance Table:**

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	Target	Boolean	2	0	0.40	0.49	0	
discharge_disposition_id	Categorical	21	469					
number_diagnoses	Numeric	9	0	7.03	2.02	7		
number_inpatient	Numeric	11	0	0.39	0.85	0		
medical_specialty	Categorical	52	4,100					
admission_source_id	Categorical	10	936					
diag_3	Categorical	460	208					
diag_2	Categorical	429	59					
num_medications	Numeric	68	0	15.56	8.39	14		
num_lab_procedures	Numeric	108	0	43.08	19.45	44		
diag_2_desc	Text	429	59					
weight	Categorical	7	9,592					
age	Categorical	10	0					
diag_1_desc	Text	457	2					
payer_code	Categorical	15	5,341					
diag_3_desc	Text	460	208					
time_in_hospital	Numeric	14	0	4.43	3.02	4		
- Bottom Left:** 10K_diabetes.xlsx, Total features: 51, Datapoints: 10,000, Target: readmitted, Metric: LogLoss.
- Bottom Center:** Current Feature List: All Features, Showing 1 - 50 of 51 Active Features.
- Right Panel:**
 - Workers:** 002 (orange box highlights the number).
 - Processing (2):**
 - ExtraTrees Classifier (Gini) (26)
 - Nystroem Kernel SVM Classifier
 - Queue (5):**
 - Gradient Boosted Trees Clas...
 - eXtreme Gradient Boosted Tr...
 - Gradient Boosted Greedy Tre...
 - Auto-tuned K-Nearest Neighb...
 - Elastic-Net Classifier (mix...)

Adjusting the Number of Workers

You can adjust the maximum number of simultaneous workers by clicking the orange arrows at the right of the green numbers in the Workers field at the top of the Worker Queue.

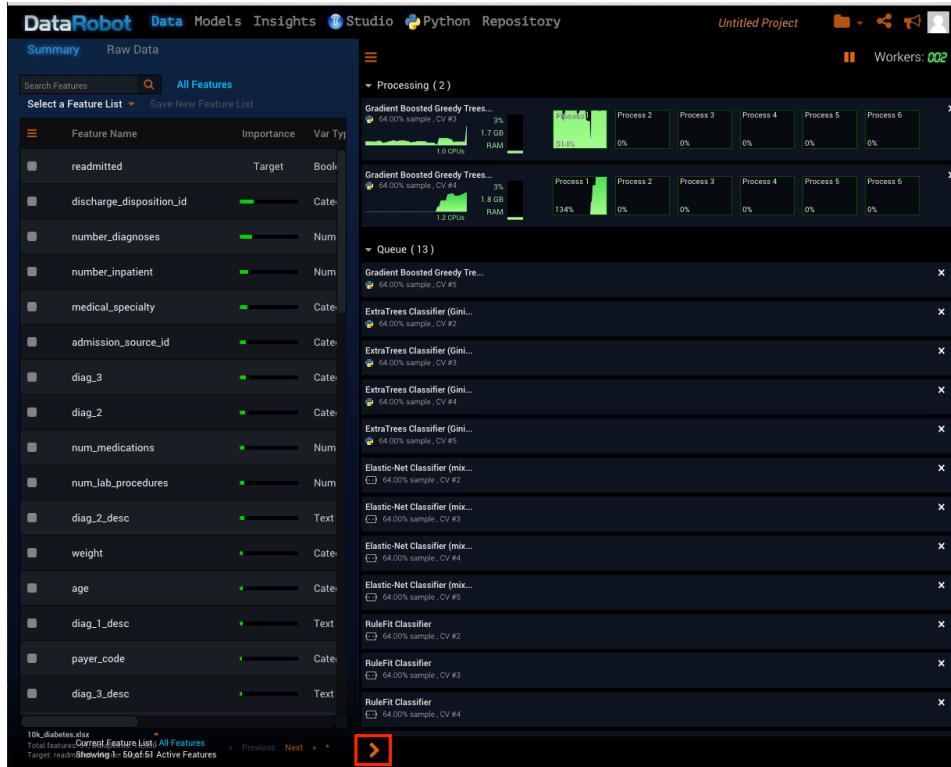
The screenshot shows the DataRobot interface with the following details:

- Top Navigation:** DataRobot, Data, Models, Insights, Studio, Python Repository, Untitled Project.
- Left Panel:** Summary, Raw Data, Search Features, All Features, Select a Feature List, Save New Feature List.
- Feature Importance Table:**

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	Target	Boolean	2	0	0.40	0.49	0	
discharge_disposition_id	Categorical	21	469					
number_diagnoses	Numeric	9	0	7.03	2.02	7		
number_inpatient	Numeric	11	0	0.39	0.85	0		
medical_specialty	Categorical	52	4,100					
admission_source_id	Categorical	10	936					
diag_3	Categorical	460	208					
diag_2	Categorical	429	59					
num_medications	Numeric	68	0	15.56	8.39	14		
num_lab_procedures	Numeric	108	0	43.08	19.45	44		
diag_2_desc	Text	429	59					
weight	Categorical	7	9,592					
age	Categorical	10	0					
diag_1_desc	Text	457	2					
payer_code	Categorical	15	5,341					
diag_3_desc	Text	460	208					
time_in_hospital	Numeric	14	0	4.43	3.02	4		
- Bottom Left:** 10K_diabetes.xlsx, Total features: 51, Datapoints: 10,000, Target: readmitted, Metric: LogLoss.
- Bottom Center:** Current Feature List: All Features, Showing 1 - 50 of 51 Active Features.
- Right Panel:**
 - Workers:** 002 (orange box highlights the number).
 - Processing (2):**
 - ExtraTrees Classifier (Gini) (26)
 - Nystroem Kernel SVM Classifier
 - Queue (5):**
 - Gradient Boosted Trees Clas...
 - eXtreme Gradient Boosted Tr...
 - Gradient Boosted Greedy Tre...
 - Auto-tuned K-Nearest Neighb...
 - Elastic-Net Classifier (mix...)

Viewing Worker Queue Details

To view more details about the running jobs, click the orange arrow at the bottom of the Worker Queue. Click the arrow again to hide queue details.



Pausing the Worker Queue

To pause the Worker Queue, click the Pause symbol at the top of the Worker Queue. After you pause the queue, the Pause symbol changes to a Play symbol (arrow). To resume running models, click the Play arrow.

The screenshot shows the DataRobot interface with the following details:

- Top Navigation:** DataRobot, Data, Models, Insights, Studio, Python Repository, Untitled Project.
- Left Panel:** Summary, Raw Data, Search Features, All Features, Select a Feature List, Save New Feature List.
- Feature List:**

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	Target	Boolean	2	0	0.40	0.49	0	
discharge_disposition_id	Categorical	21	469					
number_diagnoses	Numeric	9	0	7.03	2.02	7		
number_inpatient	Numeric	11	0	0.39	0.85	0		
medical_specialty	Categorical	52	4,100					
admission_source_id	Categorical	10	936					
diag_3	Categorical	460	208					
diag_2	Categorical	429	59					
num_medications	Numeric	68	0	15.56	8.39	14		
num_lab_procedures	Numeric	108	0	43.08	19.45	44		
diag_2_desc	Text	429	59					
weight	Categorical	7	9,592					
age	Categorical	10	0					
diag_1_desc	Text	457	2					
payer_code	Categorical	15	5,341					
diag_3_desc	Text	460	208					
time_in_hospital	Numeric	14	0	4.43	3.02	4		
- Bottom Left:** 10K_diabetes.xlsx, Total features: 51, Datapoints: 10,000, Target: readmitted, Metric: LogLoss.
- Bottom Center:** Current Feature List: All Features, Showing 1 - 50 of 51 Active Features.
- Right Panel:**
 - Workers:** 002
 - Processing (2):**
 - ExtraTrees Classifier (Gini) (26)
 - Nystroem Kernel SVM Classifier
 - Queue (5):**
 - Gradient Boosted Trees Clas...
 - eXtreme Gradient Boosted Tr...
 - Gradient Boosted Greedy Tre...
 - Auto-tuned K-Nearest Neighb...
 - Elastic-Net Classifier (mix...)

Killing Workers

You can kill a worker by clicking the X next to the job name in the Worker Queue. If a worker fails for any reason, it will be listed under Errors at the bottom of the Worker Queue.

The screenshot shows the DataRobot interface with the following details:

- Top Navigation:** DataRobot, Data, Models, Insights, Studio, Python Repository, Untitled Project.
- Left Panel:** Summary, Raw Data, Search Features, All Features, Select a Feature List, Save New Feature List.
- Feature List:**

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	Target	Boolean	2	0	0.40	0.49	0	
discharge_disposition_id	Categorical	21	469					
number_diagnoses	Numeric	9	0	7.03	2.02	7		
number_inpatient	Numeric	11	0	0.39	0.85	0		
medical_specialty	Categorical	52	4,100					
admission_source_id	Categorical	10	936					
diag_3	Categorical	460	208					
diag_2	Categorical	429	59					
num_medications	Numeric	68	0	15.56	8.39	14		
num_lab_procedures	Numeric	108	0	43.08	19.45	44		
diag_2_desc	Text	429	59					
weight	Categorical	7	9,592					
age	Categorical	10	0					
diag_1_desc	Text	457	2					
payer_code	Categorical	15	5,341					
diag_3_desc	Text	460	208					
time_in_hospital	Numeric	14	0	4.43	3.02	4		
- Bottom Left:** 10K_diabetes.xlsx, Total features: 51, Datapoints: 10,000, Target: readmitted, Metric: LogLoss.
- Bottom Center:** Current Feature List: All Features, Showing 1 - 50 of 51 Active Features.
- Right Panel:**
 - Workers:** 002
 - Processing (2):**
 - ExtraTrees Classifier (Gini) (26)
 - Nystroem Kernel SVM Classifier
 - Queue (5):**
 - Gradient Boosted Trees Clas...
 - eXtreme Gradient Boosted Tr...
 - Gradient Boosted Greedy Tre...
 - Auto-tuned K-Nearest Neighb...
 - Elastic-Net Classifier (mix...)

After modeling has started, you can click **Models** in the top navigation menu to view the Leaderboard. The Leaderboard is a list of models ranked by the chosen performance metric, with the best models at the top

of the list. The rankings keep changing until all models have finished running. You cannot view full details about a model until it finishes running.

When all workers have finished running, the Worker Queue displays a message at the top of the queue and provide options to rerun the models a different feature list, or to unlock the holdout dataset and continue running workers.

Feature Name	Importance	Var Type	Unique	Missing	Mean	SD	Median	Min
readmitted	Target	Boolean	2	0	0.40	0.49	0	
discharge_disposition_id		Categorical	21	469				
number_diagnoses		Numeric	9	0	7.03	2.02	7	
number_inpatient		Numeric	11	0	0.39	0.85	0	
medical_specialty		Categorical	52	4,100				
admission_source_id		Categorical	10	936				
diag_3		Categorical	460	208				
diag_2		Categorical	429	59				
num_medications		Numeric	68	0	15.56	8.39	14	
num_lab_procedures		Numeric	108	0	43.08	19.45	44	
diag_2_desc		Text	429	59				
weight		Categorical	7	9,592				
age		Categorical	10	0				
diag_1_desc		Text	457	2				
payer_code		Categorical	15	5,341				

10k_diabetes.xlsx
Total features: 51, Datapoints: 10,000
Target: readmitted, Metric: LogLoss

Current Feature List: All Features
Showing 1 - 50 of 51 Active Features

Workers: 002

Autopilot has finished

- Run Autopilot On A Different Feature List
- Unlock Holdout
- View Summary

Datarobot Prime

Add a New Model

Use the following steps to create a new model.

