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.

Copyright ©2024, DataRobot, Inc.



Table of Contents

* 1 How To Use This Document
* 2 DataRobot Model Development Documentation
* 3 Executive Summary and Model Overview
* 3.1 Model Stakeholders
* 3.2 Model Development Purpose and Intended Use
* 3.3 Model Description and Overview
* 3.4 Overview of Model Results
* 3.5 Model Interdependencies
* 4 Model Data Overview
* 4.1 Data Source Overview and Appropriateness
* 4.2 Input Data Extraction, Preparation, and Quality & Completeness
* 4.3 Data Assumptions
* 4.4 Personal Data Detection
* 5 Model Theoretical Framework and Methodology
* 5.1 Model Development Overview
* 5.2 Model Methodology
* 5.2.1 One-Hot Encoding Task
* 5.2.2 Median Value-Based Numeric Imputation (V2 with quick median algorithm)
* 5.2.3 Forecast Distance column extraction transformer
* 5.2.4 Extracts Naive Predictions
* 5.2.5 Eureqa Generalized Additive Model Regressor
* 5.3 Literature Review and References
* 5.4 Alternative Model Frameworks and Theories Considered
* 5.5 Variable Selection
* 5.5.1 DataRobot Quantitative Analysis
* 5.5.2 Expert Judgement and Variable Selection
* 5.5.3 Final Model Variables
* 5.5.3.1 Model Features and Summary Statistics
* 5.5.3.2 Data Quality Handling Report
* 6 Model Performance and Stability
* 6.1 Model Validation Stability
* 6.1.1 Data Partitioning Methodology
* 6.2 Model Performance (Backtesting)
* 6.3 Sensitivity Testing and Analysis
* 6.3.1 Lift Chart
* 6.3.2 Key Relationships
* 6.3.3 Sensitivity Analysis (Partial Dependence)
* 6.3.4 Accuracy Over Time
* 7 Model Implementation and Output Reporting
* 7.1 Version Control

2 DataRobot Model Development Documentation

A key component of effective model risk management is sufficiently detailed documentation for model development, implementation, and use, so that reasonable parties unfamiliar with a model can understand how the model operates, its limitations, and its key assumptions. Additionally, model documentation should contain enough detail for an independent party (e.g., independent model validation) to replicate all aspects of the underlying modeling process.

The purpose of this document is not to be prescriptive in format and content, but rather to serve as a guide in creating sufficiently rigorous model development, implementation, and use documentation. The documentation should provide enough evidence to show that the components of the model work as intended, the model is appropriate for its intended business purpose, and that it is conceptually sound.

3 Executive Summary and Model Overview

3.1 Model Stakeholders

Describe the model's purpose and its intended business use. Describe all stakeholders of this model, including their role, line-of-business, and team. This should include stakeholders of model ownership, model development, model implementation, and model risk management.

Model Owner(s): The individual who owns the business risk addressed by the model and provides approval for the model to be used within the line-of-business or enterprise function.

Model Developer(s): The individual responsible for building new models with DataRobot or maintaining existing models.

Model User(s): Those teams who will use the model output as part of their ongoing business operations.

Model Validator(s): The validators are responsible for independent model review and approval prior to its first use.

3.2 Model Development Purpose and Intended Use

Describe the model's purpose, including a summary of the business need for this particular model. Concisely describe how the model will be used to address this business problem. Furthermore, describe with great precision all model uses covered by this document. These descriptions will address this statement made in regulatory guidance, FRB SR-11-7, "Even a fundamentally sound model producing accurate outputs consistent with the design objective of the model may exhibit high model risk if it is misapplied or misused."

