EBOOK

Automating **Model Risk Compliance**

Improving Model Risk Management for Financial Organizations

DataRobot

CONTENTS

INTRODUCTION

Predictive modeling is a key tool in finance. But when you make predictions, no matter what tools you use, there's a chance you'll get it wrong. Improper subprime lending, incorrect market predictions, or disastrous business decisions based on faulty financial forecasts can be costly for an organization.

This is why model risk management is an inescapable reality. No matter what modeling tools you use, they must be monitored, audited, and adjusted over time. There is no solution that includes a model to be generated and left alone. Doing so presents too many risks for both the financial institution and their customers.

To help prevent the harm that can come from improper financial model use, governments have issued regulations establishing the elements of proper Model Risk Management (MRM).

In the decade since the Federal Reserve Board (FRB) and the Office of the Comptroller of the Currency (OCC) published their guidance focused on this (SR 11-7[1] & OCC Bulletin 2011-12[2], respectively), financial institutions across the United States have invested in both processes and talent to make sure their models are compliant with these regulatory mandates.

[1] Board of Governors of the Federal Reserve System, "SR 11-7: Guidance on Model Risk Management
[2] Office of the Comptroller of the Currency, "OCC 2011-12: Sound Practices for Model Risk Management"

Since the time of that issuance, there has been an explosion of new models, new techniques, and new thinking about financial modeling, including the advancement of artificial intelligence and machine learning (AI/ML).

These tools have the potential to benefit almost every industry, but the financial sector may be where they have the greatest opportunity. With the ability to analyze large datasets, make predictive and prescriptive recommendations, and more, AI/ML is a key tool for any financial organization.

However, these new AI and ML tools have made it increasingly harder to follow the issued guidelines on model risk management. AI/ML represents the potential for creating powerful models. But an improperly managed model also represents the potential for significant risk.

To minimize risk and follow regulatory standards, you must develop and implement a model risk management (MRM) plan.

There are three critical components of managing model risk prescribed by SR 11-7 that financial institutions should follow to get the benefits of modern machine learning, while still being compliant to their model risk management framework.

- Model Development
- Model Validation
- Model Governance, Policies, and Controls

In this ebook, we'll look at each of these topics and explore what an institution can do to manage their risk and meet the guidance.

DataRobot

DEVELOPING MODELS

Making use of machine learning while still staying compliant with regulations means financial institutions must use models that are both technically correct and are appropriate to the business context.

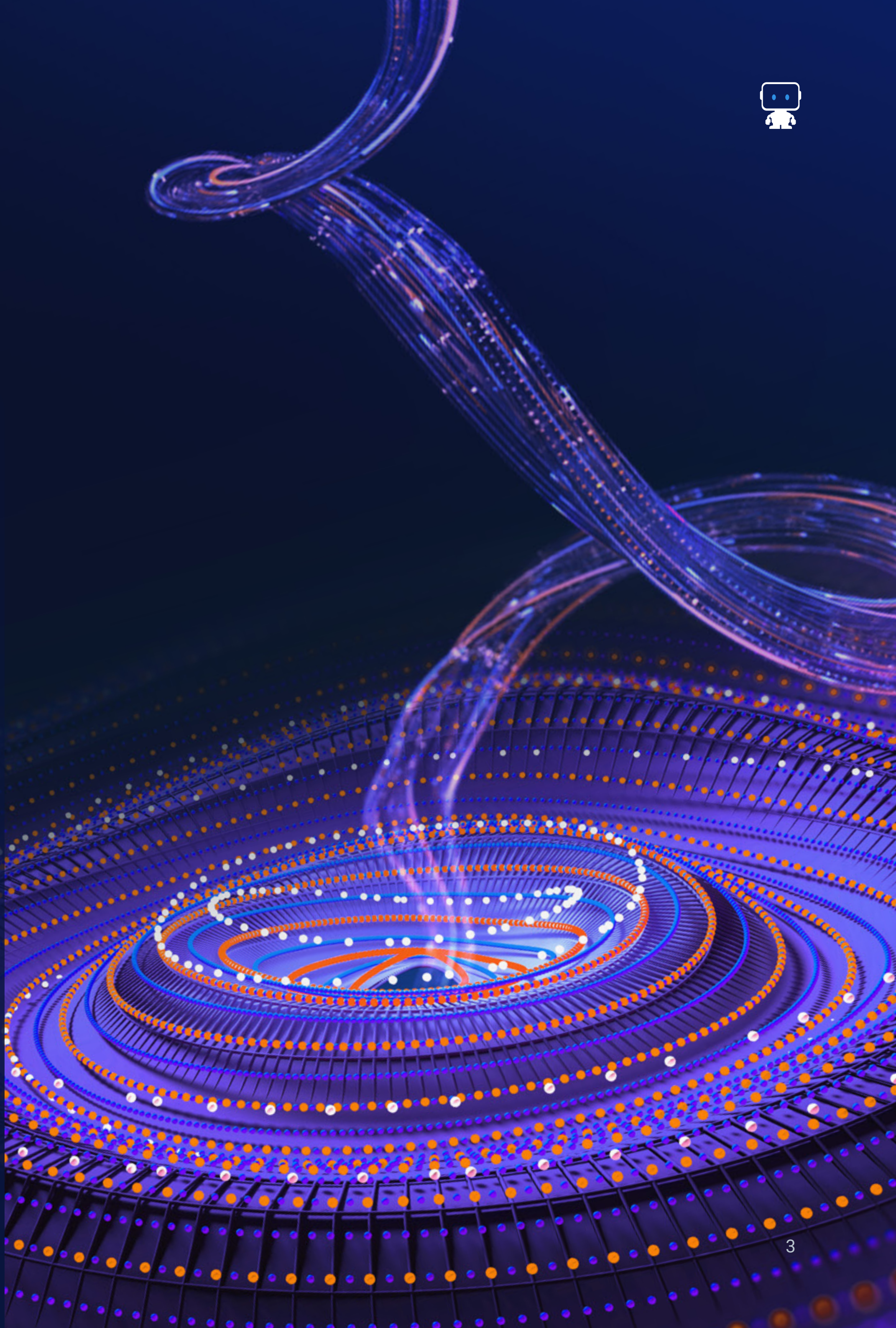
SR 11-7 describes model risk as the "adverse consequences from decisions based on incorrect or misused model outputs and reports" With this definition of model risk in mind, how do we ensure the models we build are both technically correct and compliant?

The first step is to make sure the data being used has been thoroughly vetted for its specific use.

"The data and other information used to develop a model are of critical importance; there should be rigorous assessment of data quality and relevance, and appropriate documentation."

SR 11-7

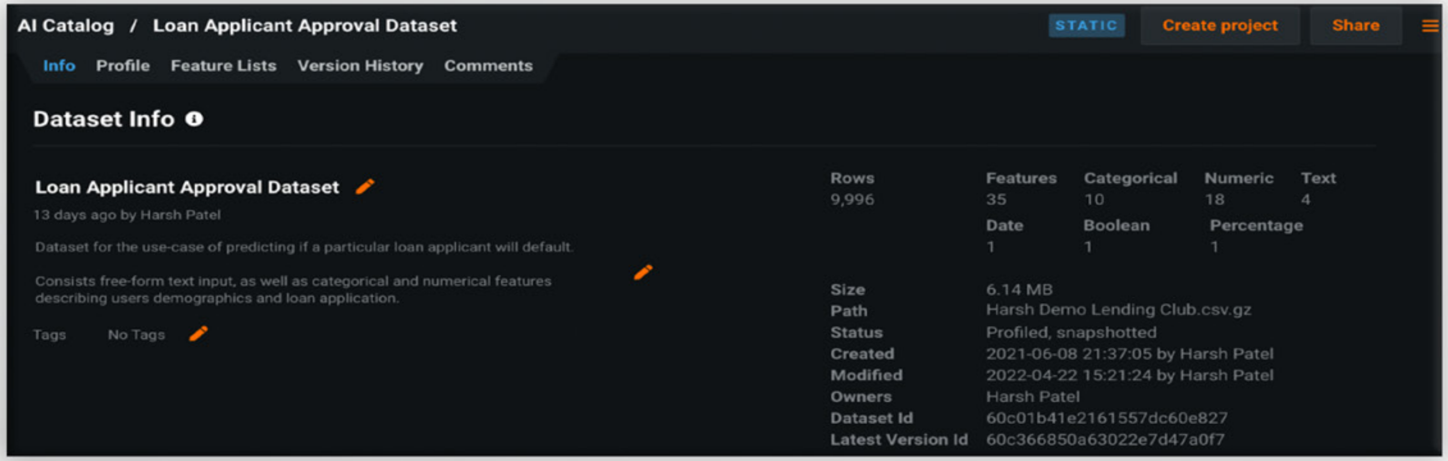
How can you be confident the data meets this threshold?