1. Click **Add New Model** at the top of the model Leaderboard.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. At the top, there's a search bar labeled 'Search Leaderboard' and a button 'Add New Model+'. Below the search bar, the table lists various models with their names, descriptions, feature lists, sample sizes, validation metrics, cross-validation metrics, and holdout status. The first few models listed are ENET Blender, Advanced AVG Blender, AVG Blender, GLM Blender, Nystroem Kernel SVM Classifier, Advanced GLM Blender, Nystroem Kernel SVM Classifier, Regularized Logistic Regression, Elastic-Net Classifier, Gradient Boosted Greedy Trees Classifier, ExtraTrees Classifier, and Elastic-Net Classifier. The interface also shows a sidebar with 'Autopilot has finished' and 'Datarobot Prime' sections.

2. Select a model type and a feature list, specify sample size and the number of cross-validation runs, then click **Submit**.

The screenshot shows the 'Add a new model' dialog box. It includes fields for 'Select a model:' (set to 'Stochastic Gradient Descent Classifier'), 'Run on feature list:' (set to 'Informative Features'), 'Sample Size:' (set to '80% + 8000 of 10000 Rows'), and 'CV Runs:' (set to '1'). Below the dialog is the same Leaderboard table as the previous screenshot, showing the same list of models. The sidebar on the right remains the same.

3. The new model appears in the list on the Leaderboard.

The screenshot shows the DataRobot interface with the 'Models' tab selected. The main area is the 'Leaderboard' showing various machine learning models. One model, 'Stochastic Gradient Descent Classifier (33)', is highlighted with a red box. The table columns include Model Name and Description, Feature List, Sample Size, Validation, Cross Validation, and Hold Out. The validation metric is LogLoss. The sidebar on the right displays 'Autopilot has finished' with options like 'Run Autopilot On A Different Feature List', 'Unlock Holdout', and 'View Summary'. It also shows 'Workers: 002' and a 'Datarobot Prime' icon.

Create a Blended Model

Blending enables you to combine the predictions of two or more models, which often leads to better results than running either of the individual models. Currently DataRobot provides options for Average, GLM, ENET, PLS, and Median blending. For each target point, the Average and Median blenders will calculate the average or median values of the predictions of the selected individual models. GAM ([Generalized Additive Model](#)) and GLM (Generalized Linear Model) blenders are essentially a second layer of models on the top of the existing models. They use the predictions of the selected models as predictors for GAM and GLM, while keeping the same target as individual models.

Use the following steps to create a blended model.

1. Use the check boxes on the left side of the model Leaderboard to select two or more models.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. The page displays a list of machine learning models, each with its name, description, feature list, sample size, validation metric, cross-validation metric, and a 'Run' button. A red box highlights the 'RuleFit Classifier (14)' model.

Model Name and Description	Feature List	Sample Size	Validation	Cross Validation	Hold Out
Nystrom Kernel SVM Classifier (20)	Informative Features	32.0 %	0.6193	Run	🔒
Regularized Logistic Regression (L2) (6)	Informative Features	32.0 %	0.6203	Run	🔒
RuleFit Classifier (14)	Informative Features	32.0 %	0.6205	Run	🔒
Elastic-Net Classifier (mixing alpha=0.5 / Binomial Deviance) (24)	Informative Features	32.0 %	0.6212	Run	🔒
Elastic-Net Classifier (mixing alpha=0.5 / Binomial Deviance) (17)	Informative Features	32.0 %	0.6212	Run	🔒
ExtraTrees Classifier (Gini) (26)	Informative Features	32.0 %	0.6218	Run	🔒
Elastic-Net Classifier (mixing alpha=0.5 / Binomial Deviance) (23)	Informative Features	32.0 %	0.6224	Run	🔒
Gradient Boosted Trees Classifier (3)	Informative Features	32.0 %	0.6230	Run	🔒
Regularized Logistic Regression (L2) (21)	Informative Features	32.0 %	0.6274	Run	🔒

10K_diabetes.xlsx
Total features: 81, Datapoints: 10,000
Target: readmitm, Metric: LogLoss

- Click the model menu icon at the top left of the Leaderboard, then select one of the blending options listed under **Blending**. Hover over each menu item displays a description of the blending option.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. The 'Blending' menu is open, showing various blending options: Average Blend, GLM Blend, ENET Blend, PLS Blend, Median Blend, Select All, and Select None. A red box highlights the 'GLM Blend' option, and a tooltip appears stating 'Blend the selected models with a Elastic Net Model'.

Model Name and Description	Feature List	Sample Size	Validation	Cross Validation	Hold Out
Trees Classifier with Early Stopping (30)	Informative Features	64.0 %	0.6139	0.6118	🔒
(26)	Informative Features	64.0 %	0.6143	0.6105	🔒
Blending x14	Informative Features	64.0 %	0.6148	0.6108	🔒
Select All	Informative Features	32.0 %	0.6186	Run	🔒
Select None	Informative Features	32.0 %	0.6193	Run	🔒
Regularized Logistic Regression (L2) (6)	Informative Features	32.0 %	0.6203	Run	🔒
RuleFit Classifier (14)	Informative Features	32.0 %	0.6205	Run	🔒
Elastic-Net Classifier (mixing alpha=0.5 / Binomial Deviance) (24)	Informative Features	32.0 %	0.6212	Run	🔒

10K_diabetes.xlsx
Total features: 81, Datapoints: 10,000
Target: readmitm, Metric: LogLoss

- A new job appears in the Worker queue while the blended model is processed.

The screenshot shows the DataRobot interface with the 'Models' tab selected. The left pane displays the 'Leaderboard' with various models listed, including 'ENET Blender (14+24)'. The right pane shows the 'Autopilot' status, indicating a processing job for 'ENET Blender (14+24)' with a progress bar at 3%.

- When processing is complete, the new blended model appears in the list on the Leaderboard.

The screenshot shows the DataRobot interface with the 'Models' tab selected. The left pane displays the 'Leaderboard' with the 'ENET Blender (14+24)' model now highlighted. The right pane shows the 'Autopilot' status, indicating the process has finished.

Run Selected Models

After models are created, you can use **Run Selected Models** to retrain selected models with a different set of parameters.



Note:

You cannot use **Run Selected Models** on blended models.

Use the following steps to run one or more selected models.

- Use the check boxes on the left side of the model Leaderboard to select one or more models. Click the model menu icon at the top left of the Leaderboard, then click **Run Selected Model(s)**.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. At the top, there are buttons for 'Search Leaderboard', 'Add New Model', and a dropdown for 'Metric: LogLoss'. Below these are sections for 'Comparison', 'Batch Processing', and 'Add New Model'. The main area displays a list of models with columns for 'Model Name and Description', 'Feature List', 'Sample Size', 'Validation', 'Cross Validation', and 'Hold Out'. A red box highlights the 'Run Selected Model(s)' button at the top of the list. On the right side, there's a sidebar titled 'Autopilot has finished' with options like 'Run Autopilot On A Different Feature List', 'Unlock Holdout', and 'View Summary'. A 'Datarobot Prime' badge is also present.

2. Use the box at the top of the Leaderboard to specify a feature list, sample size, and the number of cross-validation (CV) runs, then click **Submit** to retrain the selected models with the specified parameters.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. A modal dialog is open, prompting the user to 'Run 2 selected models with the following options :'. The dialog includes fields for 'Run on feature list' (set to 'Univariate Selections'), 'Samplesize' (set to '75%'), and 'CV Runs' (set to 'all'). A red box highlights the 'Submit' button. The background shows the same model list as the previous screenshot. On the right side, there's a sidebar titled 'Autopilot has finished' with options like 'Run Autopilot On A Different Feature List', 'Unlock Holdout', and 'View Summary'. A 'Datarobot Prime' badge is also present.

3. New jobs appear in the Worker queue while the models are processed.

The screenshot shows the DataRobot interface with the 'Models' tab selected. The main area is the 'Leaderboard' showing various machine learning models with their names, descriptions, feature counts, sample sizes, validation metrics, cross-validation metrics, and holdout metrics. Below the leaderboard, it says '10k_diabetes.xlsx' with 'Total features: 81, Datapoints: 10,000' and 'Target: readmitted, Metric: LogLoss'. On the right side, there's a 'Workers' section with a 'DataRobot Prime' button. A red box highlights the 'Processing (1)' section, which shows a specific model named 'Nystroem Kernel SVM Classifier' with its progress bar at 80%, memory usage of 3% (2.2 GB), and RAM usage of 0.8 GB.

DataRobot Prime

You can use DataRobot Prime to build a flexible model that finds a balance between complexity and accuracy.

Use the following steps to create an optimized model using DataRobot Prime.

1. In the Worker Queue at right of the model Leaderboard, click **DataRobot Prime**.

This screenshot is similar to the previous one but shows a different state in the 'Workers' section. The 'Autopilot has finished' message is still present, but the 'DataRobot Prime' button is now highlighted with a red box. This indicates that a new job has been initiated or is currently processing.

2. The DataRobot Prime job progress is displayed in the Worker Queue.

The screenshot shows the DataRobot interface. On the left, the Model Leaderboard displays various machine learning models with their names, descriptions, feature lists, sample sizes, validation metrics, cross-validation metrics, and hold-out metrics. One model, "DataRobot Prime (29)", is highlighted with a red border. On the right, the "Processing (5)" queue shows five DataRobot Prime models (29, 30, 31, 32, 33) each with a progress bar and memory usage information (1.2 GB, 1.2 GB, 1.2 GB, 1.3 GB, 1.1 GB RAM).

- When processing is complete, the DataRobot Prime model appears in the list on the model Leaderboard.

This screenshot is identical to the one above, but the "DataRobot Prime (29)" model has been successfully processed and is now listed at the top of the Model Leaderboard with a red border around its row. The other models remain in the list below it.

Delete Models

Use the following steps to delete models.

Use the check boxes on the left side of the model Leaderboard to select one or more models. Click the model menu icon at the top left of the Leaderboard, then click **Delete Selected Model(s)**.

DataRobot Data Models Insights Studio Python Repository Untitled Project Workers: 202 Autopilot has finished

Search Leaderboard Add New Model + Metric: LogLoss

Model Name and Description	Feature List	Sample Size	Validation	Cross Validation	Hold Out
Comparison					
Compare Selected	Classifier (20) using v4	Informative Features 80.0 %	60.6064	60.6119 *	Locked
Batch Processing	0+24+2... (93)	Informative Features 64.0 %	60.6072	60.6048	Locked
Add New Model	+14+17+20+24+2... (92)	Informative Features 64.0 %	60.6075	60.6054	Locked
Run Selected Model(s)	+14+17+20+24+2... (92)	Informative Features 80.0 %	60.6079	60.6119 *	Locked
Delete Selected Model(s)	Remove Models from Leaderboard				
Blending					
Average Blend	(90)	Informative Features 64.0 %	60.6081	60.6064	Locked
GLM Blend	(89)	Informative Features 64.0 %	60.6088	60.6053	Locked
ENET Blend	Classifier (20) using v4	Informative Features 64.0 %	60.6094	60.6106	Locked
PLS Blend					
Median Blend					
Select All	+14+17+20+24+2... (91)	Informative Features 64.0 %	60.6100	60.6049	Locked
Select None	Classifier (27)	Informative Features 64.0 %	60.6110	60.6097	Locked
Regularized Linear Model Preprocessing v5					
ENET Blender (14+24) (101)		Informative Features 64.0 %	60.6117	60.6074	Locked
Regularized Logistic Regression (L2) (6)	One Hot Encoding Missing Values Imputed Standardized Regularized Logistic Regression (L2)	Informative Features 64.0 %	60.6132	60.6116	Locked
Elastic-Net Classifier (mixing alpha=0.5 / Binomial Deviance) (17)	Regularized Linear Model Preprocessing v1	Informative Features 64.0 %	60.6136	60.6118	Locked
10k_diabetes.xlsx	Total features: 81, Datapoints: 10,000 Target: readmission, Metric: LogLoss				

Autopilot has finished

- Run Autoplot On A Different Feature List
- Unlock Holdout
- View Summary

Datarobot Prime

10k_diabetes.xlsx Total features: 81, Datapoints: 10,000 Target: readmission, Metric: LogLoss

Evaluating Models

After you create a DataRobot project, you can select a target feature and start the modeling process. You can then evaluate and select the best models to use for prediction.