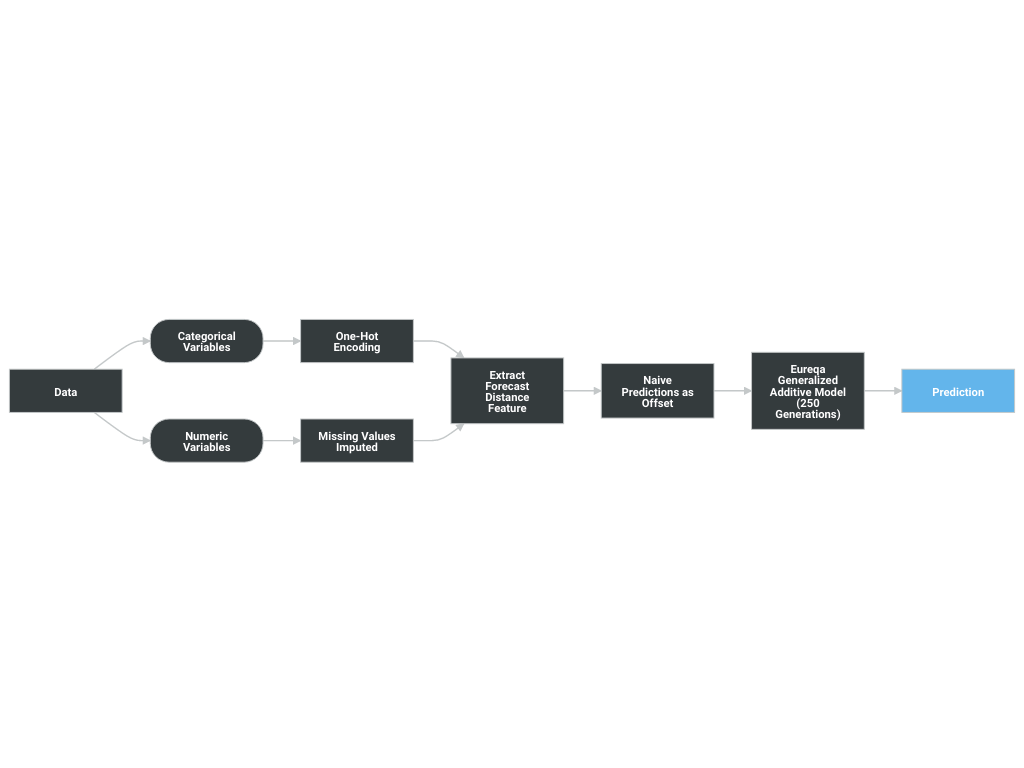
3.3 Model Description and Overview

The particular model referenced in this document: Eureqa Generalized Additive Model (250 Generations). This model was developed in a project created with vfb53b6f8ee350741 of DataRobot. This model is denoted within DataRobot by the Project ID: 67602b40fb3010b52c114b1e and the Model ID: 67602cfd72e45b01310b2b2e. The project was created on 2024-12-16 13:29:36.

The selected Eureqa model (complexity 1):

Target = 0.546254769137672\*(total net eur (6 month max) (diff 24 month mean)) - 6843.22679588615

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:

* One-Hot Encoding
* Missing Values Imputed
* Extract Forecast Distance Feature
* Naive Predictions as Offset
* Eureqa Generalized Additive Model (250 Generations)

3.4 Overview of Model Results

DataRobot runs performance testing during the model development process to evaluate model results and reliability. An overview of the out-of-sample performance scores are included below. The performance metric used for this project was RMSE and the project included a total of 1,254 observations.

|  |  |
| --- | --- |
| Scoring Type | Score (RMSE) |
| backtesting\_scores | 93107.9069, 86615.7141, 50781.5342, 135827.2314 |
| holdout | 119972.9334 |
| validation | 93107.9069 |

3.5 Model Interdependencies

Understanding interdependent relationships allows for enhanced understanding and improved ability to manage and aggregate model risk. Explain how this model is interconnected with other models in the model inventory--that is, the relative direction with regard to the model's position in the receiving input/sending output role. In addition to the directional relationship, also provide a brief description of each interconnected model.

4 Model Data Overview

4.1 Data Source Overview and Appropriateness

Explain how the data is suitable and relevant for the business problem and model use. For example:

Describe how, and from where, the data was obtained.

Provide a detailed description of the data source and its relevance to the business problem being addressed by this model.

Assess whether the data used for model development is appropriate given the populations to which the model will be applied.

If the model development and model implementation data sources differ, provide a detailed explanation justifying the use of different data sources.

4.2 Input Data Extraction, Preparation, and Quality & Completeness

Provide a detailed description of the data extraction and preparation process, and discuss any analysis conducted to confirm the data are complete and of sufficient quality (e.g., data validation). Include a detailed description of the data extraction process, hierarchical by extraction and preparation stage, and calling sequence. Provide data extraction code (e.g., SQL, Spark, etc.) in the Appendix.

Review and comment on any data weaknesses and limitations and their probable potential effects on the model. For example, data truncation, extraction timing, through-the-cycle data, and data exclusions could potentially cause unintended effects on the model.

4.3 Data Assumptions

Comment on data assumptions, the potential effects on the model, and any mitigating data controls. For example, assumptions related to data truncation, extraction timing, through-the-cycle data, reliability of source system, manual data overrides or imputation, and data exclusions could potentially cause unintended effects on the model.

4.4 Personal Data Detection

The feature list used to develop this model has not been scanned for the presence of Personal Data.