Standards for Model Development

DataRobot AI Platform is a single system of record that accelerates the delivery of AI to production for every organization. In order to be confident that your data meets the rigorous assessment set forth in the guidance, you must first ensure that the model is reproducible and validated. Using DataRobot AI Catalog, you can register the datasets you'll be using to build your model and tag them with the appropriate metadata to describe their function, origin, and intended use.

The AI Catalog helps foster reproducibility between developers and validators and sees to it no datasets are unaccounted for during the model development lifecycle.



Dataset Info

Loan Applicant Approval Dataset

13 days ago by Harsh Patel

Dataset for the use-case of predicting if a particular loan applicant will default.

Consists free-form text input, as well as categorical and numerical features describing users demographics and loan application.

Tags	No Tags

Rows	Features	Categorical	Numeric	Text
9,996	35	10	18	4
	Date	**Boolean**	**Percentage**	
	1	1	1	

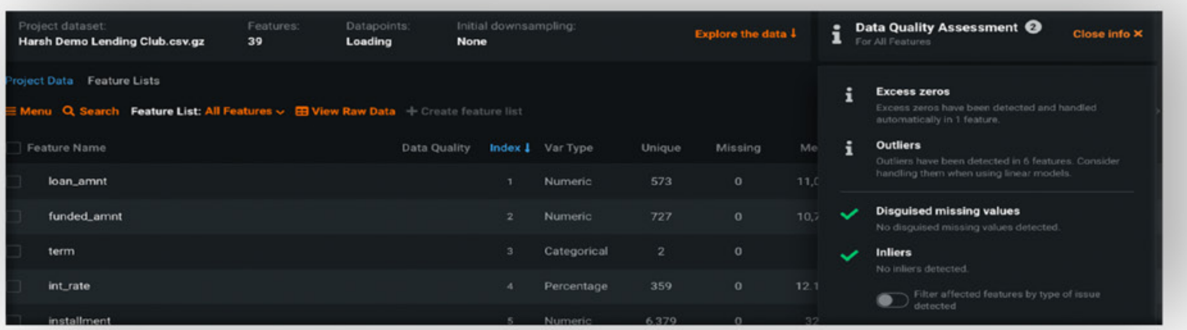
Size	6.14 MB
Path	Harsh Demo Lending Club.csv.gz
Status	Profiled, snapshotted
Created	2021-06-08 21:37:05 by Harsh Patel
Modified	2022-04-22 15:21:24 by Harsh Patel
Owners	Harsh Patel
Dataset Id	60c01b41e2161557dc60e827
Latest Version Id	60c366850a63022e7d47a0f7

Figure 1: AI Catalog within DataRobot provides critical capabilities for data management, version tracking, and profiling.

Second, you must check the data for any potential quality issues that may affect model results. At the start of a modeling project, DataRobot automatically performs a rigorous data quality assessment, which checks for common data quality issues.
These checks include:

1. Looking for redundant and non-informative data variables and removing them
2. Identifying potentially disguised missing values
3. Flagging both outliers and inliers to the user
4. Highlighting potential target leakage in variables

For a detailed description of all the data quality checks DataRobot performs, please see the Data Quality Assessment documentation.



Project dataset: Harsh Demo Lending Club.csv.gz | Features: 39 | Datapoints: Loading | Initial downsampling: None

Feature Name	Data Quality	Index	Var Type	Unique	Missing	Me
loan_amnt		1	Numeric	573	0	11,
funded_amnt		2	Numeric	727	0	10,
term		3	Categorical	2	0	
int_rate		4	Percentage	359	0	12.1
installment		5	Numeric	6,379	0	32

Data Quality Assessment For All Features

Excess zeros
Excess zeros have been detected and handled automatically in 1 feature.

Outliers
Outliers have been detected in 6 features. Consider handling them when using linear models.

✓ **Disguised missing values**
No disguised missing values detected.

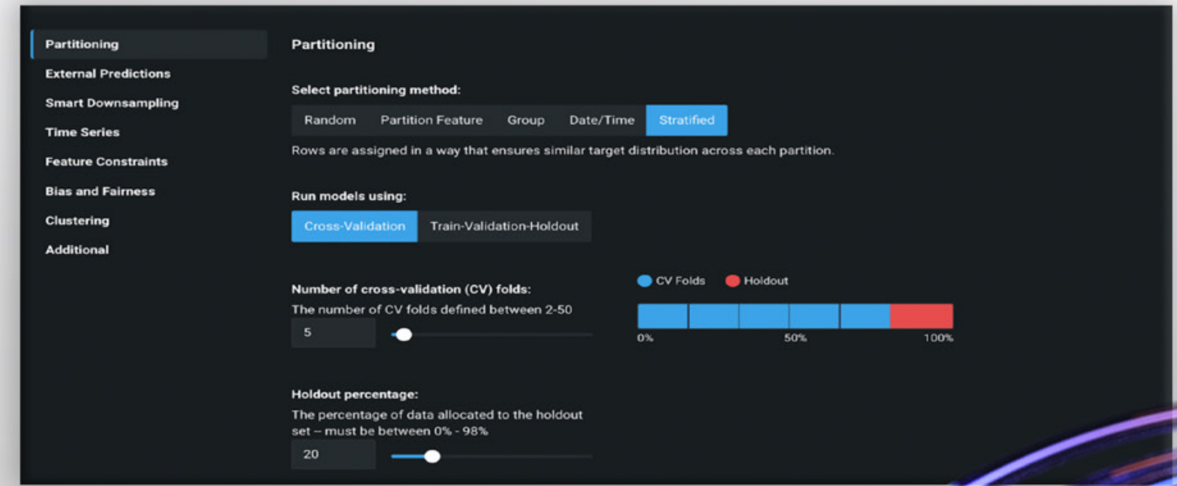
✓ **Inliers**
No inliers detected.

Filter affected features by type of issue detected

Figure 2: Output of the automated Data Quality Assessment provided through DataRobot, flagging potential data issues to the modeler.

Third, when building a model, there are several decisions that need to be made regarding partitioning, setting feature constraints, and selecting the right optimization metrics. These decisions make sure a model doesn't overfit existing data and generalizes well to new inputs.

Out of the box, DataRobot provides intelligent presets based upon the datasets and offers flexibility to further customize settings for their specific needs. The Advanced Options documentation has a detailed description of the all design methodologies provided.



Partitioning

Select partitioning method: Random | Partition Feature | Group | Date/Time | [✓ Stratified]

Rows are assigned in a way that ensures similar target distribution across each partition.