The Model Leaderboard

Click **Models** in the top navigation menu to view the Leaderboard. The Leaderboard is a list of models ranked by the chosen performance metric, with the best models at the top of the list. You cannot view full details about a model until it finishes running.

Model Name and Description	Feature List	Sample Size	Validation	Cross Validation	Hold Out
ENET Blender (6+14+17+20+24+2...) (93)	Informative Features	64.0 %	0.6072	0.6048	
Advanced AVG Blender (6+14+17+20+24+2...) (92)	Informative Features	64.0 %	0.6075	0.6054	
AVG Blender (20+26+27) (90)	Informative Features	64.0 %	0.6081	0.6064	
GLM Blender (20+26+27) (89)	Informative Features	64.0 %	0.6088	0.6053	
Nystroem Kernel SVM Classifier (20)	Informative Features	64.0 %	0.6094	0.6106	
Regularized Linear Model Preprocessing v4					
Advanced GLM Blender (6+14+17+20+24+2...) (91)	Informative Features	64.0 %	0.6100	0.6049	
(6+14+17+20+24+2...) (91)					
Nystroem Kernel SVM Classifier (27)	Informative Features	64.0 %	0.6110	0.6097	
Regularized Logistic Regression (L2) (6)	Informative Features	64.0 %	0.6132	0.6116	
One-Hot Encoding Missing Values Imputed Standardize Regularized Logistic Regression (L2)					
Elastic-Net Classifier (mixing alpha=0.5 / Binomial Deviance) (17)	Informative Features	64.0 %	0.6136	0.6118	
Regularized Linear Model Preprocessing v1					
Gradient Boosted Greedy Trees Classifier with Early Stopping (30)	Informative Features	64.0 %	0.6139	0.6118	
10K_diabetes.xlsx					
Total Features: 51, Datapoints: 10,000					
Target: readmitted, Metric: LogLoss					

Click a model name to view details about the model. Model details are discussed in the following sections.

Blueprint

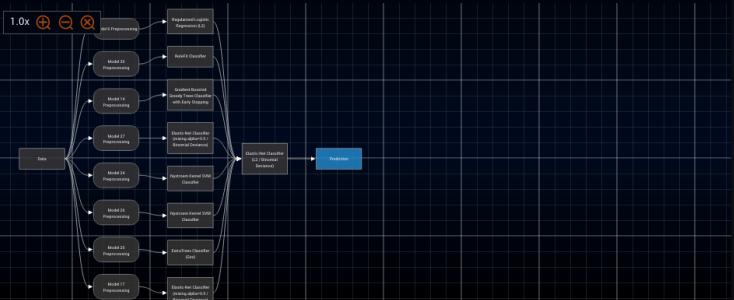
Blueprints provide a graphical representation of the many steps involved in transforming input predictors and targets into a model. A blueprint represents the high-level end-to-end procedure for fitting the model, including any pre-processing steps, algorithms, and post-processing. Each box in a blueprint may represent multiple steps.

To view a graphical representation of a blueprint, click a model in the leaderboard, click the Blueprint link in the navigation toolbar, or zoom in on the blueprint.

DataRobot Data Models Insights Studio Python Repository Untitled Project

Leaderboard Learning Curves Speed vs Accuracy Comparison

Search Leaderboard Add New Model + Metric: LogLoss

Model Name and Description	Feature List	Sample Size	Validation	Cross Validation	Hold Out
ENET Blender (6+14+17+20+24+2...) (93)	Informative Features	64.0 %	0.6072	0.6048	
Blueprint Lift Chart Model X-Ray Model Info Model Log ROC Curve Grid Search Deploy Model Predict					
					
Advanced AVG Blender (6+14+17+20+24+2...) (92)	Informative Features	64.0 %	0.6075	0.6054	
AVG Blender (20+26+27) (90)	Informative Features	64.0 %	0.6081	0.6064	
GLM Blender (20+26+27) (89)	Informative Features	64.0 %	0.6088	0.6053	
10K_diabetes.xlsx Total features: 9, 51 Datapoints: 10,000 Target readmitted; Metric: LogLoss					
Autopilot has finished					
Run Autopilot On A Different Feature List					
Unlock Holdout					
View Summary					
Datarobot Prime					

Input Types

Different columns in the dataset require different types of preparation and transformation. For example, some algorithms recommend subtracting the mean and dividing by the standard deviation of the input data -- but this would not make sense for text input data. The first step in the execution of a blueprint is to identify data types that belong together so they can be processed separately.

Transformers and Models

The other nodes in the blueprint are other types of data transformations or models. Click a blueprint node to display additional information. Many of the models use DataRobot proprietary approaches to data pre-processing.

The screenshot shows the DataRobot interface. In the top navigation bar, the tabs are Data, Models, Insights, Studio, Python, and Repository. The current project is titled "Untitled Project". On the left, there's a "Leaderboard" tab, followed by Learning Curves, Speed vs Accuracy, and Comparison. The main area is titled "Search Leaderboard" and "Add New Model +". A dropdown menu for "Metric: LogLoss" is open. Below this, the "Model Name and Description" table lists several models:

- ENET Blender (6+14+17+20+24+2...) (93)
- Advanced AVG Blender (6+14+17+20+24+2...) (92)
- AVG Blender (20+26+27) (90)
- GLM Blender (20+26+27) (89)
- Nystroem Kernel SVM Classifier (20)

Each model entry includes columns for Feature List, Sample Size, Validation, Cross Validation, and Hold Out. The "Validation" column shows values like 64.0 %, 0.6072, 0.6048, etc. The "Cross Validation" column shows values like 0.6075, 0.6054, etc. The "Hold Out" column has a lock icon.

On the right side of the interface, there's a sidebar with the title "Autopilot has finished" and options: "Run Autopilot On A Different Feature List", "Unlock Holdout", "View Summary", and "Datarobot Prime".

Lift Chart

The lift chart depicts how effective a model is at predicting the target. The chart is sorted by predicted values, so you can see how well the model performs for different ranges of values of the target variable.

To view a lift chart, click a model in the Leaderboard list, then click **Lift Chart**.

The screenshot shows the DataRobot interface with the "Lift Chart" tab selected for the ENET Blender model. The "Blueprint" tab is highlighted with a red box. The "Lift Chart" section contains the following controls:

- Data Source: Cross Validation
- Number of Bins: 10 Bins
- Enable Drill Down
- Download Lift Table

The chart itself shows the relationship between "Value" (Y-axis, ranging from 0.11 to 0.70) and "Sorted Prediction" (X-axis, ranging from 1 to 10). It features two data series: "Predicted" (blue line with '+' markers) and "Actual" (orange line with circle markers). The Predicted values are generally higher than the Actual values, particularly at the lower prediction levels. The chart is titled "DataRobot Data Models Insights Studio Python Repository Untitled Project".

Below the chart, the "Model Name and Description" table lists the same models as the previous screenshot, with their respective validation and cross-validation scores.

Display Options

- Data Source -- select validation, cross-validation, or holdout.
- Number of Bins -- use this drop-down to adjust the granularity of the displayed values.
- Enable Drill Down -- click this link to compute and download cross-validation predictions.
- Download Lift Table -- download a .csv file with the lift table data.

Model X-Ray

The Model X-Ray enables you to view a chart of model details on a per-feature basis.

To view the Model X-Ray, click a model in the Leaderboard list, select **Model X-Ray**, then click **Compute data for this model**.

Autopilot has finished

Run Autopilot On A Different Feature List
Unlock Holdout
View Summary

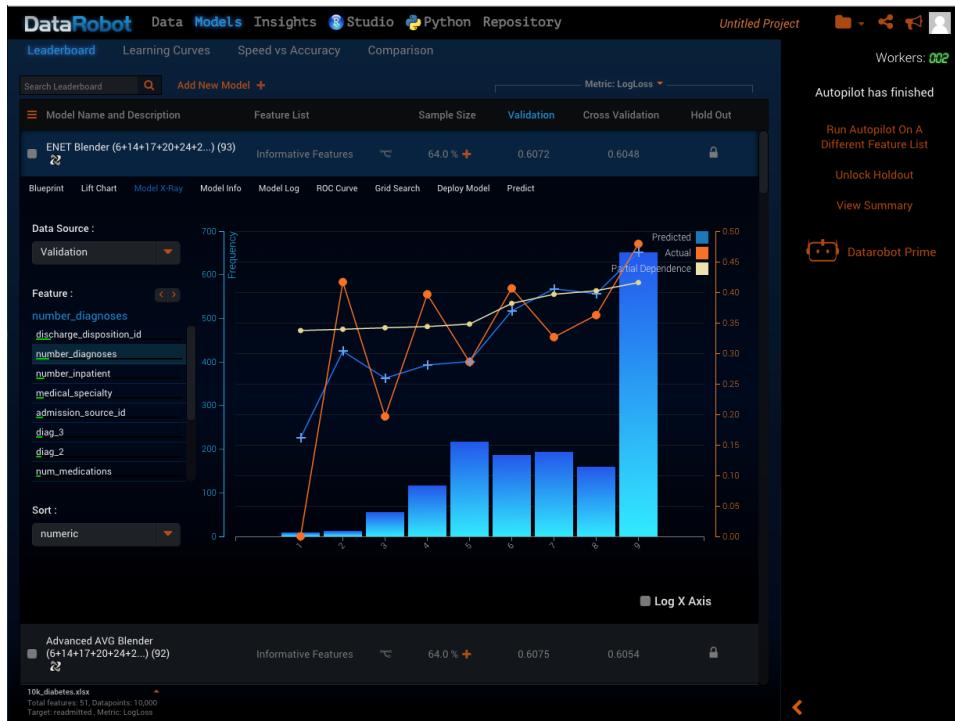
Datarobot Prime

Compute data for this model

10K_diabetes.xlsx
Total features: 51, Datapoints: 10,000
Target: readmitted, Metric: LogLoss

Model Name and Description	Feature List	Sample Size	Validation	Cross Validation	Hold Out
ENET Blender (6+14+17+20+24+2...) (93)	Informative Features	64.0 % +	0.6072	0.6048	🔒
Advanced AVG Blender (6+14+17+20+24+2...) (92)	Informative Features	64.0 % +	0.6075	0.6054	🔒
Avg Blender (20+26+27) (90)	Informative Features	64.0 % +	0.6081	0.6064	🔒
GLM Blender (20+26+27) (89)	Informative Features	64.0 % +	0.6088	0.6053	🔒
Nystroem Kernel SVM Classifier (20)	Informative Features	64.0 % +	0.6094	0.6106	🔒
Regularized Linear Model Preprocessing v4					
Advanced GLM Blender (6+14+17+20+24+2...) (91)	Informative Features	64.0 % +	0.6100	0.6049	🔒
Nystroem Kernel SVM Classifier (27)	Informative Features	64.0 % +	0.6110	0.6097	🔒
Regularized Linear Model Preprocessing v5					

Depending on the model, it may take a few minutes for all of the features to appear in the Feature list.



Display Options

- Data Source -- select a data source.
- Feature -- click a name in the Feature list to view information for that feature, or use the orange arrows to scroll through the list. Hovering over each feature displays a thumbnail image of the chart.
- Sort -- select a sorting option.
- Log X Axis -- display the logs of the x-axis values.