5 Model Theoretical Framework and Methodology

5.1 Model Development Overview

DataRobot simplifies model development by performing a parallel heuristic search for the best model or ensemble of models, based on both the characteristics of the data and the prediction target. While some machine learning techniques tend to consistently outperform others, it is rarely possible to say in advance which will perform best for a given business problem. Therefore, during the modeling process, DataRobot develops dozens of independent challenger models, exposes the details of how these models were built and how they perform, and enables the user to select the best model for the particular business problem being addressed.

The fundamental workflow within DataRobot for model development is as follows:

* Data Ingestion: The user creates a modeling dataset that includes the prediction target and loads it into DataRobot for time-aware modeling.
* Target Selection: The user identifies the target prediction and selects the primary date/time feature for which the prediction target will vary over time. This will enable time-aware modeling using DataRobot. DataRobot detects whether the target is categorical or numeric. If the target is categorical, DataRobot selects and builds classification blueprints. If the target is numeric, DataRobot selects and builds regression blueprints. DataRobot also selects an optimization performance metric based on the type of supervised learning problem, which can be changed by the user.
* Automated Data Preparation: DataRobot automatically partitions the input dataset into date/time partitioning for validation, which can also be defined by the user.
* Stationarity Analysis & Feature Engineering: DataRobot's time series functionality works by encoding time-sensitive components (such as lags and moving averages) as features, transforming your original input dataset into a modeling dataset that can use conventional machine learning techniques. DataRobot automatically creates and selects time series features in the modeling data and automatically detects whether a project's target value is stationary (that is, whether the statistical properties of the target are constant over time). If the target is not stationary, DataRobot attempts to make it stationary by applying a differencing strategy prior to modeling. This improves the accuracy and robustness of the underlying models.
* DataRobot uses information about the selected target variable and predictors to define a set of candidate blueprints for analysis. It then trains models for each blueprint and ranks them on the model Leaderboard based on a validation and holdout accuracy score.
* Transparent Model Evaluation and Selection: DataRobot has built-in diagnostic tools to assess model accuracy and performance. Once DataRobot has trained and tested models, users can access them from the Leaderboard. From there, users can review model accuracy and, using built-in model diagnostic tools, understand how each independently built model performs. DataRobot provides many metrics for evaluating model accuracy and performance.
* Model Deployment and Monitoring: Once the final model is selected, DataRobot provides efficient solutions for deployment (i.e., model implementation) and monitoring.

5.2 Model Methodology

The modeling workflow consists of the following elements, which connect to create the blueprint:

* One-Hot Encoding
* Missing Values Imputed
* Extract Forecast Distance Feature
* Naive Predictions as Offset
* Eureqa Generalized Additive Model (250 Generations)

The following subsections include details for each node of the modeling blueprint.

5.2.1 One-Hot Encoding Task

This transformer will do binary one-hot (aka one-of-K) coding. One boolean-valued feature is constructed for each of the possible string values that the feature can take. For inputs with only 2 unique values, only one boolean-valued feature will be constructed

This encoding is needed for feeding categorical data to many estimators, notably linear models and SVMs.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Name | Description | Best Searched |
| int | card\_max | An integer that specifies the maximum number of unique values. values: [1, 99999] | 50000 |
| int | card\_min | An integer that specifies the minimum number of unique values. values: [1, 99999] | 1 |
| bool | drop\_cols | drop\_cols, If True, drop last level of each feature values: [False, True] | False |
| select | flag | flag, If all, add highcat-cols to metadata values: ['None', 'all'] | None |
| int | max\_features | If the total number of categories created across all features exceeds this value, the top max\_features most frequent categories will persist. All others will be either thrown out or grouped. A value of None disables the limit. values: [1, 999999] | 20000 |
| int | min\_support | The minimum number of records for a category to be represented in one hot encoding. If a category has fewer counts it will be grouped with other small cardinality values. values: [1, 99999] | 1 |

5.2.2 Median Value-Based Numeric Imputation (V2 with quick median algorithm)

For a numeric feature, impute rows of missing values with median value (V2).

Impute missing values on numeric variables with their median and create indicator variables to identify records that were imputed. A quick median algorithm (based on np.partition) is implemented to compute median feature value.

Imputation strategy:

A numeric feature is imputed with the median value if there are enough finite values in the feature samples used to train a numeric imputation task (e.g., > t, default: 50) and there are rows with NaN or infinite values in the samples to be imputed.

After imputation, the imputed numeric features will be scaled if the argument S is set to True. The feature will use scaled rounding (i.e., rounding to a logarithmic scale).

