1 How To Use This Document

Highly regulated industries, such as banking and insurance, must comply with government regulations for model validation before a model can be put into production. This includes creating robust model development documentation. DataRobot automates the generation of model documentation, expediting the process required for regulatory compliance and following best practice for reducing model risk.

This document is split into two components: those sections that are automatically produced by DataRobot and those that require further input by the user. The sections in ***blue italicized font*** include specific instructions for the documenter and require additional user input of organization-specific information, such as business use cases, data sources, and implementation details. Once the sections are complete, remove the instructions. The remaining sections in non-blue italicized font are automatically populated by DataRobot and require no further input.

DataRobot

Table of Contents

2 Model Performance Overview

As an additional layer of model validity, DataRobot not only evaluated the statistical metrics underlying the model, but also performed testing on in-sample records.

The performance metric used for this project was LogLoss. The model performance results are presented below for in-sample testing:

Scoring Type	Score (LogLoss)
cross_validation	0.3652*
holdout	0.3039*
validation	0.3652*

3 Model Features Summary

Below are two tables. The first contains a list of the final set of model feature variables, as well as summary statistics for the LightGBM Random Forest Classifier model. The second table contains a detailed analysis of missing values.

The Model Features and Summary Statistics table provides a brief overview of the summary statistics of model features. This includes Feature Name, variable type (Var Type), number of unique values (Unique), Number of missing values (Missing), Mean, Standard Deviation (Std Dev), Median, Minimum Value (Min), Maximum Value (Max) and Assessment of target leakage risk (Target Leakage).

3.1 Model Features and Summary Statistics

Feature Name	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max	Target Leakage
water_tech_clean	Categorical	29	77839	N/A	N/A	N/A	N/A	N/A	Low
water_source_clean	Categorical	12	21944	N/A	N/A	N/A	N/A	N/A	Low
water_tech_category	Categorical	4	77839	N/A	N/A	N/A	N/A	N/A	Low
water_source_category	Categorical	6	21944	N/A	N/A	N/A	N/A	N/A	Low
management_clean	Categorical	9	71110	N/A	N/A	N/A	N/A	N/A	Low
pay_clean	Categorical	9	180039	N/A	N/A	N/A	N/A	N/A	Low
subjective_quality_clean	Categorical	5	179116	N/A	N/A	N/A	N/A	N/A	Low
age_in_years	Numeric	16654	5112	10.304	11.0009	6.63	-29.99	115.38	Low
distance_to_primary	Numeric	247831	0	19800.29	24692.39	11990.36	0.0065	249271.072	Low

distance_to_secondary	Numeric	247831	0	9803.074	12624.29	5457.99	0.016	164233.93	Low
distance_to_tertiary	Numeric	247831	0	3896.92	5590.17	1816.59	0.0096	86007.508	Low
distance_to_city	Numeric	247401	1147	51982.75	41989.75	41192.14	12.12	350229.88	Low
distance_to_town	Numeric	247831	0	17249.43	14207.83	13913.68	3.8	136251.203	Low
precipitation_5year	Numeric	2926	57514	1410.89	934.87	1210.93	160.84	12236.03	Low
precipitation_10year	Numeric	2914	57514	1426.76	935.36	1228.21	172.63	12589.39	Low
acled_index	Numeric	382	1595	24.78	75.083	3.0	0.0	4065.0	Low
bgs_recharge	Numeric	2799	17025	96.78	44.94	92.28	0.0	242.74	Low
water_risk	Numeric	693	0	2.304	0.84	2.14	-1.0	4.75	Low
rwi	Numeric	1508	0	-0.21	0.39	-0.27	-1.36	1.87	Low
assigned_population	Numeric	7804	8963	649.27	2327.94	268.0	0.0	379506.0	Low
local_population	Numeric	22000	8963	3703.034	7720.87	1345.0	0.0	379506.0	Low
pressure	Numeric	13072	14285	2.25	6.13	1.01	0.001	722.86	Low
crucialness	Numeric	157152	14179	0.35	0.32	0.23	0.0003	1.0	Low
population_1km	Numeric	171262	11085	3689.54	7955.99	1255.097	-0.0	358831.44	Low
men_ratio_1km	Numeric	60636	11085	0.49	0.67	0.49	-37.0	155.53	Low
women_of_reproductive_age_ratio_1km	Numeric	61122	11085	0.24	0.34	0.24	-41.044	105.29	Low
youth_ratio_1km	Numeric	50873	11085	0.2	0.33	0.203	-91.75	64.25	Low
elderly_ratio_1km	Numeric	57937	11085	0.053	0.076	0.049	-20.36	24.69	Low
children_under_five_ratio_1km	Numeric	54834	11085	0.16	0.16	0.17	-39.0	26.0	Low
population_10km	Numeric	120342	11085	127448.43	266026.65	65527.38	-0.0	9830356.0	Low
men_ratio_10km	Numeric	104025	11085	0.49	0.036	0.49	-5.5	5.5	Low
women_of_reproductive_age_ratio_10km	Numeric	105518	11085	0.24	0.022	0.24	-1.0	2.5	Low
population_100km	Numeric	14001	11083	5554633.48	4424416.34	4330562.0	37718.29	58837100.0	Low
youth_ratio_100km	Numeric	13920	11083	0.2	0.013	0.203	0.16	0.26	Low
orig_status_id	Categorical	2	0	N/A	N/A	N/A	N/A	N/A	Low