Run models using: [✓ Cross-Validation] | Train-Validation-Holdout

Number of cross-validation (CV) folds:
The number of CV folds defined between 2-50
5

Holdout percentage:
The percentage of data allocated to the holdout set – must be between 0% - 98%
20

Figure 3: Advanced Options gives you added flexibility to find the model design that fits your needs

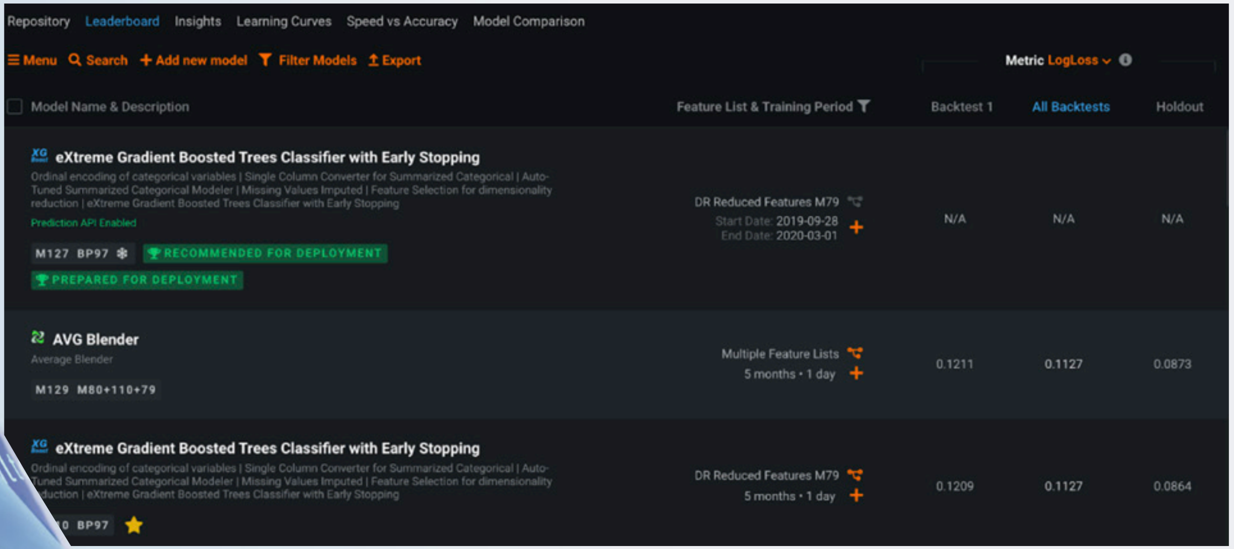
"The design, theory, and logic underlying the model should be well documented and generally supported by published research and sound industry practice."

SR 11-7

Fourth, while designing a proper model methodology is critical, it is not enough to just comply with the guidance. You may not always know what combination of data, feature preprocessing techniques, and algorithms will yield the best results for the problem at hand. While you may have a favorite modeling approach, it is not always guaranteed to be the best solution. For this purpose, it is essential to **benchmark** existing model methodologies with alternate ones for a given business objective.

> *"Comparison with alternative theories and approaches is a fundamental component of a sound modeling process."*
>
> **SR 11-7**

When kicking off a new project in DataRobot, you can automate this process and test multiple modeling approaches and compare their performance. These different approaches are captured in the Model Leaderboard, which highlights the different Blueprints and their performance against the dataset.



Model Name & Description	Feature List & Training Period	Backtest 1	All Backtests	Holdout
eXtreme Gradient Boosted Trees Classifier with Early Stopping Ordinal encoding of categorical variables | Single Column Converter for Summarized Categorical | Auto-Tuned Summarized Categorical Modeler | Missing Values Imputed | Feature Selection for dimensionality reduction | eXtreme Gradient Boosted Trees Classifier with Early Stopping. Prediction API Enabled. M127 BP97 RECOMMENDED FOR DEPLOYMENT PREPARED FOR DEPLOYMENT	DR Reduced Features M79 Start Date: 2019-09-28 End Date: 2020-03-01	N/A	N/A	N/A
AVG Blender Average Blender. M129 M80+110+79	Multiple Feature Lists 5 months • 1 day	0.1211	0.1127	0.0873
eXtreme Gradient Boosted Trees Classifier with Early Stopping Ordinal encoding of categorical variables | Single Column Converter for Summarized Categorical | Auto-Tuned Summarized Categorical Modeler | Missing Values Imputed | Feature Selection for dimensionality reduction | eXtreme Gradient Boosted Trees Classifier with Early Stopping. 0 BP97 ★	DR Reduced Features M79 5 months • 1 day	0.1209	0.1127	0.0864

Figure 4: DataRobot's Model Leaderboard displays different modeling approaches using a variety of modern machine learning algorithms.

In addition to automatically creating multiple machine learning pipelines, DataRobot provides additional flexibility through Composable ML to directly modify the blueprint. This lets you experiment and customize the model to satisfy your business needs. If you want to bring in your own code to customize specific components of the model, you can do so using Custom Tasks, which provides a way to inject your own domain expertise into the model.

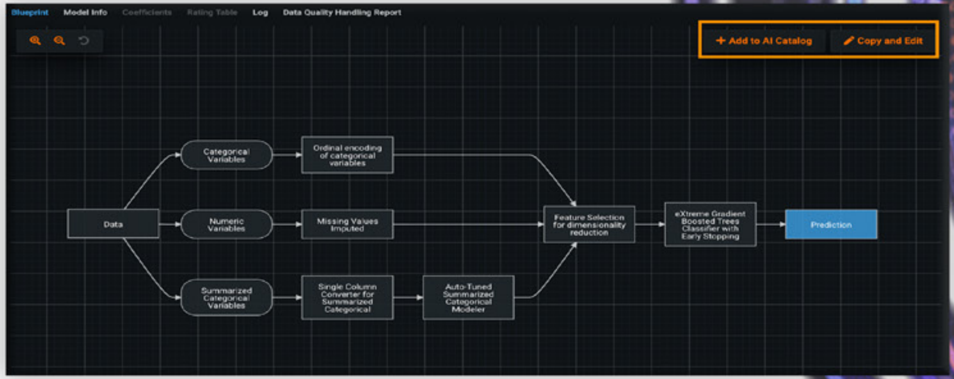
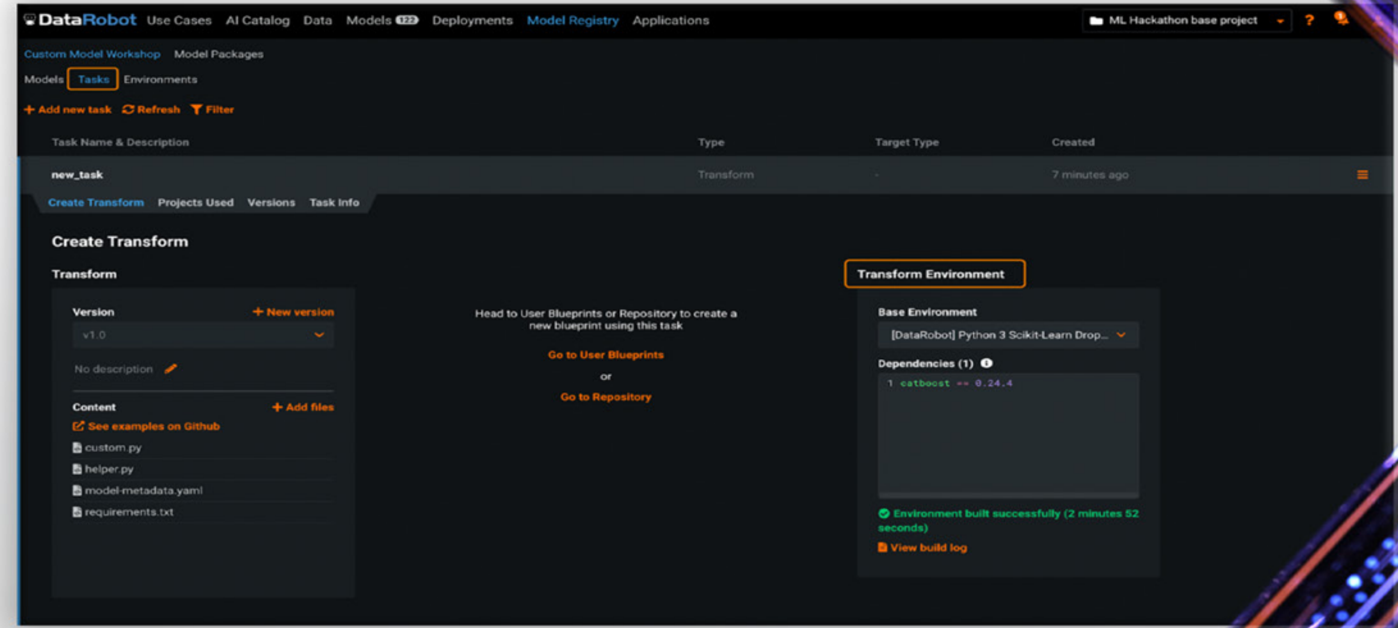

Blueprint
Model Info
Coefficients
Rating Table
Log
Data Quality Handling Report
+ Add to AI Catalog
Copy and Edit
Data
Categorical Variables
Ordinal encoding of categorical variables
Numeric Variables
Missing Values Imputed
Summarized Categorical Variables
Single Column Converter for Summarized Categorical
Auto-Tuned Summarized Categorical Modeler
Feature Selection for dimensionality reduction
eXtreme Gradient Boosted Trees Classifier with Early Stopping
Prediction

Figure 5: Customizable Blueprints let you experiment with additional feature engineering and data preprocessing techniques to evaluate competing approaches.