Imputation indicator:

The indicator column (0, 1) is added to indicate imputed rows if the numeric feature is imputed with : 1) the median value and with at least one row with NaN and 2) at least two unique values.

Example:

An imputation task is initialized with t=2.

Input numeric features of this task:

feature0, feature1, feature2, feature3

1, 2, np.nan, np.nan

2, 3, np.nan, 18

3, 2, np.nan, 16

4, 1, 13, 14

20, 1, 45, 46

Output numeric features of this task:

feature0, feature1, feature2, feature2-mi, feature3, feature3-mi

1, 2, 45, 1, 18, 1

2, 3, 45, 1, 18, 0

3, 2, 45, 1, 16, 0

4, 1, 13, 0, 14, 0

20, 1, 45, 0, 46, 0

In the imputation output, median value imputation is run on feature2 and feature3. The feature2-mi is the indicator column for the imputation on feature2. The feature3-mi is the indicator column for the imputation on feature3.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Name | Description | Best Searched |
| bool | scale\_small | True if small values (range of the numeric variable is <= 1) are to be scaled. values: [False, True] | False |
| int | threshold | Minimum number of required finite elements in a column to impute the data onto NaNs and INFs. values: [1, 99999] | 10 |

5.2.3 Forecast Distance column extraction transformer

This transformer extracts the forecast distance column data from modeling data and returns that data as X.

5.2.4 Extracts Naive Predictions

Compared to BaselineExtractorV2, this disables tuning output\_type.

This transformer extracts the Naive Prediction column data from modeling data. When the naive prediction is used for offset, the one with the longest periodicity will be selected. For non-stationary time series, it returns that data as a predictions\_to\_boost. For stationary time series, it returns that data as a feature. If output\_type is both, it returns that data as both offset and feature. The output is converted to the link function scale of the main modeler.

5.2.5 Eureqa Generalized Additive Model Regressor

Eureqa Generalized Additive Model is a surrogate model that approximates Gradient Boosting Machine predictions using Eureqa modeling engine.

Eureqa is a proprietary AI-powered modeling engine that automates much of the heavy lifting inherent in analytics and data science. Leveraging automated evolutionary algorithms, the Eureqa modeling engine (in DataRobot) churns through your data to quickly create accurate predictive models. Developed in Cornell’s Artificial Intelligence Lab, the Eureqa engine leverages an evolutionary approach to model creation, testing billions of potential models per second, and converging on the simplest, most accurate models that explain your data.

Compared to other machine learning outputs, Eureqa Generalized Additive Models are simple and transparent. The models are presented as mathematical equations, enabling users to easily understand results and recommendations.

Eureqa Expressions:

Eureqa Expressions are human-readable mathematical equations describing relationships between features in a dataset. For example, the following is a valid Eureqa Expression string; also a model that Eureqa found to describe the behavior of an actual physical double-pendulum system, based on measured position, velocity, and acceleration data from the two pendulums:

Target = v1^2\*sin(x1 - x2) - a1\*cos(x2 - x1) - 9.82841593744553\*sin(x2)

with dataset variables: x1; x2; v1; v2; a1; and a2 as Target.

Generally speaking, Eureqa’s Expressions syntax is intended to work such that, if an engineer familiar with common mathematical tools types in an equation that they would expect to work in other common programming languages, it will work in Eureqa as well. Its properties should be familiar to a user with a background in basic algebra. (For example, infix operators such as +, prefix operators such as sin(), variables, numeric constants, parentheses as a grouping symbol, whitespace between operators, etc., are generally not important.)

Eureqa Expressions cannot, in general, be parsed correctly without a list of variable names that may be used in that expression. For example, consider the following expression:

Target = x\*y + 2\*(x + y)

A casual reader might think that this evaluates to the product of x and y, plus twice the sum. But let’s try that again, with the list of variables:

Target = x\*y + 2\*(x + y)

with dataset variables: x; y; and (x + y), which is a valid name for a single Eureqa variable. In this case, Eureqa will parse the expression as the product of x and y plus twice the value of the variable named (x + y).

Eureqa Expressions are based on the set of variable names available to the Eureqa Estimator when the Eureqa Estimator is fitting a model. This, in turn, depends on: (1) the columns available in the raw data; (2) the features selected by the current featurelist; (3) any features that are added or removed by preprocessing steps in Eureqa’s blueprint.

This most commonly becomes relevant with text columns: Eureqa does not support text input. All Eureqa variables must be numeric. As a result, DataRobot will encode text-based variables into multiple numeric columns.

The list of variable names does not include the original name of the target variable. It should always be called Target in Eureqa expressions. In the double-pendulum example above, this is the reason the expression begins Target = ... and not a2 = ....