The last column in this table is an assessment of target leakage risk. DataRobot automatically tests for target leakage on a per-feature basis during the Autopilot process. Target leakage, sometimes called data leakage, occurs when a model is trained using a dataset that includes information that would not be available at the time of prediction. This can produce overly optimistic model performance results during training, given a feature will near-completely describe the target (e.g., the number of late payments on a loan as a predictor for loan default at loan application date.)

DataRobot tests for target leakage risk using Alternating Conditional Expectation (ACE) to measure the association between each feature and the target; the ACE score is normalized using the project optimization metric so that its value is in the range [0,1]. If above a certain threshold (see below), DataRobot will create a new feature list with those features flagged and possibly removed, and the

user is notified by a banner in the user interface during modeling. Notably, because the definition of target leakage is directly tied with prediction time and not strength of association between a feature and the target, it's possible for DataRobot to not identify all sources of target leakage. Therefore, to reduce the risk for potential target leakage in the feature list, it's important to apply subject matter expertise.

The thresholds for target leakage risk are based on a normalized ACE score:

- High risk: > 0.975, flagged and removed
- Moderate risk: > 0.85, flagged but not removed
- Low risk: < 0.85, no action

The following table provides a summary of missing values. It includes the name of the feature, its type, a summary of the missing value count (both number of rows and as a percentage), and information on the type of imputation applied to the feature.

3.2 Data Quality Handling Report

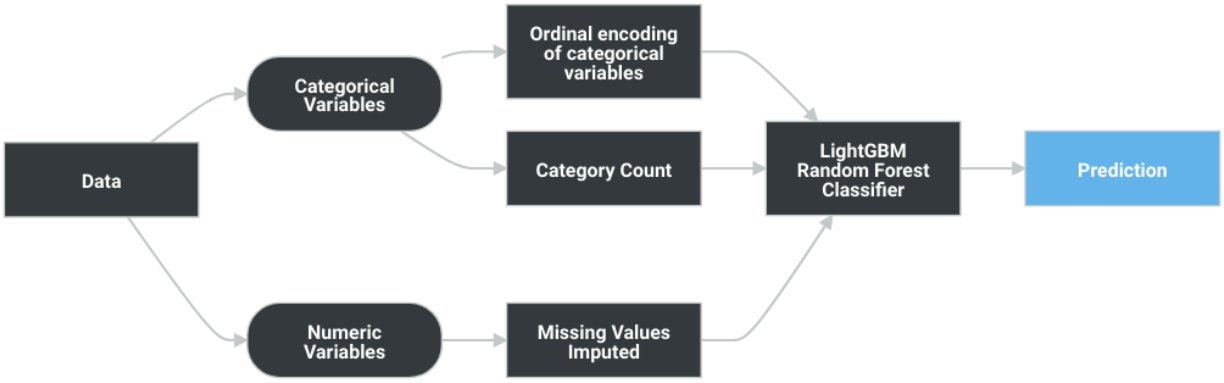
Feature Name	Var Type	Missing Count	Missing Percentage	Imputation Name	Imputation Description
subjective_quality_clean	Categorical	214044	68	Ordinal encoding of categorical variables	Imputed value: -2
pay_clean	Categorical	213754	68	Ordinal encoding of categorical variables	Imputed value: -2
water_tech_clean	Categorical	90574	29	Ordinal encoding of categorical variables	Imputed value: -2
water_tech_category	Categorical	90574	29	Ordinal encoding of categorical variables	Imputed value: -2
management_clean	Categorical	86912	27	Ordinal encoding of categorical variables	Imputed value: -2
precipitation_5year	Numeric	68024	21	Missing Values Imputed	Imputed value: -9999
precipitation_10year	Numeric	68024	21	Missing Values Imputed	Imputed value: -9999
water_source_clean	Categorical	29908	9	Ordinal encoding of categorical variables	Imputed value: -2
water_source_category	Categorical	29908	9	Ordinal encoding of categorical variables	Imputed value: -2
bgs_recharge	Numeric	20017	6	Missing Values Imputed	Imputed value: -9999
pressure	Numeric	18425	6	Missing Values Imputed	Imputed value: -9999
crucialness	Numeric	18302	6	Missing Values Imputed	Imputed value: -9999
population_1km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999
men_ratio_1km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999
women_of_reproductive_age_ratio_1km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999
youth_ratio_1km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999