Task Name & Description	Type	Target Type	Created
new_task	Transform	-	7 minutes ago

Create Transform

Transform

Version: v1.0

No description

Content

custom.py
helper.py
model-metadata.yaml
requirements.txt

Head to User Blueprints or Repository to create a new blueprint using this task

Transform Environment

Base Environment: [DataRobot] Python 3 Scikit-Learn Drop...

Dependencies (1)

```
1 catboost == 0.24.4
```

Environment built successfully (2 minutes 52 seconds)

Figure 6: Custom Tasks lets you bring your own code to the blueprint, providing a means to inject your domain expertise into the model.

VALIDATING MODELS

Once you've built a model for a specific business application, how do you know it is working as expected? What are some steps you can take to evaluate the model to see if it fits the design goals?

> ***"Model validation is the set of processes and activities intended to verify that models are performing as expected, in line with their design objectives and business uses. Effective validation helps ensure that models are sound. It also identifies potential limitations and assumptions and assesses their possible impact."***
>
> **SR 11-7**

SR 11-7 details the components of an effective validation. This includes:

1. Evaluation of conceptual soundness
2. Ongoing monitoring
3. Outcome analysis

It can be hard to apply the SR 11-7 guidelines to modern ML methods. When the FRB's guidance was first introduced in 2011, modelers often employed traditional regression-based models for their business needs.

However, with the widespread adoption of modern ML techniques such as gradient-boosted decision trees (GBDTs) and deep learning algorithms, many traditional validation techniques become difficult or impossible to apply. These newer approaches often have the benefit of higher performance compared to regression-based approaches but come at the cost of added model complexity.

To deploy these models with confidence, you need to adopt new techniques to validate your model.

Conceptual Soundness of the Model

When considering machine learning models, decide if it will fit your needs. This should include reviewing your assumptions in selecting the input features and analyzing the model's behavior over a variety of input values.

Model explainability is a critical component of understanding a model's behavior and building trust in the results.

Traditional statistical models like linear and logistic regression made this process relatively straightforward. You could use your domain expertise and directly code factors relevant to the target. In the model-fitting procedure, you measure the impact of each factor against the outcome.

By contrast, many modern machine learning methods may now combine data inputs in non-linear ways to produce outputs. This makes model explainability more difficult. With that in mind, how do you ensure the data inputs and model behavior match the expectations?

One approach is to assess the importance of the input variables in the model and evaluate their impact on the outcome being predicted. Within DataRobot, each model created in the model leaderboard contains a feature impact visualization, which makes use of a mathematical technique called **permutation importance** to measure variable importance.

Permutation importance is model agnostic, making it perfect for modern machine learning approaches. It works by measuring the impact of shuffling the values of an input variable against the performance of the model. The more important a variable is, the more negatively the model performance is affected by randomizing its values.

"Where appropriate to the model, banks should employ sensitivity analysis in model development and validation to check the impact of small changes in input and parameter values on model outputs to make sure they fall within an expected range."

SR 11-7

As an example, consider a modeler who has been tasked with constructing a probability of default (PD) model. After building the model, the validator would inspect the feature impact plot to examine the most influential variables the model used.

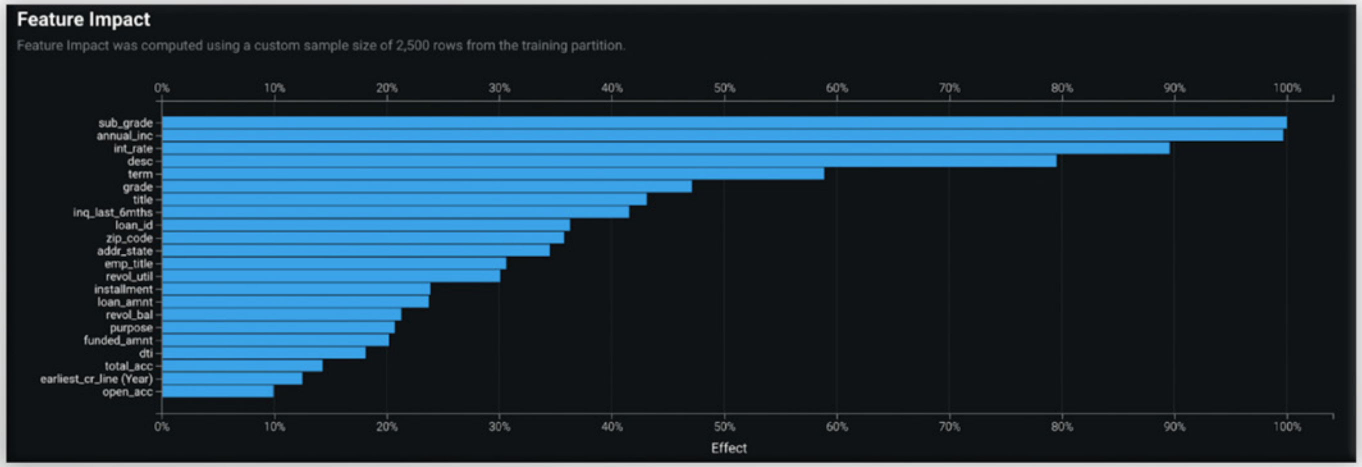


Figure 7: Feature Impact plot shows which features are driving model decisions the most

The two most influential variables in this example were the grade of the loan assigned and the annual income of the applicant. Given the context of the problem, the validator could approve the model construction, since these inputs are context appropriate.

Another step a validator may take to review the conceptual soundness of a model is to perform a sensitivity analysis.

By examining the relationship between its inputs and outputs, the validator can decide if a model is fit for its design objectives and will yield reasonable outputs across a range of input values.

DataRobot's feature effects plot makes use of a technique called partial dependence[3] to highlight how the outcome of the model changes as a function of the input variable. You can see in the figure the likelihood of an applicant defaulting on a loan decreases with an increase in their salary.

[3] Christoph Molnar, "A Guide for Making Black Box Models Explainable"