Eureqa Expressions are used when setting the following tuning parameters: prior\_solutions, target\_expression\_string, training\_split\_expr, validation\_split\_expr, weight\_expr.

Variable Name Encoding:

Eureqa has very broad support for different variable names, but it doesn’t support arbitrary variable names. If an unsupported variable name is provided, that variable’s name is encoded.

Unusual special characters will be replaced with underscores. Purely numeric variable names will be surrounded in underscores. Unusual whitespace characters are stripped out entirely.

If a variable name looks like an expression, it will be wrapped in parentheses. If a variable name contains mismatched parentheses, parentheses will be treated as “unusual special characters” (see above).

If Eureqa is provided with two variables that share the same name, or if the rules above otherwise cause a name collision, the duplicates will be renamed as (<var> #1), (<var> #2), etc. (replacing <var> with the original encoded variable name).

Eureqa has a small set of reserved keywords (generally these are terms that are commonly used as mathematical constants, such as pi or TRUE). Variables with these names are treated as name collisions, and are de-duplicated (see above).

Building Blocks:

Eureqa “Building Blocks” are the mathematical operators and other components that comprise Eureqa Expressions. For example, + is a building block; sqrt() is a building block; there’s a building block representing numerical constants (real numbers); etc.

The building\_blocks Advanced Tuning parameters affect model evolution, so they are only relevant to model solution expressions. Expressions that don’t directly describe models – for example, a weight expression, or a training or validation row-selection expression – aren’t affected by building\_blocks settings because those expressions do not get evolved by the modeling process.

Building blocks can be disabled, and disabled building blocks will not be added to models. If the block is injected artificially, such as via the target\_expression\_string, it may still appear in a model even if its building block is disabled.

Building blocks can also be set to a numeric value. This value is the block’s “complexity penalty”. A penalty of 0 means that Eureqa models can use the building block as many times as they want without being penalized. This may cause models to become “cluttered”; they may use the operator in question unnecessarily.

Complexity penalties factor into an overall model’s “Complexity” score: A model’s base Complexity score is computed by iterating over all building blocks in the model and adding up their individual complexity scores. For example, if + appears 3 times in a model, and + is given a complexity penalty of 2, that will increase the model’s complexity by 6. (Additional factors may also be taken into account when converting a base complexity score into the final complexity score that is displayed to users.)