elderly_ratio_1km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999
children_under_five_ratio_1km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999
population_10km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999
men_ratio_10km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999
women_of_reproductive_age_ratio_10km	Numeric	13033	4	Missing Values Imputed	Imputed value: -9999
population_100km	Numeric	13031	4	Missing Values Imputed	Imputed value: -9999
youth_ratio_100km	Numeric	13031	4	Missing Values Imputed	Imputed value: -9999
assigned_population	Numeric	12705	4	Missing Values Imputed	Imputed value: -9999
local_population	Numeric	12705	4	Missing Values Imputed	Imputed value: -9999
age_in_years	Numeric	6528	2	Missing Values Imputed	Imputed value: -9999
acled_index	Numeric	1871	1	Missing Values Imputed	Imputed value: -9999
distance_to_city	Numeric	1147	0	Missing Values Imputed	Imputed value: -9999
orig_status_id	Categorical	0	0	Ordinal encoding of categorical variables	Imputed value: -2
distance_to_primary	Numeric	0	0	Missing Values Imputed	Imputed value: 12392.117
distance_to_secondary	Numeric	0	0	Missing Values Imputed	Imputed value: 5511.5679
distance_to_tertiary	Numeric	0	0	Missing Values Imputed	Imputed value: 1787.7548
distance_to_town	Numeric	0	0	Missing Values Imputed	Imputed value: 14049.666
water_risk	Numeric	0	0	Missing Values Imputed	Imputed value: 2.1433
rwi	Numeric	0	0	Missing Values Imputed	Imputed value: -0.265

4 Model Summary and Description

The particular model referenced in this document: LightGBM Random Forest Classifier. This model was developed in a project created with vb754f77d02337f3d of DataRobot. This model is denoted within DataRobot by the Project ID: 63fbd44ff0eafeaf83c6320f and the Model ID: 640ba44a889ec21491ac5bb7. The project was created on 2023-02-26 21:51:11.

The model development workflow process (i.e., the model blueprint) is detailed in the figure below.