Model Info

To display model information, click a model in the Leaderboard list, then click **Model Info**.

The screenshot shows the DataRobot platform interface. On the left, the 'Leaderboard' section displays a list of models, with the first model selected: 'ENET Blender (6+14+17+20+24+2...) (93)'. This model has an Informative Features count of 93, a Sample Size of 64.0%, a Validation metric of 0.6072, and a Cross Validation metric of 0.6048. Below the model list are tabs for Blueprint, Lift Chart, Model X-Ray, Model Info (which is highlighted with a red box), Model Log, ROC Curve, Grid Search, Deploy Model, and Predict. A 'Resource Usage Summary' table provides details on CV Partitions, Avg. CPU Core Utilization, Max Ram, Wall Clock Time, Cache Time Savings, Cost, Cache Cost Savings, Instance Type, and Instance Size. The 'Tuning Parameters' section shows a table for 'partition' and 'NonZeroCoefficients' across five partitions. The 'Sample Size' section shows Training (6399), Test (1601), and Total (8000) observations. On the right, the 'Model Log' section is visible, showing the message 'Autopilot has finished' and options like 'Run Autopilot On A Different Feature List', 'Unlock Holdout', and 'View Summary'. A 'Datarobot Prime' logo is also present.

Resource Usage Summary

A summary of resource use that includes elapsed time, computing resources, cost, and the type of machine used to build the model (by cross-validation repetition, if applicable). Savings due to the reuse of previously computed tasks (Cache Time Savings) is also provided.

Tuning Parameters

Model optimization parameters.

Sample Size

The number of observations used to train and validate the model (for each cross-validation repetition, if applicable).

Model Log

To display the model log, click a model in the Leaderboard list, then click **Model Log**.

DataRobot Data Models Insights Studio Python Repository Untitled Project

Leaderboard Learning Curves Speed vs Accuracy Comparison

Search Leaderboard Add New Model + Metric: LogLoss ▾

Model Name and Description	Feature List	Sample Size	Validation	Cross Validation	Hold Out
ENET Blender (6+14+17+20+24+2...) (93)	Informative Features	64.0 %	0.6072	0.6048	
Blueprint Lift Chart Model X-Ray Model Info Model Log ROC Curve Grid Search Deploy Model Predict					
DataRobot Blueprint Console [06-23-2015 15:19:03] INFO: Final Cross Validation completed successfully					
Datarobot Prime					
Advanced AVG Blender (6+14+17+20+24+2...) (92)	Informative Features	64.0 %	0.6075	0.6054	
AVG Blender (20+26+27) (90)	Informative Features	64.0 %	0.6081	0.6064	
GLM Blender (20+26+27) (89)	Informative Features	64.0 %	0.6088	0.6053	
Total features: 51 Datasources: 10,000 Target: readmitted, Metric: LogLoss					

Autopilot has finished

Run Autopilot On A Different Feature List

Unlock Holdout

View Summary

ROC Curve

You can use the ROC Curve page to assess model quality. To view the ROC Curve page, click a model in the Leaderboard list, then click **ROC Curve**.

 Note: The ROC curve is displayed only for models created for a binary classification target, i.e., a target with two unique values.

The ROC Curve page contains a set of interactive graphical displays that includes the ROC Curve, a selection summary, the prediction distribution, and a confusion matrix.

DataRobot Data Models Insights Studio Python Repository Untitled Project

Workers: 002 Autopilot has finished

Run Autopilot On A Different Feature List

Unlock Holdout

View Summary

Blueprint Lift Chart Model X-Ray Model Info Model Log ROC Curve Grid Search Deploy Model Predict

Selection Summary: (BETA)

F1 Score							Actual	Predicted		968
	True Positive Rate (Sensitivity)	False Positive Rate (Fallout)	True Negative Rate (Specificity)	Positive Predictive Value (Precision)	Negative Predictive Value	Matthews Correlation Coefficient		NOT(1)	1	
0.6113	0.8136	0.5548	0.4452	0.4895	0.7851	0.5909	0.2666	431 (TN)	537 (FP)	633
								118 (FN)	515 (TP)	1601
								549	1052	

ROC Curve Data source: Validation

AUC 0.6967

Prediction Distribution Threshold (0-1): 0.3145

Density

Frequency

Informative Features 64.0 % + Cross Validation 0.6075 Hold Out

Advanced AVG Blender (6+14+17+20+24...) (92) Informative Features 64.0 % + Cross Validation 0.6054 Hold Out

10K Diabetes.xlsx Total features: 81! Datasource: 10,000 Target: readmitted, Metric: LogLoss

ROC Curve

The ROC Curve plots the true positive rate against the false positive rate. It has two important characteristics: the area under the curve, and the shape of the curve.

Area Under the ROC Curve

The Area Under the Curve (AUC) is displayed at the lower right of the ROC Curve. The goal is for the AUC to be as large as possible.

- If the area = 0.5, it means that the predictions based on this model are no better than a random model.
- If the area = 1.0, it would mean that predictions based on the model are perfect.

A perfect model is not realistic -- if it does happen, there is likely something wrong with the model (it may contain variables that depend on the response and should be excluded). Therefore the goal is to achieve an area as large as possible between 0.5 and 1.0.

ROC Curve Shape

Another informative feature of the ROC Curve is its shape. It is better when the curve grows quickly for small X-values, and slowly for values of X closer to 1. An ideal ROC curve would hug the top left corner, indicating a high true positive rate and a low false positive rate.

Display Options

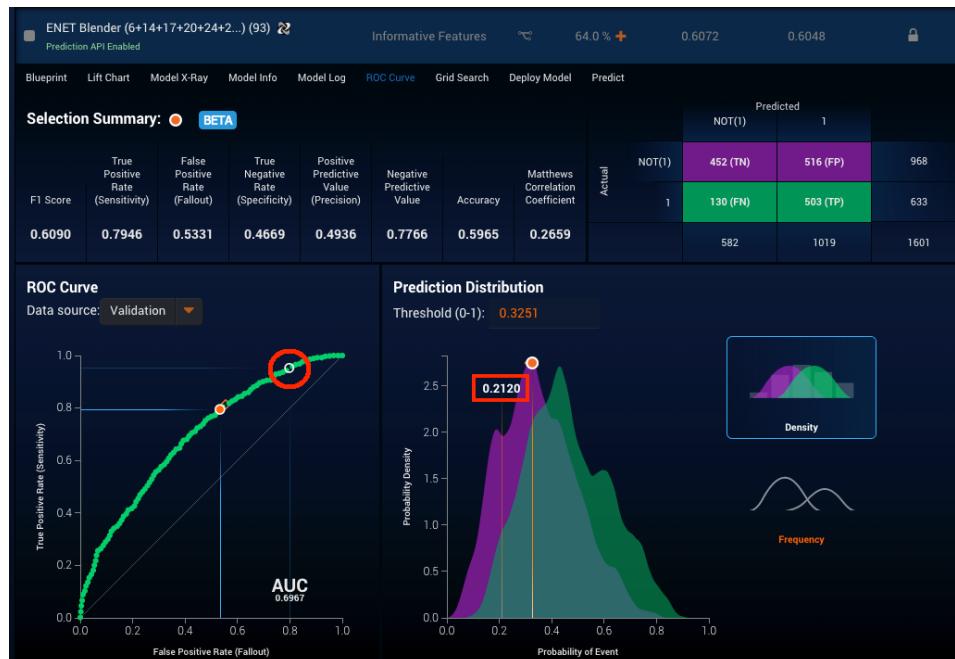
You can use the Data Source drop-down to display validation or cross-validation data.

Prediction Distribution

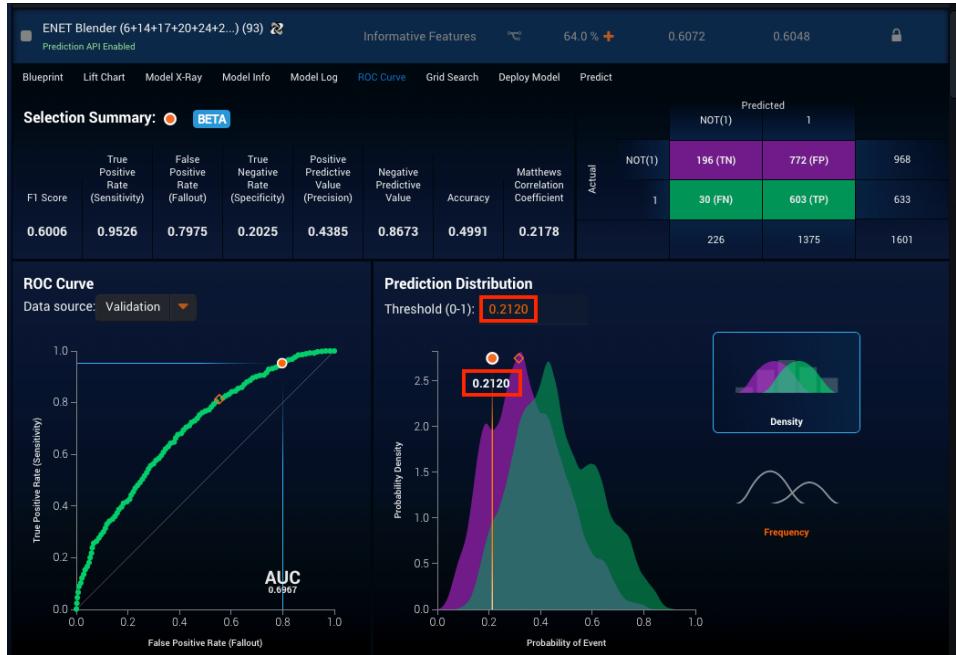
The Prediction Distribution chart expresses the performance of the model on the validation, cross-validation, or holdout dataset. You can use the Prediction Distribution chart to change the threshold value. When you change the threshold value, the ROC Curve, the Selection Summary table, and the Confusion Matrix are also updated to reflect the new value.

Use the following steps to set a new threshold value.

1. Pass the cursor over the Prediction Distribution chart. The threshold value will be displayed in white text as you move the cursor over the chart, and guidelines for the new values will also be dynamically displayed on both the Prediction Distribution chart and the ROC Curve.



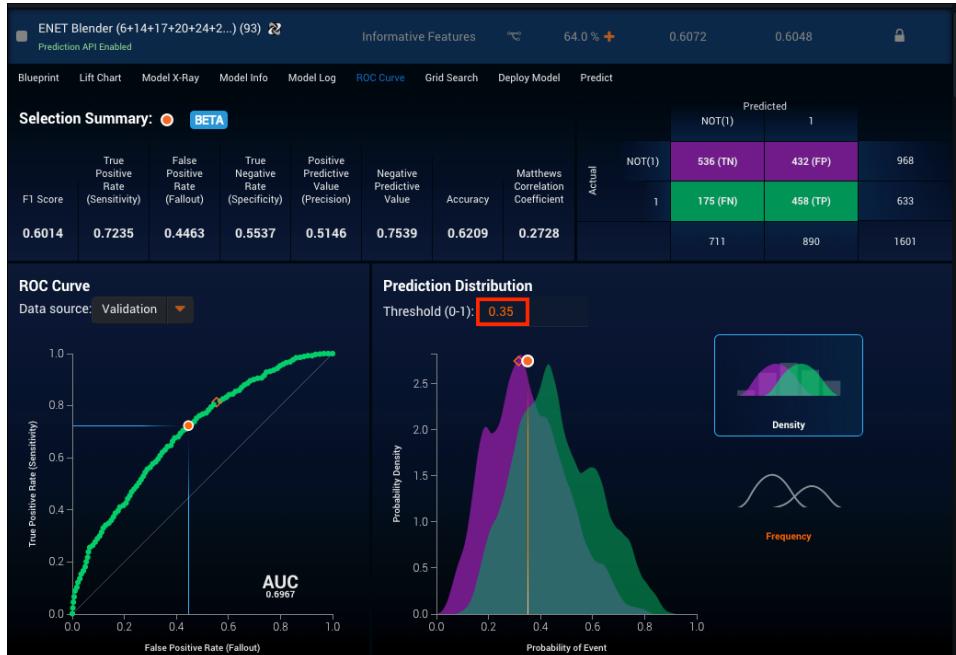
2. Click to select a new threshold value. The new value will appear in orange text in the Threshold field, and the orange dots and guidelines on the Prediction Distribution graph and ROC Curve will move to reflect the new threshold value. The information in the Selection Summary table and the Confusion Matrix will also be updated to reflect the new value.