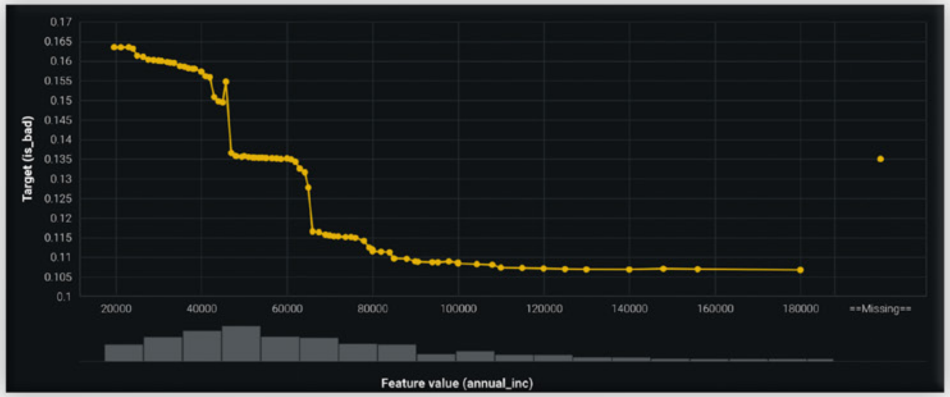
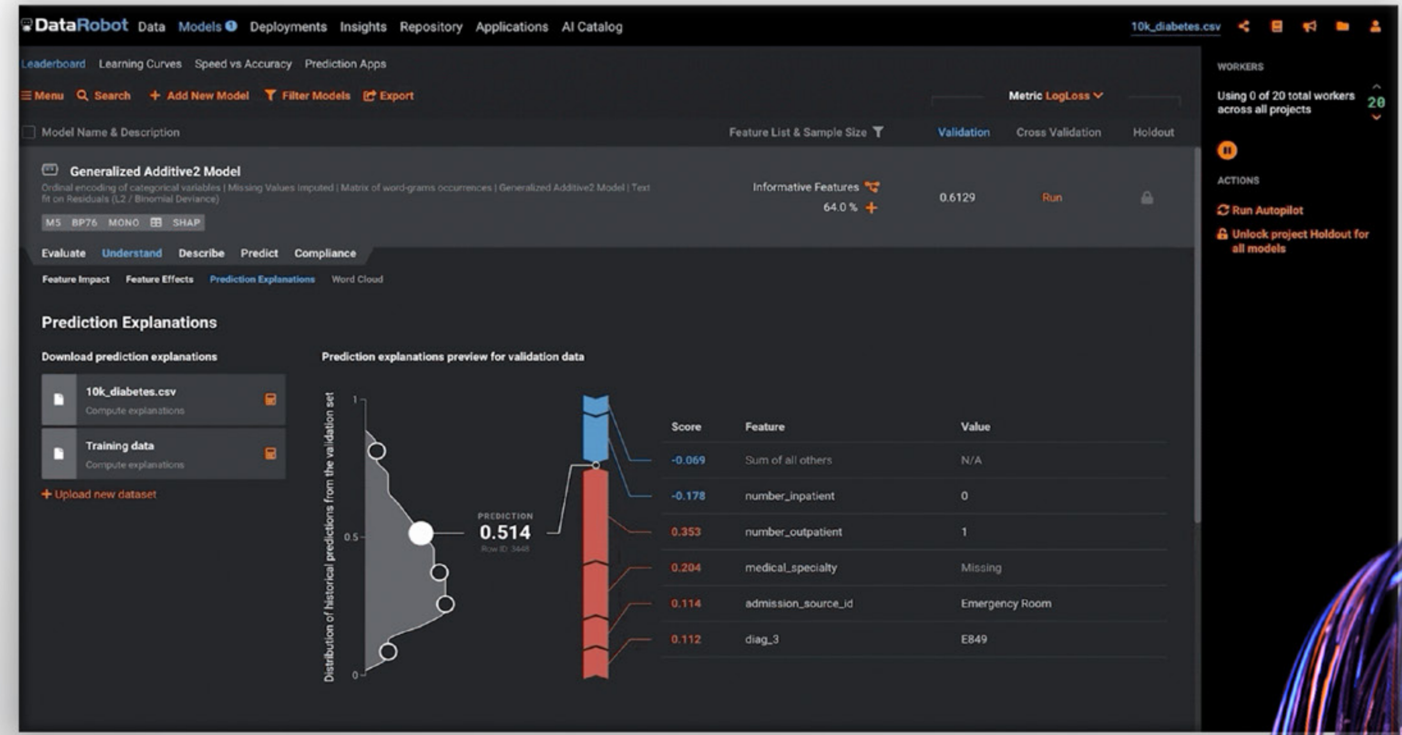


Figure 8: Feature Effect plot making use of partial dependence within DataRobot.

Another way to understand the contribution of each input variable against the model output is to make use of "local" feature explanations. Within DataRobot, you can configure the modeling project to make use of Shapley Additive Explanations (SHAP) to generate prediction explanations. This method makes sure the model adheres to domain-specific rules when making predictions. It also fosters trust with model consumers by showing them the factors driving a particular model outcome.



Generalized Additive2 Model

Ordinal encoding of categorical variables | Missing Values Imputed | Matrix of word-grams occurrences | Generalized Additive2 Model | Text fit on Residuals (L2 / Binomial Deviance)

Informative Features 64.0 % | 0.6129 | Run

Prediction Explanations

10k_diabetes.csv

Training data

PREDICTION 0.514

Score	Feature	Value
-0.069	Sum of all others	N/A
-0.178	number_inpatient	0
0.353	number_outpatient	1
0.204	medical_specialty	Missing
0.114	admission_source_id	Emergency Room
0.112	diag_3	E849

Figure 9: SHAP-based prediction explanations enabled within a DataRobot project.
These predictions quantify the relative impact of each input variable against the outcome.

DataRobot

Outcomes Analysis

Outcomes Analysis is a core component of the model validation process. It compares the model's outputs against actual outcomes. These comparisons enable you to evaluate the model's performance against your business goals. DataRobot supplies a variety of different model performance metrics based on the model architecture used. You can also perform your own analysis through its API.

In this example of a supervised binary classification problem, DataRobot automatically calculated the model's F1, Precision, and Recall scores. These are performance metrics that capture the model's ability to accurately identify classes of interest. Using the interactive interface, you can perform multiple what-if analyses to see the impact of changing the prediction threshold on the corresponding model precision and recall.

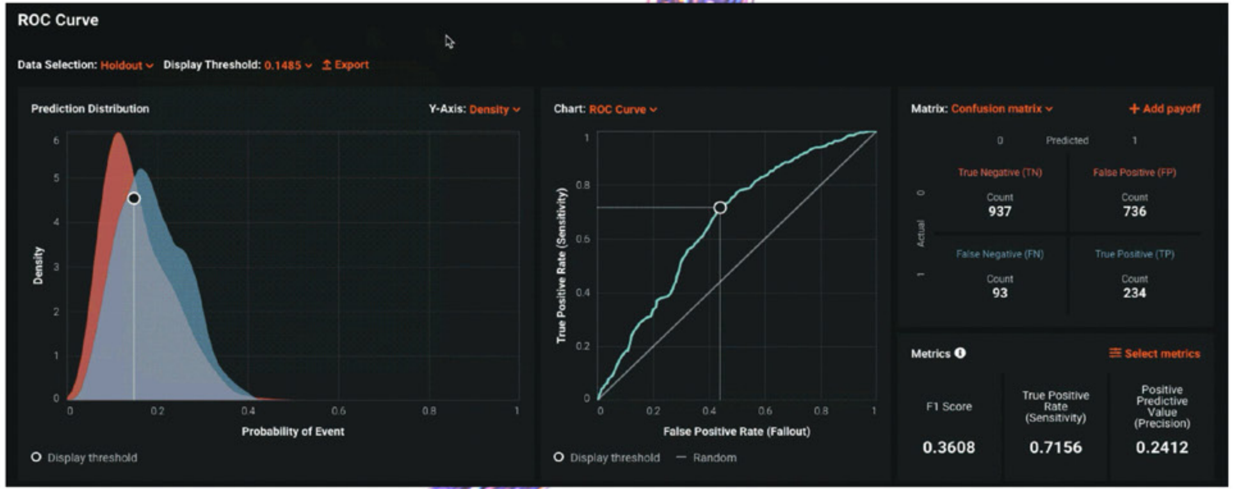
"The precise nature of the comparison depends on the objectives of a model and might include an assessment of the accuracy of estimates or forecasts, an evaluation of rank-ordering ability, or other appropriate tests."

SR 11-7

These metrics would be especially useful to financial services institutions in evaluating Anti-Money-Laundering (AML) models, where the model performance can be measured by the number of false positives it generates.

DataRobot also provides fit metrics for regression models to help you visualize the spread of model errors.

While model metrics help to quantify the model's performance, it is by no means the only way of evaluating the overall quality of the model.