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Name | Description | Best Searched |
| int or select | EUREQA\_building\_block\_\_absolute\_value | ‘Absolute Value’ building block. Allows Eureqa to use the “abs()” operator in model expressions, and sets its complexity penalty. Usage: abs( x ), which returns the positive value of x, without regard for its sign. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_addition | ‘Addition’ building block. Allows Eureqa to use the “+” operator in model expressions, and sets its complexity penalty. Usage: x + y or add( x, y ), which returns the sum of x and y. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_arccosine | ‘Arccosine’ building block. Allows Eureqa to use the “acos()” operator in model expressions, and sets its complexity penalty. Usage: acos( x ). (The standard trigonometric arccosine function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_arcsine | ‘Arcsine’ building block. Allows Eureqa to use the “asin()” operator in model expressions, and sets its complexity penalty. Usage: asin( x ). (The standard trigonometric arcsine function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_arctangent | ‘Arctangent’ building block. Allows Eureqa to use the “atan()” operator in model expressions, and sets its complexity penalty. Usage: atan( x ). (The standard trigonometric arctangent function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_ceiling | ‘Ceiling’ building block. Allows Eureqa to use the “ceil()” operator in model expressions, and sets its complexity penalty. Usage: ceil( x ), which returns the smallest integer not less than x. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_complementary\_error\_function | ‘Complementary Error Function’ building block. Allows Eureqa to use the “erfc()” operator in model expressions, and sets its complexity penalty. Usage: erfc( x ). 1.0 - erf( x ) where erf( x ) is the integral of the normal distribution and returns a value between 2 and 0. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_constant | ‘Constant’ building block. Allows Eureqa to use constants in model expressions, and sets their complexity penalty. Usage: c, where c is a real valued constant. values: [0, 100] or "Disabled" | None |
| int or select | EUREQA\_building\_block\_\_cosine | ‘Cosine’ building block. Allows Eureqa to use the “cos()” operator in model expressions, and sets its complexity penalty. Usage: cos( x ), where the angle (x) is in radians. (The standard trigonometric cosine function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_division | ‘Division’ building block. Allows Eureqa to use the “/” operator in model expressions, and sets its complexity penalty. Usage: x / y or div( x, y ), which returns the quotient of x and y (where y must be non-zero). values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_equal-to | ‘Equal-To’ building block. Allows Eureqa to use the “=” operator in model expressions, and sets its complexity penalty. Usage: equal( x, y ) or x = y, which returns 1 if x is numerically equal to y, 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_error\_function | ‘Error Function’ building block. Allows Eureqa to use the “erf()” operator in model expressions, and sets its complexity penalty. Usage: erf( x ). Integral of the normal distribution; returns a value between -1 and +1. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_exponential | ‘Exponential’ building block. Allows Eureqa to use the “exp()” operator in model expressions, and sets its complexity penalty. Usage: exp( x ), which returns e^x. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_factorial | ‘Factorial’ building block. Allows Eureqa to use the “!” operator in model expressions, and sets its complexity penalty. Usage: factorial( x ) or x!, which returns the product of all positive integers from 1 to x. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_floor | ‘Floor’ building block. Allows Eureqa to use the “floor()” operator in model expressions, and sets its complexity penalty. Usage: floor( x ), which returns the largest integer not greater than x. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_gaussian\_function | ‘Gaussian Function’ building block. Allows Eureqa to use the “gauss()” operator in model expressions, and sets its complexity penalty. Usage: gauss( x ), which returns exp( -x^2 ). This is a bell-shaped squashing function. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_greater-than | ‘Greater-Than’ building block. Allows Eureqa to use the “>” operator in model expressions, and sets its complexity penalty. Usage: greater( x, y ) or x > y, which returns 1 if x > y, 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_greater-than-or-equal | ‘Greater-Than-Or-Equal’ building block. Allows Eureqa to use the “>=” operator in model expressions, and sets its complexity penalty. Usage: greater\_or\_equal( x, y ) or x >= y, which returns 1 if x >= y, 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_hyperbolic\_cosine | ‘Hyperbolic Cosine’ building block. Allows Eureqa to use the “cosh()” operator in model expressions, and sets its complexity penalty. Usage: cosh( x ). (The standard trigonometric hyperbolic cosine function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_hyperbolic\_sine | ‘Hyperbolic Sine’ building block. Allows Eureqa to use the “sinh()” operator in model expressions, and sets its complexity penalty. Usage: sinh( x ). (The standard trigonometric hyperbolic sine function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_hyperbolic\_tangent | ‘Hyperbolic Tangent’ building block. Allows Eureqa to use the “tanh()” operator in model expressions, and sets its complexity penalty. Usage: tanh( x ). (The hyperbolic tangent of x.) Hyperbolic tangent is a common squashing function that returns a value between -1 and +1. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_if-then-else | ‘If-Then-Else’ building block. Allows Eureqa to use the “if()” operator in model expressions, and sets its complexity penalty. Usage: if( x, y, z ), which returns y if x is greater than 0, z otherwise; if x is nan, the function returns z. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_input\_variable | ‘Input Variable’ building block. Allows Eureqa to use variables in model expressions, and sets their complexity penalty. Usage: x, where x is a variable in your prepared dataset. values: [0, 100] or "Disabled" | None |