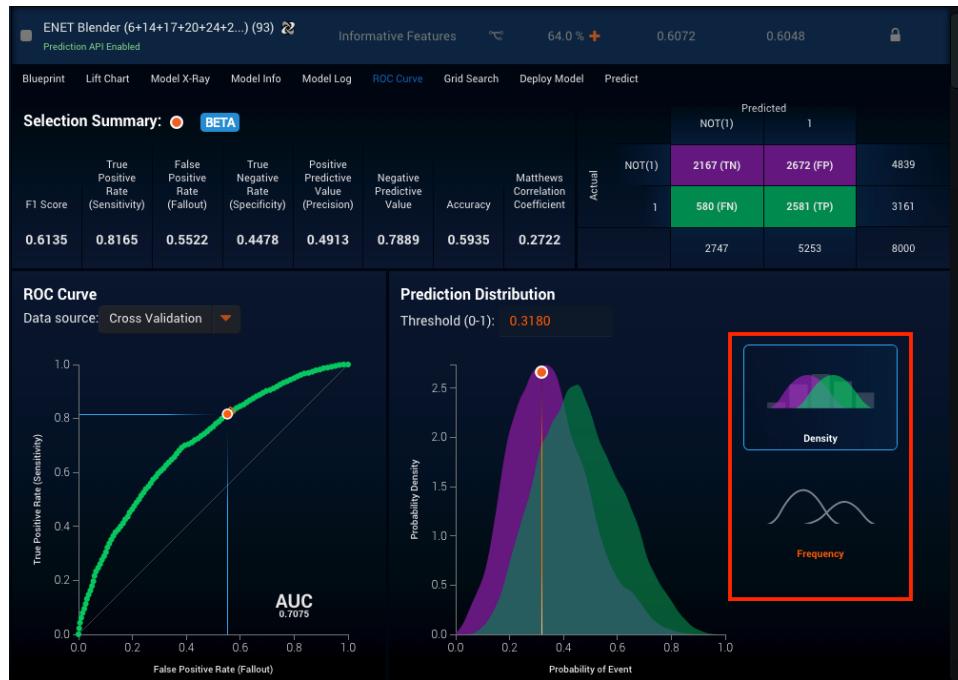
3. You can also change the threshold setting by typing a new value in the Threshold field. Click the orange Threshold text, then type in a new value.

Note: "Invalid value" will appear next to the field if the number entered is not between 0 and 1. If a non-numeric character is entered, "NaN" (Not a Number) will appear in the Threshold field.

The orange dots and guidelines on the Prediction Distribution graph and ROC Curve will move to reflect the new threshold value, and the information in the Selection Summary table and the Confusion Matrix will also be updated to reflect the new value.

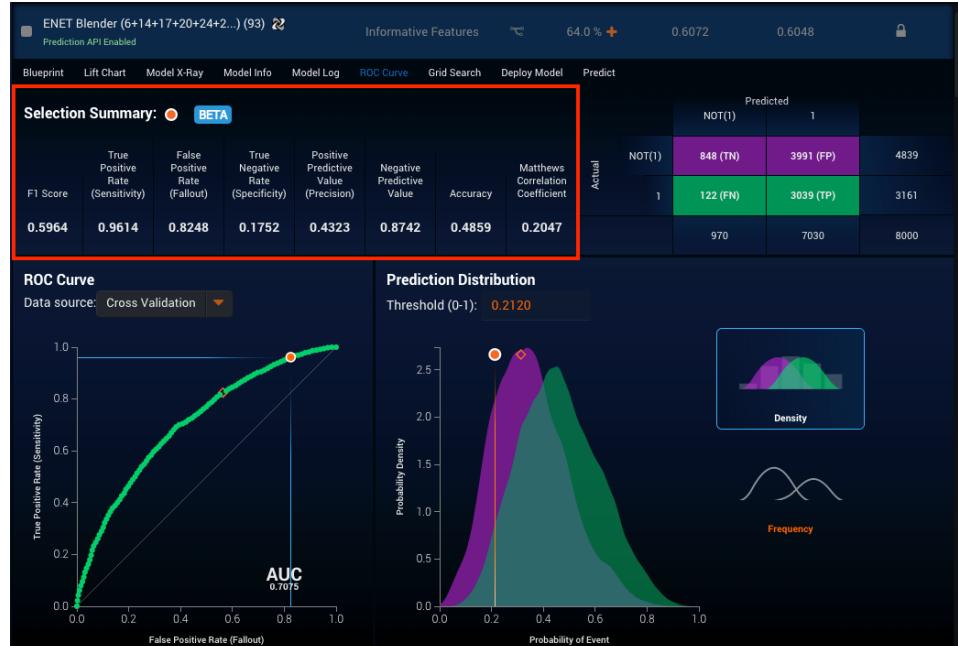


4. Click the **Density** and **Frequency** symbols at the right of the Prediction Distribution chart to display the Predictions distribution as a density or frequency curve:
- Density curve -- There is an equal area underneath both the positive and negative curves.
 - Frequency curve -- The area underneath each curve varies, and is determined by the number of observations in each class.



Selection Summary

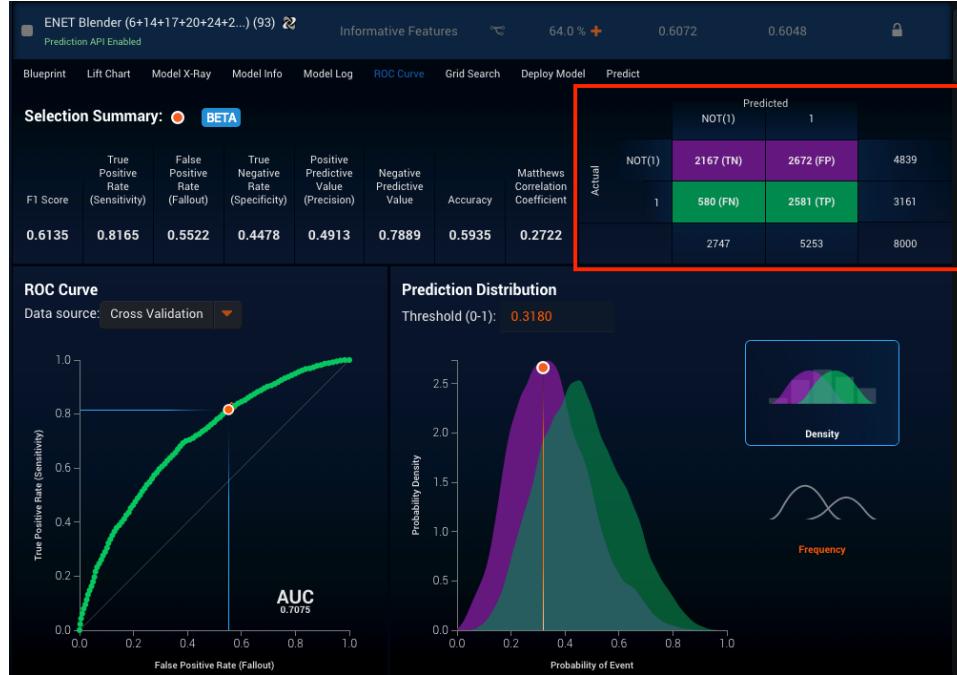
The Selection Summary table contains a number of statistics describing model performance at the selected threshold, including the F1 Score (the normalized accuracy), sensitivity, fallout, specificity, precision, negative predictive value, accuracy, and the Matthews Correlation Coefficient.



Confusion Matrix

The Confusion Matrix is a table with two rows and two columns that reports the number of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The name "confusion matrix" refers to the fact that the matrix makes it easy to see if the model is confusing two classes (consistently mislabeling one class as another class).

The information in this table facilitates more detailed analysis than relying on accuracy alone. Accuracy is not always a reliable metric for the real performance of a classifier, because it will yield misleading results if the data set is unbalanced (that is, when the number of samples in different classes varies greatly).



Grid Search

To display the model Grid Search, click a model in the Leaderboard list, then click **Grid Search**.

Model hyperparameters are tuned using internal cross-validation over the training sample, which completely omits the Leaderboard test set. If the hyperparameter space is large, a pattern search algorithm is used to find the optimum more quickly without testing every point on the grid.

You can use Grid Search to start an additional parameter grid search to help fine-tune a model or explore additional tuning parameters. To run a new grid search, type parameters in the **enet_alpha** and **enet_lambda** boxes, select **Smart Search** or **Full Search**, then click **Start**. Parameters can be entered as a comma-separated list of values.

Deploy Model

Deploy Model only applies to DataRobot Enterprise edition. If you are using the hosted service, you can click the **Request Information** link to get more information about purchasing DataRobot Enterprise edition.

If you are an Enterprise user, you can use Deploy Model to deploy the model to your hosted prediction instance.

To deploy a model, click a model in the Leaderboard list, click **Deploy Model**, then click **Activate Now**. When you deploy and activate a model:

- The model is cached to maximize speed.
- A layer of security is added that restricts access to the prediction server.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. The main area displays a list of models with their names, descriptions, feature lists, sample sizes, validation metrics, cross-validation metrics, and hold-out metrics. One model, 'DataRobot Prime', is highlighted. To the right, a sidebar titled 'Autopilot has finished' provides options to 'Run Autopilot On A Different Feature List', 'Unlock Holdout', and 'View Summary'. Below this is a section for deploying the model, with a callout box pointing to the 'Activate Now' button.

After you activate a model, you can click **Show Example** to display sample Python code for running a prediction using the model. The code sample also shows how to create an API token.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. The main area displays a list of models. A model named 'ENET Blender' is highlighted. To the right, a sidebar titled 'Autopilot has finished' provides options to 'Run Autopilot On A Different Feature List', 'Unlock Holdout', and 'View Summary'. Below this is a section titled 'Currently Deployed To:' which lists a host named 'Shared Prediction API Server' with a status of 'Ready'. Underneath, the 'API Usage : [Python]' section contains a sample Python script for making predictions using the API. The script uses the requests library to interact with the DataRobot API.

```

# Usage: python datarobot-predict.py <input-file.csv>
# Note: Before running this, change USERNAME and PASSWORD
#       to their appropriate values
USERNAME = '<USERNAME>'
PASSWORD = '<PASSWORD>'

PROJECT_ID = '5589a0ea75f6c033025hd967'
MODEL_ID = '5589ae214dd710042a51441'

# This example uses the requests library which you can install with:
# pip install requests
import requests, sys

# Gets the API token associated with your account
# Note: if you haven't created a token, change the GET to a POST to generate one.
api_token = requests.get('https://beta.datarobot.com/api/v1/api_token',
                        auth=(USERNAME, PASSWORD)).json()['api_token']

print "API token: %s" % api_token

# Make predictions on your data
# The URL has the following format: https://beta.datarobot.com/api/v1/<PROJECT_ID>/<MODEL_ID>.tds/predict
10K_diabetes.xls
Total features: 31, Datapoints: 10,000
Target: readmitted, Metric: LogLoss
    
```

Click **Hide Example** to hide the code sample. Click **Deactivate** to remove the model from the prediction server.