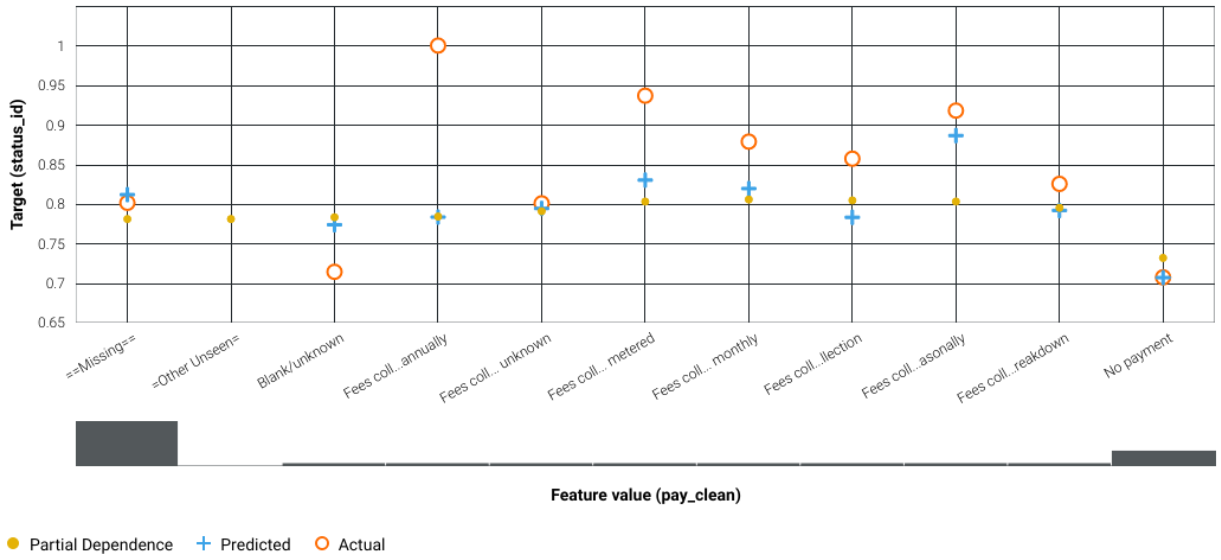
A Blueprint represents the high-level end-to-end procedure for fitting the model, including any preprocessing steps, algorithms, and post-processing. It illustrates the many steps involved in transforming input predictors and targets into a model. Each element (or, "node") in a blueprint can represent multiple steps.

The following elements connect to create the blueprint:

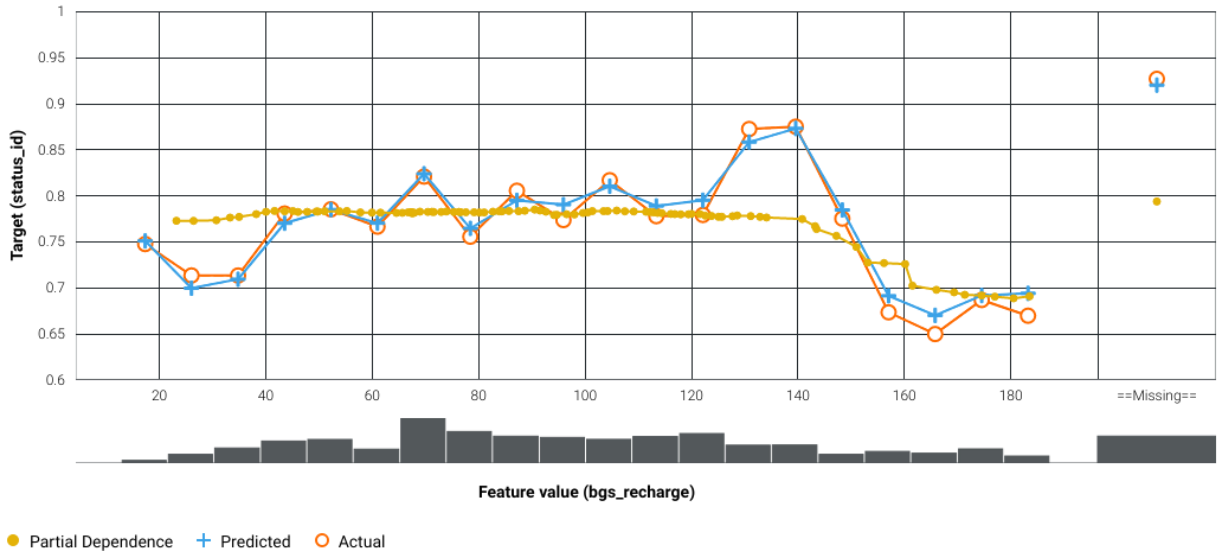
- Ordinal encoding of categorical variables
- Category Count
- Missing Values Imputed
- LightGBM Random Forest Classifier