| int or select | EUREQA\_building\_block\_\_integer\_constant | ‘Integer Constant’ building block. Allows Eureqa to use integer constants in model expressions, and sets their complexity penalty. Usage: c, where c is an integer constant. values: [0, 100] or "Disabled" | None |
| int or select | EUREQA\_building\_block\_\_inverse\_hyperbolic\_cosine | ‘Inverse Hyperbolic Cosine’ building block. Allows Eureqa to use the “acosh()” operator in model expressions, and sets its complexity penalty. Usage: acosh( x ). (The standard inverse hyperbolic cosine function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_inverse\_hyperbolic\_sine | ‘Inverse Hyperbolic Sine’ building block. Allows Eureqa to use the “asinh()” operator in model expressions, and sets its complexity penalty. Usage: asinh( x ). (The standard inverse hyperbolic sine function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_inverse\_hyperbolic\_tangent | ‘Inverse Hyperbolic Tangent’ building block. Allows Eureqa to use the “atanh()” operator in model expressions, and sets its complexity penalty. Usage: atanh( x ). (The standard inverse hyperbolic tangent function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_less-than | ‘Less-Than’ building block. Allows Eureqa to use the “<” operator in model expressions, and sets its complexity penalty. Usage: less( x, y ) or x < y, which returns 1 if x < y, 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_less-than-or-equal | ‘Less-Than-Or-Equal’ building block. Allows Eureqa to use the “<=” operator in model expressions, and sets its complexity penalty. Usage: less\_or\_equal( x, y ) or x <= y, which returns 1 if x <= y, 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_logical\_and | ‘Logical And’ building block. Allows Eureqa to use the “and” operator in model expressions, and sets its complexity penalty. Usage: and( x, y ), which returns 1 if both x and y are greater than 0, 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_logical\_not | ‘Logical Not’ building block. Allows Eureqa to use the “not” operator in model expressions, and sets its complexity penalty. Usage: not( x ), which returns 0 if x is greater than 0, 1 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_logical\_or | ‘Logical Or’ building block. Allows Eureqa to use the “or” operator in model expressions, and sets its complexity penalty. Usage: or( x, y ), which returns 1 if either x or y are greater than 0, 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_logical\_xor | ‘Logical Xor’ building block. Allows Eureqa to use the “xor” operator in model expressions, and sets its complexity penalty. Usage: xor( x, y ), which returns 1 if (x <= 0 and y > 0) or (x > 0 and y <= 0), 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_logistic\_function | ‘Logistic Function’ building block. Allows Eureqa to use the “logistic()” operator in model expressions, and sets its complexity penalty. Usage: logistic( x ), which returns 1/( 1 + exp( -x ) ). This is a common sigmoid (s-shaped) squashing function that returns a value between 0 and 1. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_maximum | ‘Maximum’ building block. Allows Eureqa to use the “max()” operator in model expressions, and sets its complexity penalty. Usage: max( x, y ), which returns the maximum (signed) result of x and y. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_minimum | ‘Minimum’ building block. Allows Eureqa to use the “min()” operator in model expressions, and sets its complexity penalty. Usage: min( x, y ), which returns the minimum (signed) result of x and y. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_modulo | ‘Modulo’ building block. Allows Eureqa to use the “mod()” operator in model expressions, and sets its complexity penalty. Usage: mod( x, y ), which returns the remainder of x / y. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_multiplication | ‘Multiplication’ building block. Allows Eureqa to use the “\*” operator in model expressions, and sets its complexity penalty. Usage: x \* y or mul( x, y ), which returns the product of x and y. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_natural\_logarithm | ‘Natural Logarithm’ building block. Allows Eureqa to use the “log()” operator in model expressions, and sets its complexity penalty. Usage: log( x ), which returns the natural logarithm (base e) of x. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_negation | ‘Negation’ building block. Allows Eureqa to use the “-” unary operator in model expressions, and sets its complexity penalty. Usage: -x, which returns the inverse of x. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_power | ‘Power’ building block. Allows Eureqa to use the “^” operator in model expressions, and sets its complexity penalty. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_round | ‘Round’ building block. Allows Eureqa to use the “round()” operator in model expressions, and sets its complexity penalty. Usage: round( x ), which returns an integer of x rounded to the nearest integer. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_sign\_function | ‘Sign Function’ building block. Allows Eureqa to use the “sign()” operator in model expressions, and sets its complexity penalty. Usage: sgn( x ), which returns -1 if x is negative, +1 if x is positive, and 0 if x is zero. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_sine | ‘Sine’ building block. Allows Eureqa to use the “sin()” operator in model expressions, and sets its complexity penalty. Usage: sin( x ), where the angle (x) is in radians. (The standard trigonometric sine function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_square\_root | ‘Square Root’ building block. Allows Eureqa to use the “sqrt()” operator in model expressions, and sets its complexity penalty. Usage: sqrt( x ), which returns the square root of x (where x must be positive). values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_step\_function | ‘Step Function’ building block. Allows Eureqa to use the “step()” operator in model expressions, and sets its complexity penalty. Usage: step( x ), which returns 1 if x is positive, 0 otherwise. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_subtraction | ‘Subtraction’ building block. Allows Eureqa to use the “-” binary operator in model expressions, and sets its complexity penalty. Usage: x - y or sub( x, y ), which returns the difference of x and y. values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_tangent | ‘Tangent’ building block. Allows Eureqa to use the “tan()” operator in model expressions, and sets its complexity penalty. Usage: tan( x ), where the angle (x) is in radians. (The standard trigonometric tangent function.) values: [0, 100] or "Disabled". | None |