Predict Against a Model

Use the following steps to generate predictions on a new dataset.

- Click a model in the Leaderboard list, then click **Predict**. Use one of the following methods to select a dataset to predict with the model.
 - Drag a .csv data file to the orange box on the DataRobot platform home page.
 - Click **Browse File**, then use the file browser to select a .csv data file.
 - Type a URL in the box labeled "http://path to your file", then click **Import URL**.

The screenshot shows the DataRobot interface with the 'Leaderboard' tab selected. A large orange dashed box highlights the central area where datasets can be uploaded. Below this box are three buttons: 'Import URL', 'Browse CSV File', and 'Browse File'. At the bottom of the page, there's a section titled 'Compute & Download Model Predictions' which lists several datasets with their creation dates and a 'Compute Prediction' button next to each.

- After the file is uploaded, click **Compute Prediction** next to the file.

This screenshot is similar to the previous one, but it shows a specific dataset from the 'Compute & Download Model Predictions' section highlighted with a red box. The 'Compute Prediction' button for this dataset is also highlighted with a red box. The rest of the interface, including the leaderboard and other sections, remains the same.

- The **Compute Prediction** button changes to **In Queue** for the selected dataset, and the job status appears under Processing in the Worker queue.

The screenshot shows the DataRobot interface with the 'Models' tab selected. A list of models is displayed, including ENET Blender, AVG Blender, GLM Blender, Nystroem Kernel SVM Classifier, Advanced AVG Blender, and Advanced GLM Blender. Below the model list, there's a section for 'Compute & Download Model Predictions' with two entries: 'Informative Features' and 'Diabetes_500.xlsx'. A processing window titled 'ENET Blender...' is open, showing progress: 64.00% sample, 2%, 1.9 GB RAM, and 0.3 CPS. On the right side, there are various project management and monitoring tools.

- When the prediction has finished running, click **Download Prediction** to view the results in a .csv file.

This screenshot is similar to the previous one but focuses on the 'Compute & Download Model Predictions' section. The 'Download Prediction' button for the 'Diabetes_500.xlsx' entry is highlighted with a red box. The processing window and other interface elements are visible in the background.

- To upload and run predictions on additional datasets, click **Upload Additional Datasets**.

The screenshot shows the DataRobot interface with the following details:

- Header:** DataRobot, Data, Models, Insights, Studio, Python, Repository, Untitled Project.
- Left Sidebar:** Leaderboard, Learning Curves, Speed vs Accuracy, Comparison.
- Middle Section (Leaderboard):**
 - Search Leaderboard, Add New Model +.
 - Model Name and Description: ENET Blender (6+14+17+20+24+2...) (93).
 - Feature List: Informative Features, Sample Size: 64.0 %, Validation: 0.6072, Cross Validation: 0.6048, Hold Out: Lock icon.
 - Blueprint: Lift Chart, Model X-Ray, Model Info, Model Log, ROC Curve, Grid Search, Deploy Model, Predict.
- Right Sidebar:**
 - Workers: 002.
 - Autopilot has finished: Run Autopilot On A Different Feature List, Unlock Holdout, View Summary.
 - Datarobot Prime icon.
- Bottom Section (Compute & Download Model Predictions):**
 - Compute & Download Model Predictions: Upload Additional Datasets +.
 - Dataset: Created, Informative Features, 2015-06-25 16:50:25, Compute Prediction button.
 - Dataset: Diabetes_500.xlsx, Created: 2015-07-01 18:38:21, Download Prediction button, Lock icon.
- Bottom Left (Logs):**
 - Advanced AVG Blender (6+14+17+20+24+2...) (92), Stochastic Gradient Descent Classifier (33), AVG Blender (20+26+27) (90), GLM Blender (20+26+27) (89), Nystroem Kernel SVM Classifier (20), Advanced GLM Blender (6+14+17+20+24+2...) (91).
 - Informative Features, 64.0 %, Validation: 0.6075, Cross Validation: 0.6054, Hold Out: Lock icon.
 - Informative Features, 80.0 %, Validation: 0.6079, Cross Validation: 0.6119, Hold Out: Lock icon.
 - Informative Features, 64.0 %, Validation: 0.6081, Cross Validation: 0.6064, Hold Out: Lock icon.
 - Informative Features, 64.0 %, Validation: 0.6088, Cross Validation: 0.6053, Hold Out: Lock icon.
 - Informative Features, 64.0 %, Validation: 0.6094, Cross Validation: 0.6106, Hold Out: Lock icon.
 - Informative Features, 64.0 %, Validation: 0.6100, Cross Validation: 0.6049, Hold Out: Lock icon.
- Bottom Center (Logs):**
 - Nystroem Kernel SVM Classifier (97), 10k_diabetes.xlsx.
 - Total features: 51, Datapoints: 10,000.
 - Target: readmit, Metric: LogLoss.

Learning Curves

Learning curves demonstrate how model performance varies as the sample size changes. This is useful in determining whether it is worthwhile to increase the size of the dataset. Getting additional data can be expensive, but may be worthwhile if it increases performance.

To display the model learning curves, select **Models > Learning Curves**.

Performance versus sample size are displayed for the top-performing models. Hold the cursor over a line to display data for that model only.



Things to observe include whether there are any sharp changes, and whether models are doing worse with increased sample size. If the dataset or the validation set is small, there may be significant variation due to the exact characteristics of the datasets. It is possible for model performance to decrease with increasing sample size, as the models may become overly sensitive to particular characteristics of the training set. Note that in general, high-bias models (such as linear models) may do better at small sample sizes, while more flexible, high-variance models often perform better at large sample sizes. Preprocessing variations can also increase model flexibility.

Speed Vs. Accuracy

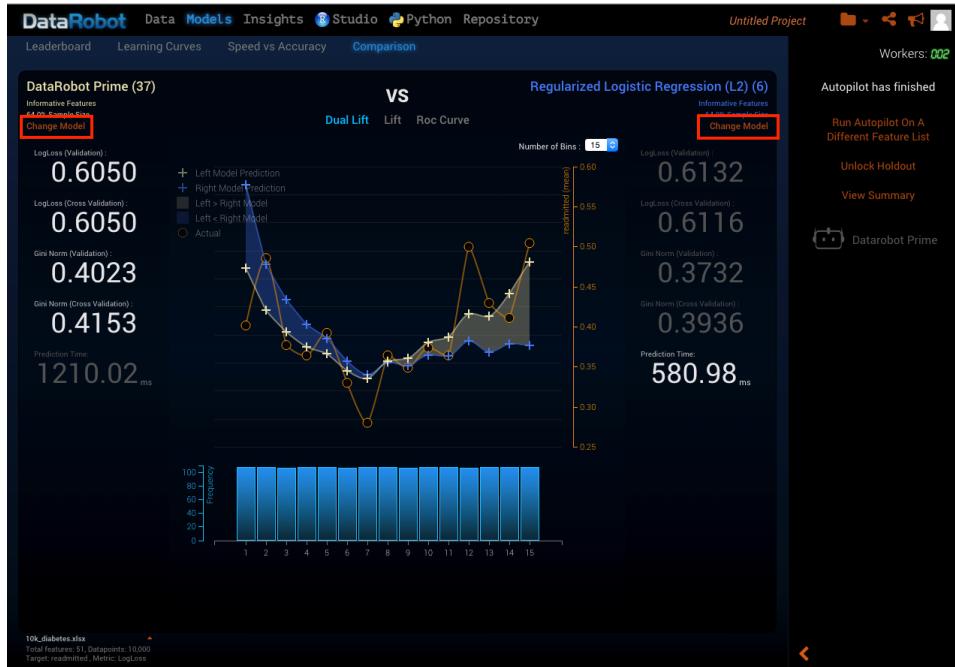
Predictive accuracy often comes with the price of increased prediction runtime. The Speed vs. Accuracy analysis plot shows the tradeoff between runtime and predictive accuracy, and helps you choose the best model with the lowest overhead.

To display this performance information, select **Models > Speed vs Accuracy**.



Comparison

To compare models, select **Models > Comparison**. You can view a plot of predicted versus actual values for the two models, along with other information, such as LogLoss and prediction runtime. To compare different models, click the left and right **Change Model** links.



Another way to access compare models is to select two models in the model Leaderboard, click the menu icon, and then click **Compare Selected**.

Insights

You can use Insights to view graphical representations of model aspects such as variable importance and variable effects. You can also use a text mining chart and word cloud to assess variable keyword relevancy, and view a hotspot chart of predictive performance.

Variable Importance

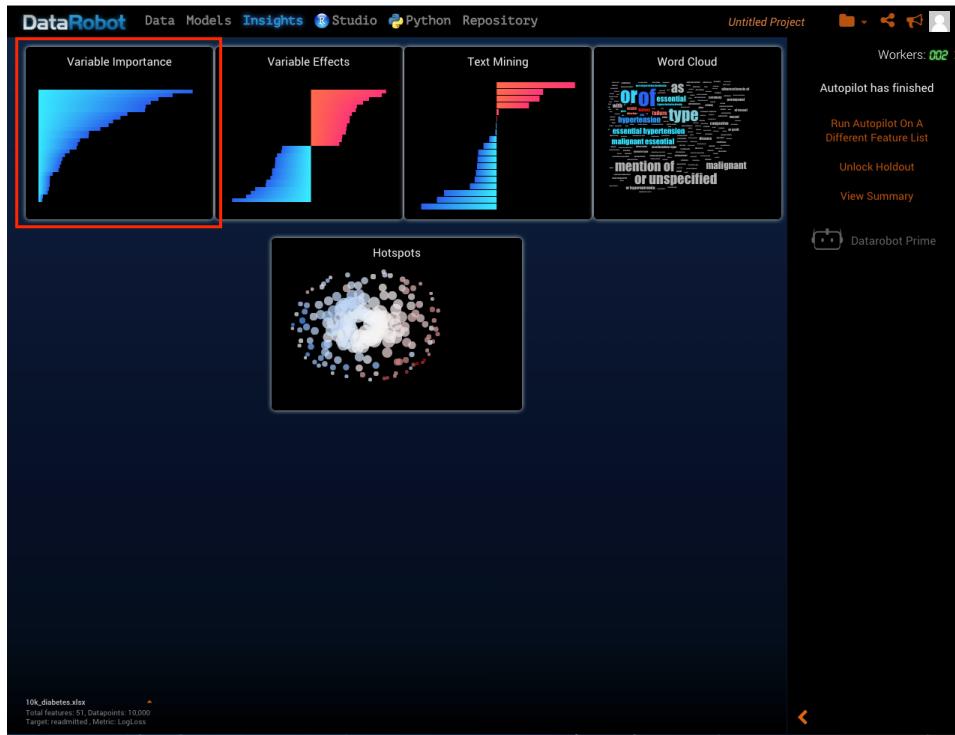
You can use the Variable Importance chart to view the sorted relative importance of all key variables driving each model in the project.

Sometimes this relative importance can be very useful, especially when a particular feature appears to be much more important for the predictions than all of the other features. It is usually worth checking to see if the values of this very important variable do not depend on the response. If this is the case, you may want to exclude this feature when training the model. Not all models have a Coefficients chart, and the Importance graph is the only way to visualize the feature impact to the model.

If a feature is included in only one model out of the dozens generated by DataRobot, it may not be particularly important, and excluding it from the feature set might help optimize model-building and future predictions.

Also, it is useful to compare how feature importance changes for the same model with different feature lists. Sometimes the features recognized as important on a reduced dataset differ substantially from those from the full feature set.

To display the Variable Importance charts, select **Insights > Variable Importance**.