ROC Curve

Data Selection: Holdout Display Threshold: 0.1485 Export

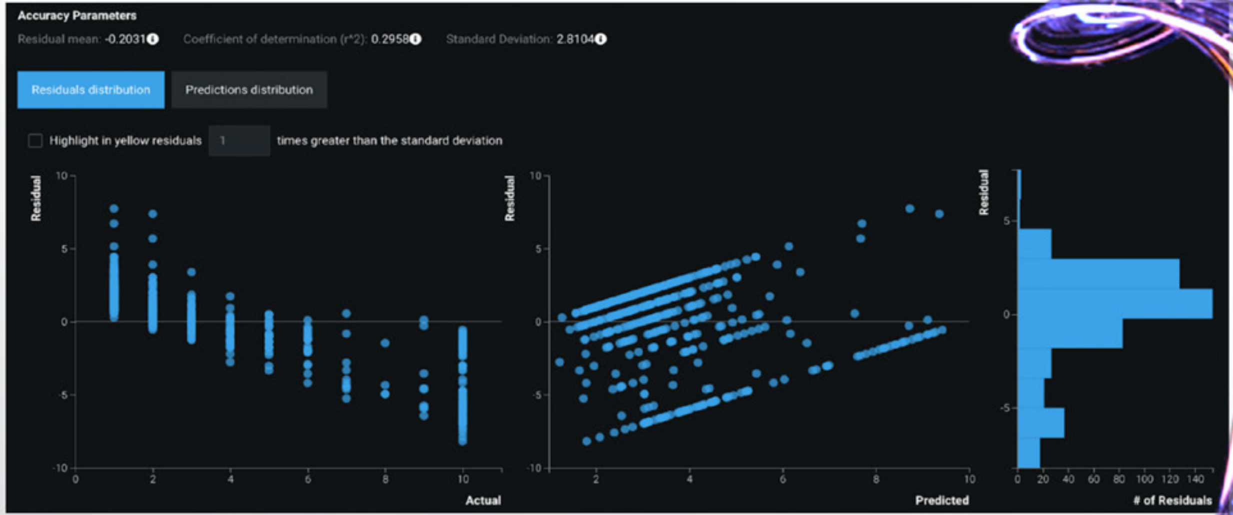
Chart: ROC Curve Y-Axis: Density Matrix: Confusion matrix

	Predicted 0	Predicted 1
Actual 0	True Negative (TN) Count 937	False Positive (FP) Count 736
Actual 1	False Negative (FN) Count 93	True Positive (TP) Count 234

Metrics

F1 Score	True Positive Rate (Sensitivity)	Positive Predictive Value (Precision)
0.3608	0.7156	0.2412

Figure 10: DataRobot provides an interactive ROC curve specifying relevant model performance metrics on the bottom right.


Accuracy Parameters

Residual mean: -0.2031 Coefficient of determination (r^2): 0.2958 Standard Deviation: 2.8104

Residuals distribution Predictions distribution

Highlight in yellow residuals 1 times greater than the standard deviation

Figure 11: Plots showcasing the distribution of errors, or model residuals, for a regression model built within DataRobot.

You could also make use of a lift chart to see if the model is well-calibrated for its objectives. For example, drawing upon the probability of default model discussed earlier in this post, a lift chart would be useful in determining if the model is able to discern between those applicants that pose the highest and least amount of credit risk for the financial institution.

In the figure shown below, the predictions made by the model are compared against observed outcomes and rank ordered based on the predicted output value.

In this case, the model looks to be well-calibrated. The outcomes align closely with the predicted values. When the model predicts an applicant is high risk, we see a higher rate of defaults (Bin 10 below). When the model predicts an applicant is low risk, we see a lower rate of defaults (Bin 1).

If the model had a flat blue line for all the ordered deciles, it would not have been fit for its business objective, since the model would have no means of discerning which applicants are at a high risk of defaulting versus those that are not.

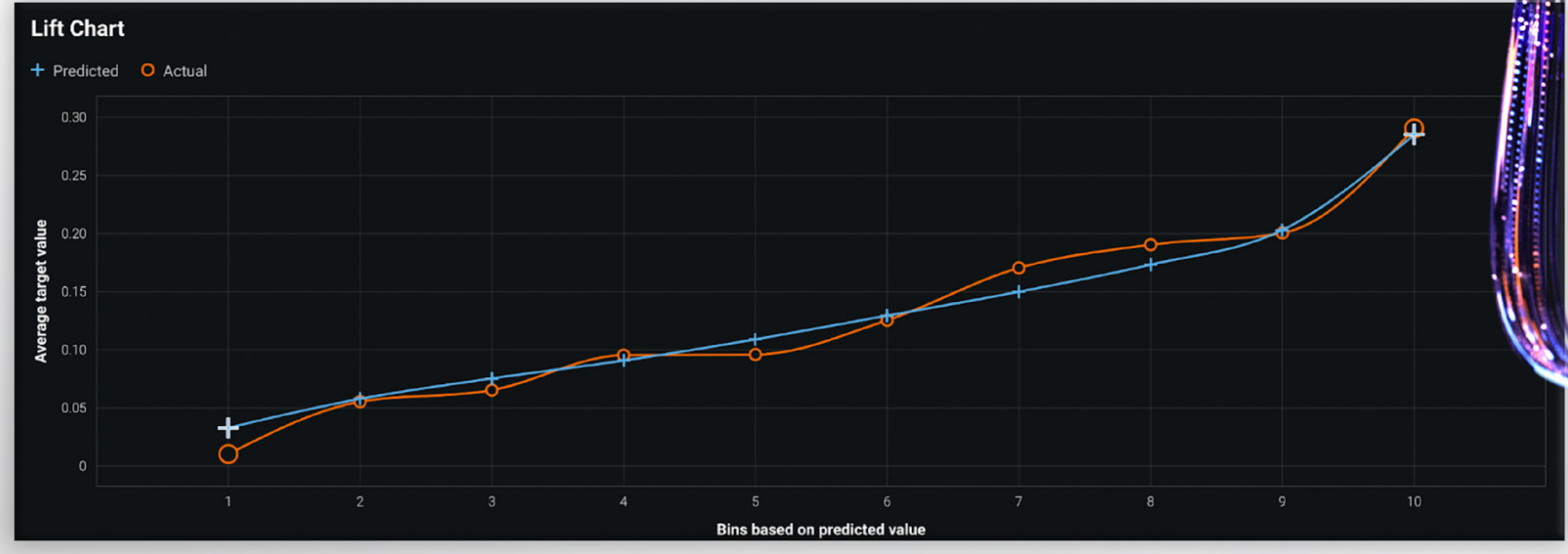


Figure 12: Model lift chart showing model predictions against actual outcomes, sorted by increasing predicted value.

MONITORING MODELS

Once a model is in production, how does a financial institution know if the model is still functioning for its intended purpose? Since models are simplified representations of reality, many of the assumptions made during development may no longer hold true. If the assumptions being used are no longer true, the model probably isn't serving its intended purpose.

"Ongoing monitoring is essential to evaluate whether changes in products, exposures, activities, clients, or market conditions necessitate adjustment, redevelopment, or replacement of the model and to verify that any extension of the model beyond its original scope is valid."

SR 11-7

Given that so many of the variables used in a model may change over time, how does a financial institution develop a robust monitoring strategy, and apply them in the context of machine learning models?

Monitoring Model Metrics

The assumptions used in designing a machine learning model may be quickly outdated due to changes in the process being modeled. This is often because the input data used to train the model was static and represented the world at only one point in time. If these changes are not monitored, the decisions made from the model's predictions may have a negative impact.

How can you identify the constantly changing conditions that may make your model useless? One way is to measure a model's performance and compare the input data to the actual business outcomes. We can measure both the data drift and model performance using these values.

Data drift measures the shift in the distribution of input values used to train the model. In the mortgage demand example from above, we could have added an input value to measure the average interest rate for different mortgage products. These would have spanned the distribution the model used to make its forecasts. If there was a shift in interest rates, we would see a change in the distribution of values.

Within the **data drift** tab of a DataRobot deployment, users can quantify the amount of shift that has occurred in the distribution, as well as visualizing it. In the image below (figure 13), we see two charts depicting the amount of drift that has occurred for a deployed model.

On the left is a chart depicting a scatter plot of the feature importance of a model input against drift. Feature importance measures the importance of an input variable from a scale of 0 to 1, making use of the permutation importance metric when the model was trained. The closer this value is to one, the more significant contribution it had on the model's performance.