5 Feature Effects

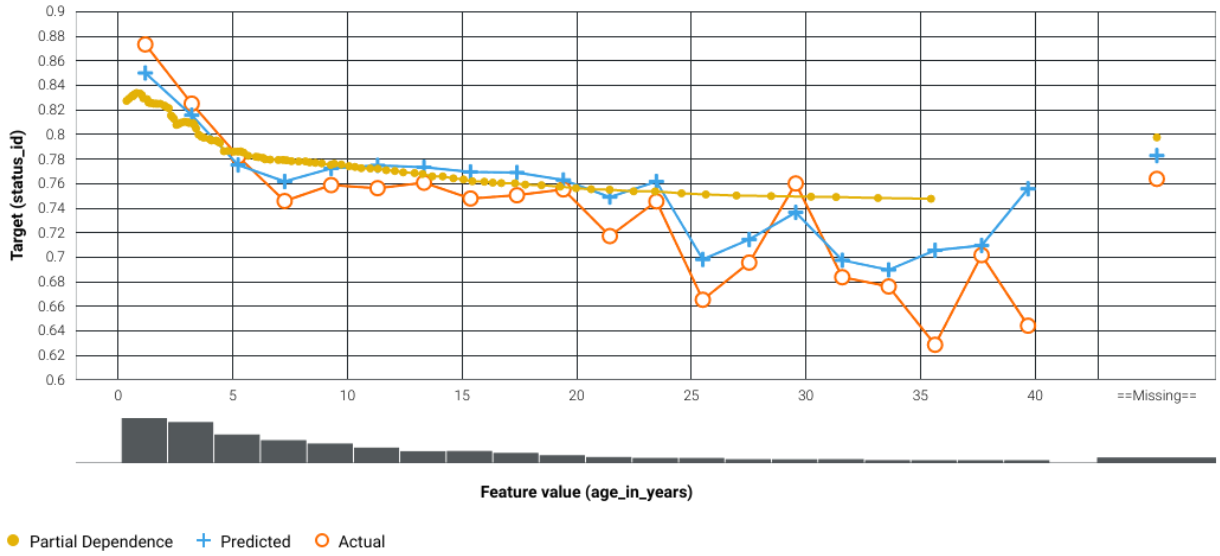
pay_clean



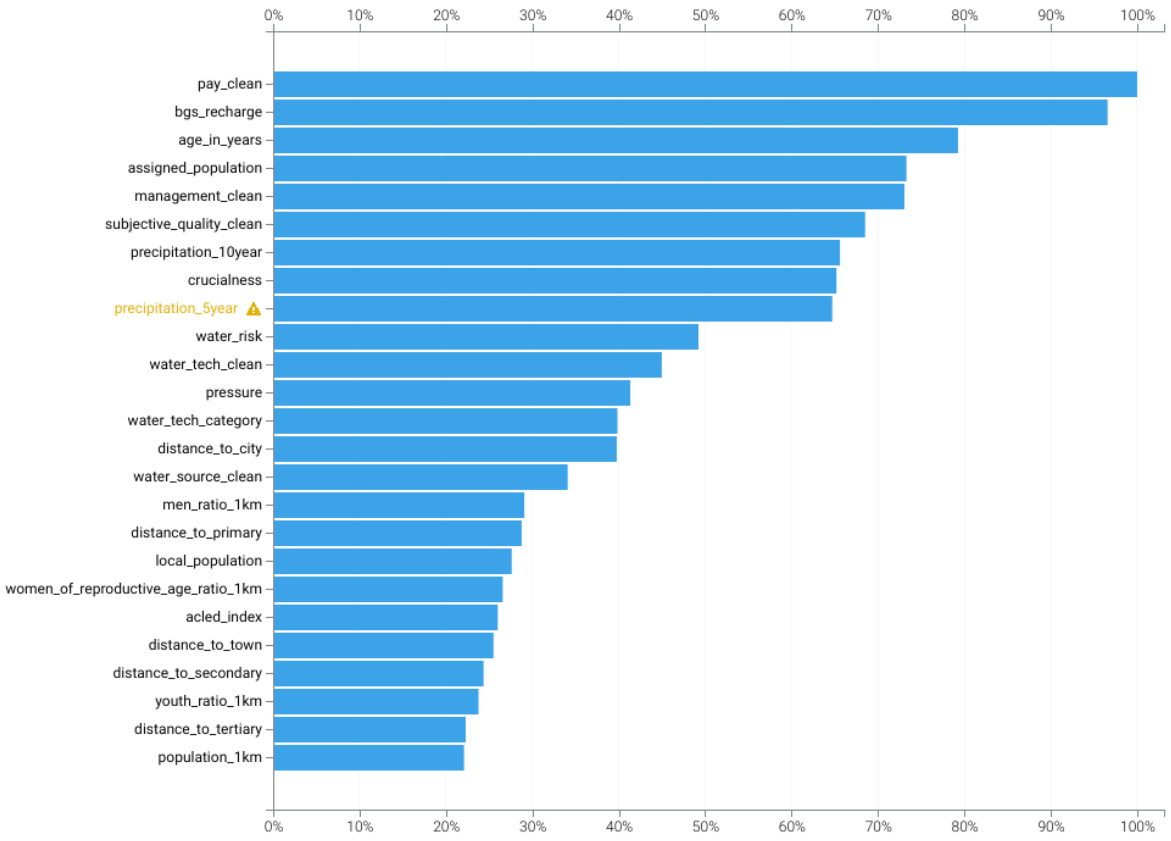
bgs_recharge



age_in_years


Target (status_id)
0.9
0.88
0.86
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0
5
10
15
20
25
30
35
40
==Missing==
Feature value (age_in_years)
Partial Dependence
Predicted
Actual

6 Feature Impact Chart



7 Feature Impact Table

Feature Name	Impact Normalized	Impact Unnormalized
pay_clean	1.0	0.0472
bgs_recharge	0.9658	0.0456
age_in_years	0.7926	0.0374
assigned_population	0.733	0.0346
management_clean	0.7306	0.0345
subjective_quality_clean	0.6852	0.0323
precipitation_10year	0.6561	0.031
crucialness	0.652	0.0308

precipitation_5year	0.6471	0.0306
water_risk	0.4923	0.0232
water_tech_clean	0.4498	0.0212
pressure	0.4133	0.0195
water_tech_category	0.3985	0.0188
distance_to_city	0.3977	0.0188
water_source_clean	0.3409	0.0161
men_ratio_1km	0.2907	0.0137
distance_to_primary	0.2876	0.0136
local_population	0.2761	0.013
women_of_reproductive_age_ratio_1km	0.2657	0.0125
acled_index	0.2599	0.0123
distance_to_town	0.255	0.012
distance_to_secondary	0.2435	0.0115
youth_ratio_1km	0.2377	0.0112
distance_to_tertiary	0.2228	0.0105
population_1km	0.2209	0.0104
water_source_category	0.2196	0.0104
rwi	0.1821	0.0086
elderly_ratio_1km	0.1623	0.0077
children_under_five_ratio_1km	0.1544	0.0073
orig_status_id	0.1172	0.0055
population_10km	0.067	0.0032
men_ratio_10km	0.0566	0.0027
women_of_reproductive_age_ratio_10km	0.0202	0.001
population_100km	0.01	0.0005
youth_ratio_100km	0.0	0.0

8 Validation Testing and Stability

To find patterns in a dataset from which it can make predictions, an algorithm must first learn from a historical example – typically from a historical dataset that contains the output variable you want to predict. However, if a model is trained too closely on its training data then it may be overfit. Overfitting is a modeling error that occurs when a model is too closely fit to training data and therefore performs poorly on out-of-sample data (data that was not used to train the model).

Overfitting generally results in an overly complex model that explains idiosyncrasies and random noise in the training data, rather than the underlying trends that the model was intended to capture. To avoid overfitting, the best practice is to evaluate model performance on out-of-sample data. If the model performs very well on in-sample data, (the training data) but poorly on out-of-sample data, that may be an indication that the model is overfit.

DataRobot uses standard modeling techniques to validate model performance and ensure that overfitting does not occur. DataRobot used a robust model k-fold cross-validation framework to test the out-of-sample stability of a model's performance. In addition to cross-validation partitioning, DataRobot uses a holdout sample to further test out-of-sample model performance and ensure the model is not overfit.

The following procedure was used during development to insure that overfitting did not occur:

- All values of the feature "fold" represent separate partitions used for cross-validation

DataRobot calculates the Cross Validation scores for each of the training data partitions or folds. The project metric used to calculate the score is LogLoss.