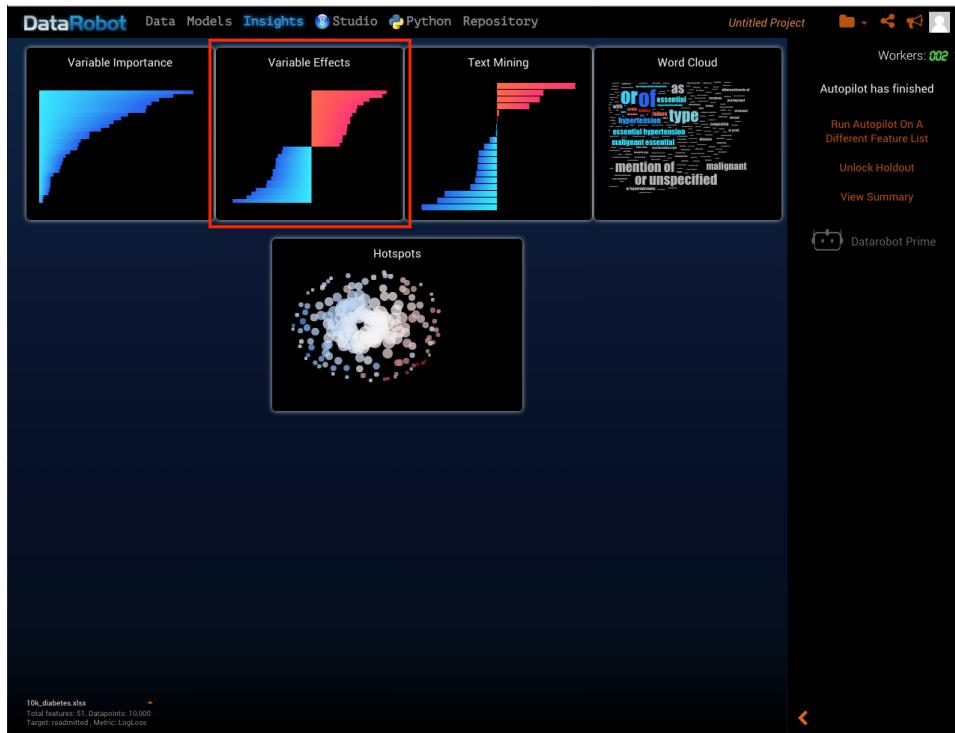
To view the Variable Importance chart for another model, click the arrow next to the model name, then select a new model.



Variable Effects

While Variable Importance shows you the relevancy of different variables for a model, the Variable Effects chart shows the impact of the top 30 variables (by magnitude) in the prediction outcomes. This is helpful when comparing the impact of a feature for different models. It is useful to ensure that the relative rank of feature importance across models does not vary wildly. If in one model a feature is regarded to be very important with positive effect, but has a negative effect in another model, there is probably something worth double-checking in both the dataset and the model.

To display the Variable Effects charts, select **Insights > Variable Effects**.



Variables with a positive effect are displayed in red, and those with a negative effect are shown in blue.



Text Mining

The Text Mining chart shows you the most relevant words and short phrases in any variables detected as text. Text variables often contain words that are highly indicative of the response.

The most important of these words and phrases are shown in the text mining chart, along with the coefficients that were built using these text strings in the model. This enables you to compare the relevant strength of the presence of these words and phrases. This side-by-side comparison is useful because individual words can be used in many different -- and sometimes counterintuitive -- ways, with many different implications for the response.

To display the Text Mining charts, select **Insights > Text Mining**.



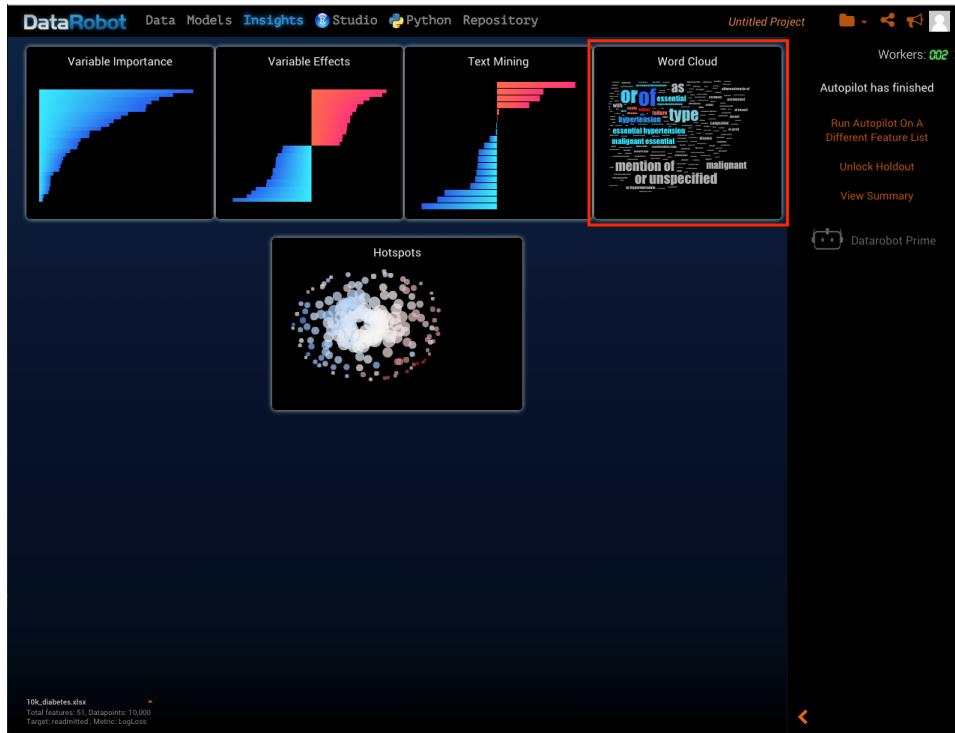
Text strings with a positive effect are displayed in red, and those with a negative effect are shown in blue. To view the Text Mining chart for another model, click the arrow next to the model name, then select a new model.



Word Cloud

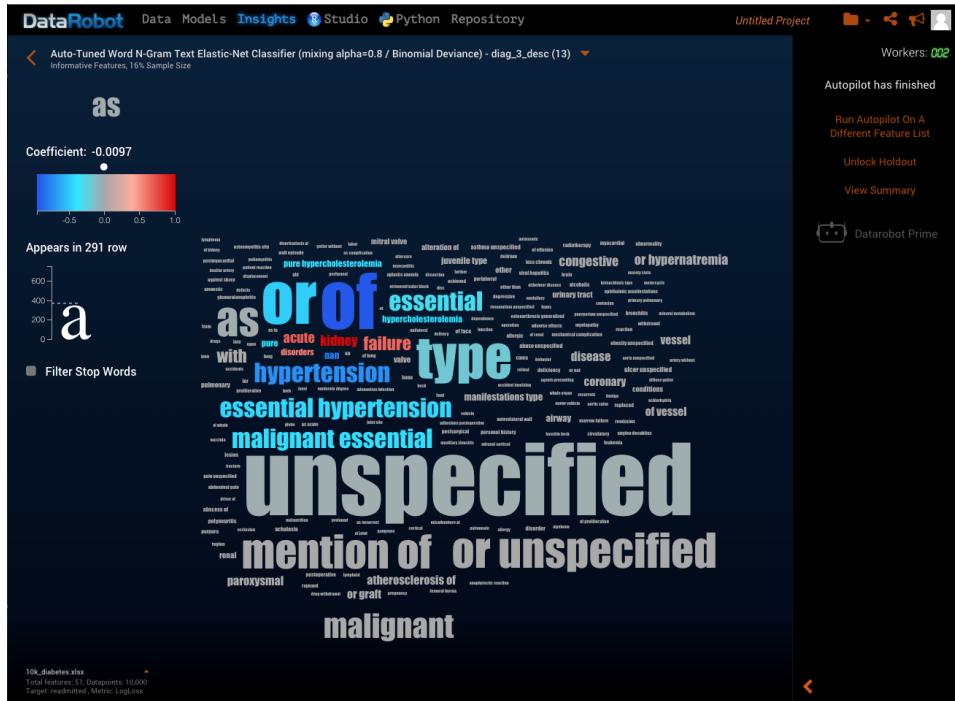
Word Cloud displays the most relevant words and short phrases in Word Cloud format. Text variables often contain words that are highly indicative of the response.

To display the Word Cloud, select **Insights > Word Cloud**.



Text strings are displayed in a color spectrum from blue to red, with blue indicating a negative effect ("cold"), and red indicating a positive effect ("hot"). Text strings that appear more frequently are displayed in a larger font size, and those that appear less frequently are displayed in smaller font sizes.

To view the Word Cloud for another model, click the arrow next to the model name, then select a new model.

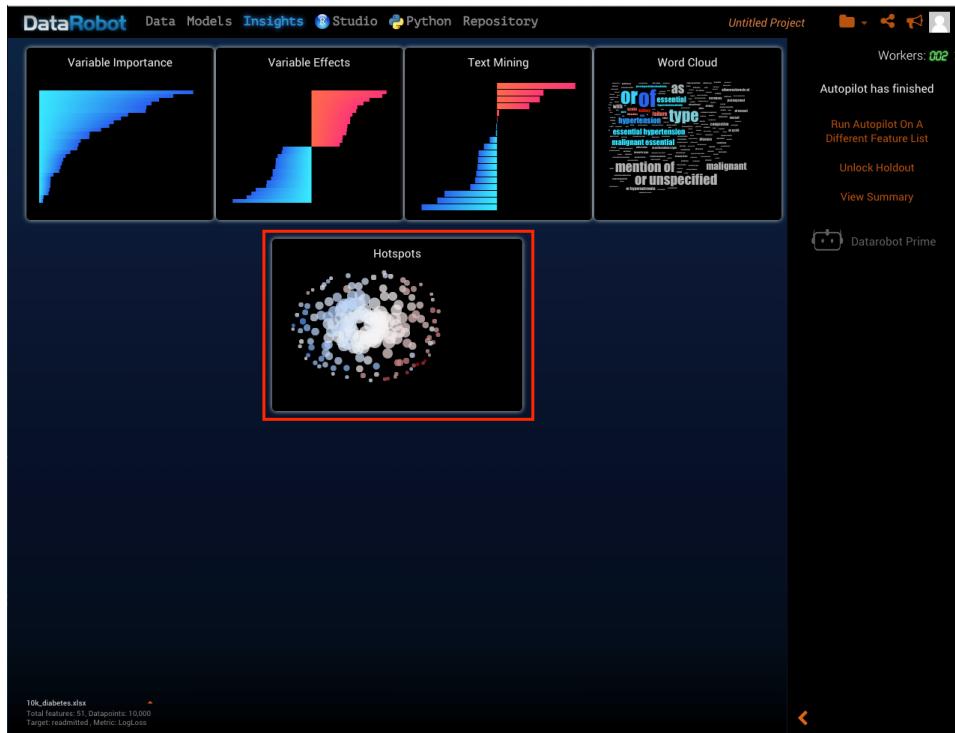


Hotspots

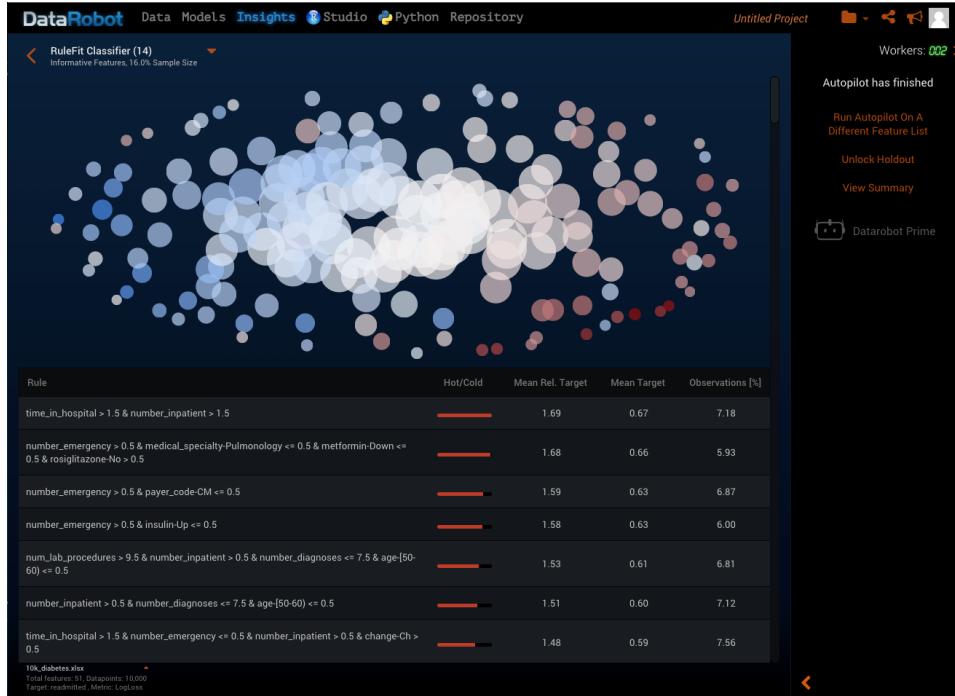
Hot and cold spots represent simple rules with high predictive performance that cover a large fraction of the data. Hotspot rules are good predictors for your data that can easily be translated and implemented as business rules.