Drift is displayed on the y-axis. This is measured using a metric called population stability index, which quantifies the shift in the distribution of values between a model in training and one in production.

On the right is a histogram depicting the frequency of values for a particular input feature.

It compares the data used to train the model (dark blue) against what was seen in a deployed setting (light blue). Combined with the Feature Drift plot on the left, these metrics can show us if there are any significant changes in the distribution of values in a live setting.

The accuracy of a model is another essential metric. It tells us about the health of a model in a deployed setting. Based upon the type of model deployed (classification vs. regression), there are metrics we may use to quantify how accurate the prediction is. Within DataRobot, the accuracy tab provides the owner of a model deployment with flexibility of what accuracy metrics they would like to monitor based upon their need.

In the image (fig.14) we see an example of a deployed classification model that highlights a time series of how a model's LogLoss metric has shifted over time.

Armed with a view of how data drift and accuracy has shifted in a production setting, a modeler is better equipped to understand if any of the assumptions used when training the model are no longer true. They can also quantify changes in accuracy compared to the actual business outcomes.

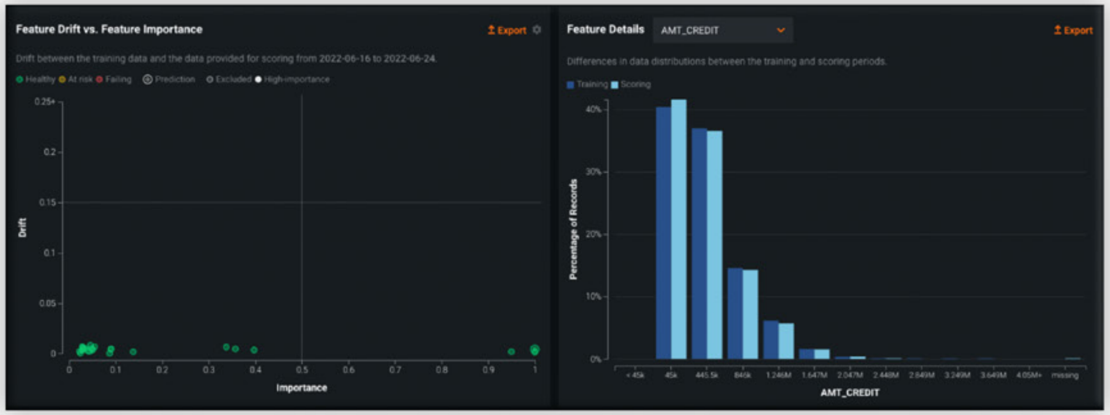
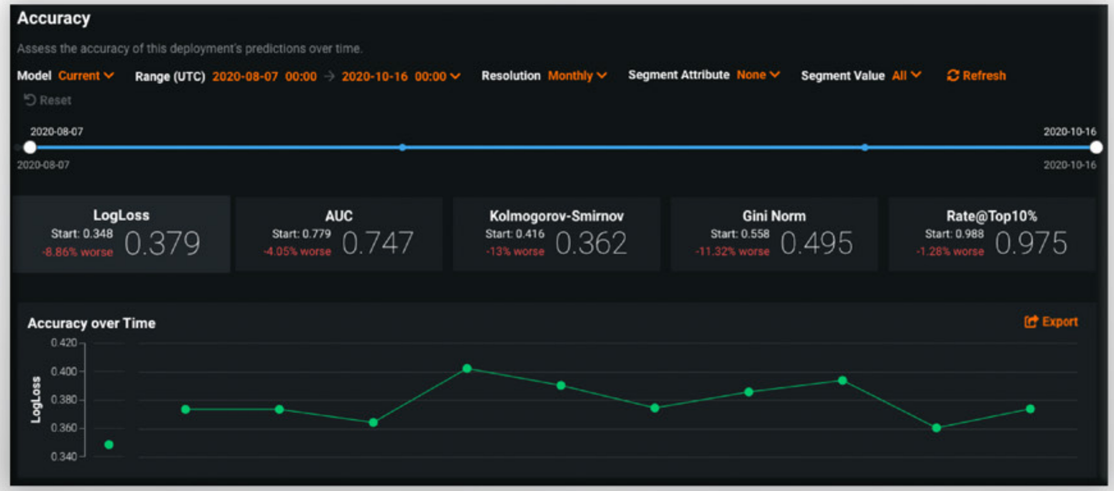

Feature Drift vs. Feature Importance
Export
Drift between the training data and the data provided for scoring from 2022-06-16 to 2022-06-24.
● Healthy ● At risk ● Failing ◎ Prediction ○ Excluded ● High-importance
Drift
Importance
Feature Details
AMT_CREDIT
Export
Differences in data distributions between the training and scoring periods.
■ Training ■ Scoring
Percentage of Records
AMT_CREDIT
< 45k
45k
445.5k
846k
1.246M
1.647M
2.047M
2.448M
2.849M
3.249M
3.649M
4.05M+
missing

Figure 13: Data drift tab of a deployed DataRobot model. Left-hand image depicts a scatter plot of Feature Drift vs. Feature Importance. The right-hand image depicts a histogram of the frequency of values observed in a live setting vs. when the model was trained.


Accuracy

Assess the accuracy of this deployment's predictions over time.

Model Current · Range (UTC) 2020-08-07 00:00 → 2020-10-16 00:00 · Resolution Monthly · Segment Attribute None · Segment Value All

LogLoss	AUC	Kolmogorov-Smirnov	Gini Norm	Rate@Top10%
Start: 0.348	Start: 0.779	Start: 0.416	Start: 0.558	Start: 0.988
-8.86% worse	-4.05% worse	-13% worse	-11.32% worse	-1.28% worse
0.379	0.747	0.362	0.495	0.975

Figure 14: Accuracy tab within a DataRobot model deployment. Model metrics here are shown for a classification problem but can be easily customized by the deployment owner.

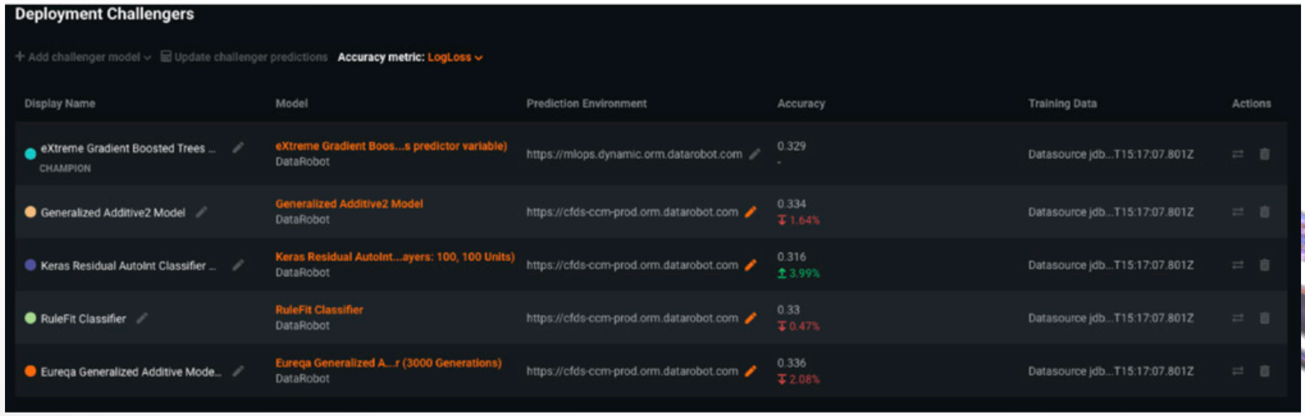
Model Benchmarking

Another fundamental principle of the modeling process in SR 11-7 is benchmarking your model against alternative models and theories. This forces you to revisit your original assumptions and try a combination of different data inputs, model architectures, and target variables.

In DataRobot, benchmarking can be accomplished during both the model building phase through the models provided in the leaderboard, but also as part of the monitoring phase through challenger models. You can compare the performance of challengers against the champion model and see if it is appropriate to swap for a better performing model.