8.1 Cross Validation Scores

Fold	Cross Validation Score (LogLoss)
Fold 1	0.36521
Fold 2	0.36718
Fold 3	0.3673
Fold 4	0.36411
Fold 5	0.36244

9 Model Results

DataRobot runs performance testing during the model development process to evaluate model results and reliability. The validation, cross-validation, and holdout (if applicable) out-of-sample performance scores are presented below, as well as the number of observations for each partition. The performance metric used for this project was LogLoss and the project included a total of 316,503 observations. An asterisk (*) next to a score, whether validation or holdout, indicates that DataRobot used in-sample predictions to derive the score. (In-samples predictions are those that include data from the validation or holdout partitions due to sample size used to build the model.)

Scoring Type	Score (LogLoss)
cross_validation	0.3652*
holdout	0.3039*

validation	0.3652*

10 Bias and Fairness

The Bias & Fairness feature is not a legal compliance tool. Please carefully review the product Documentation to ensure that you fully understand the functionality of the feature before using it. DataRobot recommends that you take local legal advice where you wish to rely on the results of this feature for complying with any legal obligations. Product Documentation can be found here: https://app.datarobot.comdocs/modeling/special-workflows/b-and-f/index.html

DataRobot's Bias and Fairness testing identifies whether the model exhibits biased behavior towards any classes in the dataset's protected features, based on the selected definition of fairness. Protected features and the fairness metric are chosen before Autopilot is started. DataRobot also provides a workflow that guides you towards an appropriate definition of fairness for the specific use case.

DataRobot's Bias and Fairness feature includes two model-level insights:

- Per-Class Bias, which shows whether the model is treating certain protected groups differently as measured by the selected fairness metric. This identifies if there is biased behavior, and if so, how that bias manifests, but not why.
- Cross-Class Data Disparity, which shows how different protected classes differ in their data distribution. This offers deeper insight into why the model is treating groups differently.

Together, these insights can help identify potential mitigation strategies for bias in the dataset and model, such as improving data collection or data sampling for specific groups.

Bias and Fairness testing was used in this project. The selected protected features were clean_country_id. The favorable target outcome was No. The selected fairness metric was favorableAndUnfavorablePredictiveValueParity. The fairness threshold was set at 0.8.

Favorable Predictive Value & Unfavorable Predictive Value Parity (also known as Precision and NPV Parity) measure fairness by Equal Error, and it is best suited for cases when the target is severely unbalanced. These metrics measure whether your model disproportionally discovers favorable cases or omits unfavorable cases correctly for any particular group. It is calculated relative to the model's predictions--the proportion of predicted favorable or predicted unfavorable cases that are classified correctly. Unlike True Favorable and Unfavorable Rate Parity, Favorable Predictive Value & Unfavorable Predictive Value Parity are calculated relative to the predicted class, rather than the ground truth, which allows it to capture more meaningful information when there are very few predictions for a certain class, such as in imbalanced datasets. Higher values for Favorable Predictive Value and Unfavorable Predictive Value mean that the model is more accurate for that class relative to that metric. It's important to measure both Favorable Predictive Value &

Unfavorable Predictive Value Parity together, because poor performance in either one can lead to biased outcomes, often for different stakeholders. These metrics will help you investigate how your model balances both of these types of accuracy across your protected classes.

The Per-Class Bias graph shows whether the model exhibits biased behavior across protected features. The top fairness score across each protected class is scaled to 1.0, and the fairness scores for every other class are scaled relative to that value. If the fairness score for a class crosses the selected fairness threshold, the bar for that class is shown in red. If DataRobot used in-sample predictions to derive the model's performance scores (see **Overview of Model Results**), the fairness scores were calculated using in-sample validation data.

If there is not enough data for a class, its score is still calculated, but the bar for that class is shown in gray. The heuristic for whether a class does not have enough data is the following:

- If the class has <100 rows in the validation data, then it does not have enough data.
- If the class has between 100 and 1,000 rows in the validation data, but has fewer than <10% of the rows of the majority class, then it does not have enough data.
- If the class has >1,000 rows, then it has enough data.

The following figure is the Per-Class Bias graph for each protected feature in this project:

clean_country_id