The predictive performance of a rule is measured by the Mean Relative Response (MRR). This is the ratio between the mean response of the data points that the rule covers and the mean response of all data points. A MRR of 1 is the worst possible score. A score that deviates from 1 provides a lift with respect to a trivial regressor. Rules with a mean relative response (MRR) significantly higher than 1 are referred to as "hotspots", and rules with a MRR significantly lower than 1 are referred to as "coldspos".

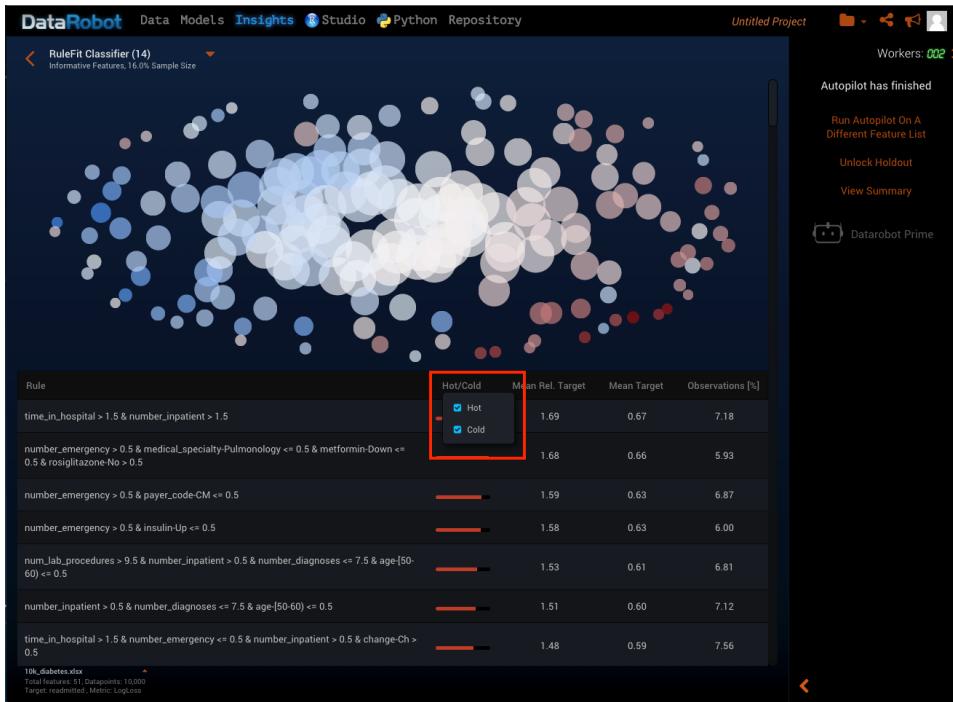
To display Hotspots, select **Insights > Hotspots**.



Hotspot rules are displayed in a color spectrum from blue to red, with blue indicating a negative effect ("cold"), and red indicating a positive effect ("hot"). Rules with a larger positive or negative effect are displayed as larger circles, and those with a smaller magnitude are displayed as smaller circles. Hotspot values are also displayed in a table.



To display only hotspots or coldspots, click **Hot/Cold**, then select or clear the **Hot** and **Cold** check boxes.

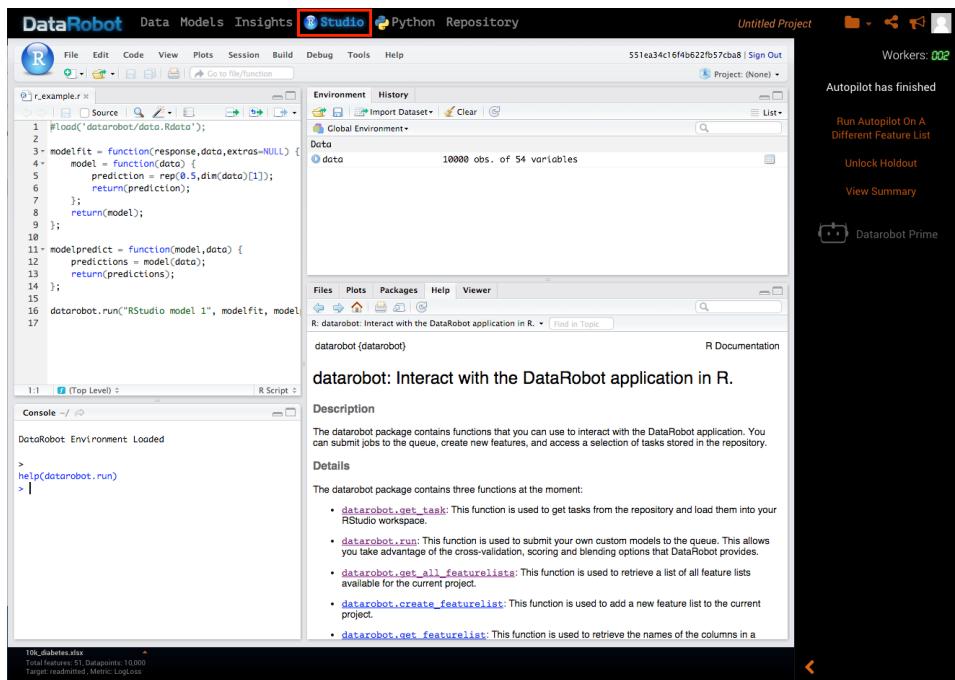


To view the Hotspots chart for another model, click the arrow next to the model name, then select a new model.

RStudio IDE

The RStudio Integrated Development Environment (IDE) enables you to run your own R models using DataRobot. After you run a model in the RStudio IDE, the results are listed on the model Leaderboard. The RStudio IDE includes documentation that describes the functions you can use to interact with DataRobot. You can submit jobs to the queue, create new features, and access models (stored as "Tasks") in the DataRobot repository.

Click **RStudio** in the top menu to access the RStudio IDE.



Python IDE

The Python Integrated Development Environment (IDE) enables you to run models written in Python using DataRobot. After you run a model in the Python IDE, the results are listed on the model Leaderboard. The Python IDE includes documentation that describes the functions you can use to interact with DataRobot. You can submit jobs to the queue, create new features and feature lists, and access models (stored as "Tasks") in the DataRobot repository.

Click **Python** in the top menu to access the RStudio IDE.

DataRobot Data Models Insights Studio **Python** Repository Untitled Project

IP[y]: Notebook Modeling Last Checkpoint: Jul 01 14:37 (autosaved)

File Edit View Insert Cell Kernel Help

Untitled Project Workers: 002 Autopilot has finished

Run Autopilot On A Different Feature List

Unlock Holdout

View Summary

Datarobot Prime

DataRobot Python

Your Data

In every notebook you open in DataRobot, the data from your project has been loaded into a variable named `data`, so you can play with your data right now. It is loaded into a [pandas DataFrame](#).

Preloaded Libraries

You will find that `numpy` and `pandas` are available as if loaded through:

```
import numpy as np
import pandas as pd
```

Running Models

Additionally, the `datarobot` module has been loaded into the environment. You can run your custom model on your data by utilizing the `datarobot.run` method. You can see the documentation for that method by executing the following in a cell:

```
datarobot.run?
```

Running this IPython Notebook will submit the `CustomModel` to be used within the DataRobot system. You may want to modify the example or write your own custom models (since this one simply predicts a 1 for every input).

You can consult the docstring for this method for a more detailed explanation.

Other Methods Available

```
datarobot.get_all_featurelists
datarobot.get_features
datarobot.get_featurelist
datarobot.create_featurelist
datarobot.create_feature
```

10L_diabetes.xlsx Total features: 81, Datapoints: 10,000 Target readmitted, Metric: LogLoss

Repository

The Python Integrated Development Environment (IDE) enables you to run models written in R and Python using DataRobot. After you run a model in the Python IDE, the results are listed on the model Leaderboard, and also stored in the DataRobot repository as "Tasks". You can use the repository to access your own custom models, as well as models shared with you by other DataRobot users.

To access the repository, click **Repository** in the top menu.

DataRobot Data Models Insights Studio Python **Repository** Untitled Project

My Tasks Shared with Me **DataRobot** Open Source Marketplace

Search Repository Batch Run

Workers: 002

Autopilot has finished

Run Autopilot On A Different Feature List

Unlock Holdout

View Summary

Databot Prime

- Auto-tuned K-Nearest Neighbors Classifier (Minkowski Distance)

Transformed text data using Converter for Text Mining. Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_1_desc. (Regulated Linear Model Preprocessing v13)

Preview Run
- Auto-tuned K-Nearest Neighbors Classifier (Minkowski Distance)

Transformed categorical data using One-Hot Encoding. (Regulated Linear Model Preprocessing v12)

Preview Run
- Auto-tuned K-Nearest Neighbors Classifier (Minkowski Distance)

Transformed text data using Converter for Text Mining. Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_1_desc. (Regulated Linear Model Preprocessing v13)

Preview Run
- Auto-tuned K-Nearest Neighbors Classifier (Minkowski Distance)

Transformed categorical data using One-Hot Encoding. (One-Hot Encoding | Missing Values Imputed | Standardize | Auto-tuned K-Nearest Neighbors Classifier (Minkowski Distance))

Preview Run
- Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_1_desc

Transformed text data using Converter for Text Mining. predicted the response using a Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_1_desc. (Converter for Text Mining | Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_1_desc)

Preview Run
- Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_2_desc

Transformed text data using Converter for Text Mining. predicted the response using a Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_2_desc. (Converter for Text Mining | Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_2_desc)

Preview Run
- Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_3_desc

Transformed text data using Converter for Text Mining. predicted the response using a Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_3_desc. (Converter for Text Mining | Auto-Tuned Word N-Gram Text Elastic-Net Classifier (mixing alpha=0.8 / Binomial Deviance) - diag_3_desc)

Preview Run
- Breiman and Cutler Random Forest Classifier

Transformed categorical data using Ordinal encoding of categorical variables. (Ordinal encoding of categorical variables | Missing Values Imputed | Breiman and Cutler Random Forest Classifier (Gini))

Preview Run
- Decision Tree Classifier (Gini)

Transformed categorical data using Ordinal encoding of categorical variables. (Ordinal encoding of categorical variables | Missing Values Imputed | Decision Tree Classifier (Gini))

Preview Run

10K_diabetes.xlsx
Total features: 81, Datapoints: 10,000
Target: readmitted, Metric: LogLoss