Consider this example. An organization is trying to develop a credit risk scorecard model to determine the likelihood of default of a loan applicant.

In the initial model design, the modeler may have used experience and expertise to define the target variable of default based on the applicant repaying the loan within the first three months. When validating the model, someone in the second line of defense decides to redefine the target variable to be six months. In the image shown here, they can register their model as a challenger to the deployed champion model within DataRobot and compare its performance.



Display Name	Model	Prediction Environment	Accuracy	Training Data	Actions
eXtreme Gradient Boosted Trees ... CHAMPION	eXtreme Gradient Boos...s predictor variable) DataRobot	https://mlops.dynamic.orm.datarobot.com	0.329 -	Datasource jdb...T15:17:07.801Z	
Generalized Additive2 Model	Generalized Additive2 Model DataRobot	https://cfds-ccm-prod.orm.datarobot.com	0.334 ↓ 1.64%	Datasource jdb...T15:17:07.801Z	
Keras Residual AutoInt Classifier ...	Keras Residual AutoInt...ayers: 100, 100 Units) DataRobot	https://cfds-ccm-prod.orm.datarobot.com	0.316 ↑ 3.99%	Datasource jdb...T15:17:07.801Z	
RuleFit Classifier	RuleFit Classifier DataRobot	https://cfds-ccm-prod.orm.datarobot.com	0.33 ↓ 0.47%	Datasource jdb...T15:17:07.801Z	
Eureqa Generalized Additive Mode...	Eureqa Generalized A...r (3000 Generations) DataRobot	https://cfds-ccm-prod.orm.datarobot.com	0.336 ↓ 2.08%	Datasource jdb...T15:17:07.801Z	

Figure 15: Deployment Challengers within DataRobot. For a model deployment, modelers can select up to five challenger models for the purposes of comparing model performance.

“Benchmarking can be used… to compare a given model’s inputs and outputs to estimates from alternatives.”

SR 11-7

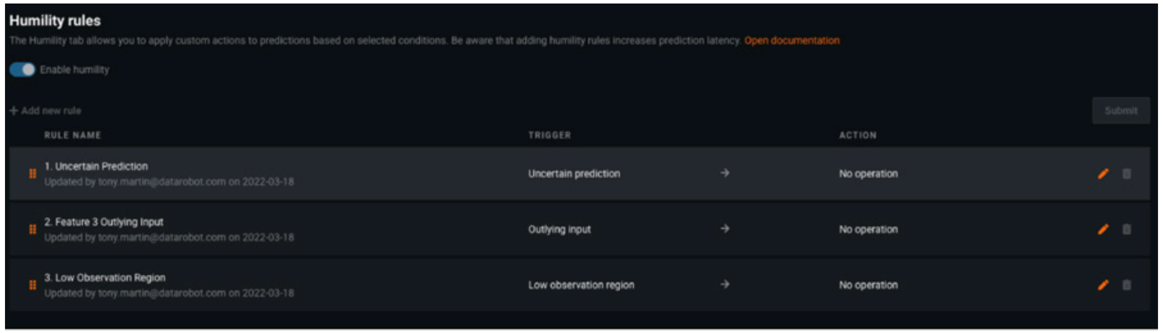
"Ongoing monitoring should include the analysis of overrides with appropriate documentation. In the use of virtually any model, there will be cases where model output is ignored, altered, or reversed based on the expert judgment from model users. Such overrides are an indication that, in some respect, the model is not performing as intended or has limitations."

SR 11-7

Overriding Model Predictions

The importance of benchmarking cannot be understated. You must constantly be evaluating the key assumptions used to design a model and iterate on a model's design and checking to be sure it still meets the intended need. However, because models are only mathematical abstractions of reality, they are still subject to limitations.

Within DataRobot, you can override rules or model overlays on both the input data and model output using [Humility Rules](). These ruleslet you acknowledge the limitations of models under certain conditions and directly codify them. The image below shows an example of a model deployment that has different Humility Rules applied.



Humility rules

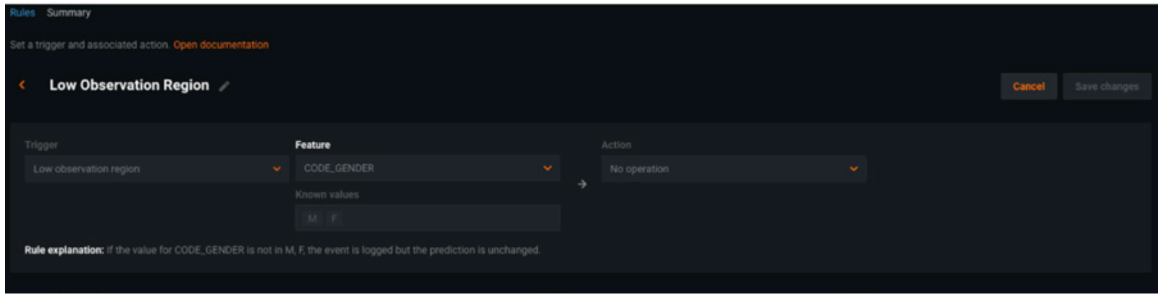
The Humility tab allows you to apply custom actions to predictions based on selected conditions. Be aware that adding humility rules increases prediction latency. Open documentation

Enable humility

+ Add new rule

Submit

RULE NAME	TRIGGER	ACTION
1. Uncertain Prediction Updated by tony.martin@datarobot.com on 2022-03-18	Uncertain prediction →	No operation
2. Feature 3 Outlying Input Updated by tony.martin@datarobot.com on 2022-03-18	Outlying input →	No operation
3. Low Observation Region Updated by tony.martin@datarobot.com on 2022-03-18	Low observation region →	No operation



Rules Summary

Set a trigger and associated action. Open documentation

Low Observation Region

Cancel Save changes

Trigger: Low observation region

Feature: CODE_GENDER

Known values: M F

Action: No operation

Rule explanation: If the value for CODE_GENDER is not in M, F, the event is logged but the prediction is unchanged.

Figure 16: Example of a humility rule configured within a model deployment. The top image illustrates the different triggers a modeler may apply, while the bottom image shows an expanded view of a configured trigger and its corresponding override action.

TAKING MODEL RISK MANAGEMENT TO THE NEXT LEVEL

Algorithmic advances in the past decade have provided AI/ML modelers with a wide range of sophisticated models to work with, but these new models have created new risks that must be managed.

Using DataRobot, modelers can build innovative models for their business applications and have access to tools for automating many of the steps as mandated in their MRM framework.

These allow data scientists to focus on the business impact of these models and deliver more value across the organization, all while being compliant to the relevant regulations.

In addition to providing unparalleled automated machine learning software, DataRobot offers Model Risk Management AI Services to customers. Our team of industry-leading data scientists and financial services professionals provide advisory services for independent model validation and assistance for regulatory compliance issues.

DataRobot and its Model Risk Management AI Services practice provides top subject matter and analytical expertise in both model development and model validation from experts with real-world experience working with banks of all sizes. We offer independent model validation services that consistently exceed client and regulatory expectations by delivering high-quality, transparent model validation analyses and reports that benefit model owners, satisfy risk managers, and stand up to regulatory scrutiny.

DataRobot

DataRobot is the leader in Value-Driven AI – a unique and collaborative approach to AI that combines our open AI platform, deep AI expertise and broad use-case implementation to improve how customers run, grow and optimize their business. The DataRobot AI Platform is the only complete AI lifecycle platform that interoperates with your existing investments in data, applications and business processes, and can be deployed on prem or in any cloud environment. DataRobot and our partners have a decade of world-class AI expertise collaborating with AI teams (data scientists, business and IT), removing common blockers and developing best practices to successfully navigate projects that result in faster time to value, increased revenue and reduced costs. DataRobot customers include 40% of the Fortune 50, 8 of top 10 US banks, 7 of the top 10 pharmaceutical companies, 7 of the top 10 telcos, 5 of top 10 global manufacturers.

Learn more at **datarobot.com**.