| int or select | EUREQA\_building\_block\_\_two-argument\_arctangent | ‘Two-Argument Arctangent’ building block. Allows Eureqa to use the “atan2()” operator in model expressions, and sets its complexity penalty. Usage: atan2( y, x ). (The standard trigonometric two-argument arctangent function.) values: [0, 100] or "Disabled". | None |
| int | EUREQA\_max\_generations | The maximum number of evolutionary generations to run. Eureqa will run until either of max\_generations or timeout\_sec is reached. values: [0, 1e16] | None |
| int | EUREQA\_num\_threads | The number of threads Eureqa will run with. Ideally equal to the number of cores available | None |
| string | EUREQA\_prior\_solutions | Prior Eureqa Solutions. This field contains multiple Eureqa Expressions, one per line. Each Expression should be a valid Eureqa Solution, such as a Solution returned by a previous run of Eureqa. (You may need to edit the Solution such that the target variable is entered as “Target”, not the original column name of the target.) Each expression is fed into Eureqa’s initial evolutionary population. Eureqa makes no guarantees about keeping the form or content of these expressions in the final Pareto front of expressions that it generates, but if the expressions are good models or if they contain sub-expressions that are predictive features, Eureqa will generally take advantage of that information to converge on good solutions more quickly. values: multiple lines, each line is a valid Eureqa Expression as a string | None |
| int | EUREQA\_random\_seed | Constant to seed Eureqa’s pseudo-random number generator. Different values will cause Eureqa to generate different models on the same data and other input parameters. values: [0, 1e16] | None |
| select | EUREQA\_split\_mode | Whether to perform in-order (2) or random (1) splitting within the training set, for evolutionary re-training and re-validaton. values: [1, 2] values: [0, 1e5] | None |
| select | EUREQA\_sync\_migrations | Should Eureqa’s migrations be synchronized? If they are synchronized, Eureqa’s fit() function will be deterministic (repeated runs on the same data and parameters should produce the same models). Note that synchronization slows modeling down and models will take more time to generate. values: [False, True] | None |
| string | EUREQA\_target\_expression\_string | Eureqa Target Expression. Constrains the form of the models that Eureqa will consider. This field is typically of the form “Target = <some Eureqa Expression>”. See the description of “Eureqa Expressions”. The Expression must contain an equality operator, ie., it must be a full equation with a left-hand and a right-hand side. Note that Target Expressions will usually use function operators in their expressions, as well as regular expression operators. Function operators are templates / pattern matchers. For example, “Target = f(x, y z)” will match any expression that uses the variables x, y, and/or z (but will fail to match any function that uses other variables); “Target = f1(x, y) + f2(z)” will match “x\*y + z” but it will not match “x\*z + y”; “Target = f(sin(x))” will match “2\*sin(x)” but it will not match “sin(2\*x)”. This is a hard constraint. If you want to seed Eureqa with information from existing models, see prior\_solutions. If this field is left blank, Eureqa will automatically generate an appropriate target expression as part of the model fitting process. This expression will be logged to the model log (visible in the UI). values: valid Eureqa Expression, as a string. Must contain an equality operator. | None |
| float | EUREQA\_timeout\_sec | The duration of time to run the Eureqa search algorithm for Eureqa will run until either of max\_generations or timeout\_sec is reached. values: [0, 1e16] | None |
| float | EUREQA\_training\_fraction | What fraction of the DataRobot training data to use for Eureqa evolutionary training? This field is ignored if training\_split\_expression is set. Note that training\_fraction + validation\_fraction does not have to equal 1.0. If it is less than 1.0, some rows in the data are ignored. If it is greater than 1.0, Eureqa’s training and validation sets overlap. (This is not recommended, but may be required in some modeling scenarios involving small datasets. Note that DataRobot will still typically have a separate validation and holdout set.) values: [0.0, 1.0] | None |
| string | EUREQA\_training\_split\_expr | Eureqa Training Split Expression. Can be any valid Eureqa Expression, including a simple variable name. The expression should evaluate to either 0.0 or 1.0. If it evaluates to 1.0 on a specific row, that row is used for training data. If this field is not set, training\_fraction is used instead. Note that setting training\_split\_expr does not affect validation rows. If you want all non-training rows to be in the validation set, make sure to set validation\_split\_expr accordingly. values: valid Eureqa Expression, as a string | None |
| float | EUREQA\_validation\_fraction | What fraction of the DataRobot training data to use for Eureqa evolutionary validation? This field is ignored if validation\_split\_expression is set. Note that training\_fraction + validation\_fraction does not have to equal 1.0. If it is less than 1.0, some rows in the data are ignored. If it is greater than 1.0, Eureqa’s training and validation sets overlap. (This is not recommended, but may be required in some modeling scenarios involving small datasets. Note that DataRobot will still typically have a separate validation and holdout set.) values: [0.0, 1.0] | None |
| string | EUREQA\_validation\_split\_expr | Eureqa Validation Split Expression. Can be any valid Eureqa Expression, including a simple variable name. The expression should evaluate to either 0.0 or 1.0. If it evaluates to 1.0 on a specific row, that row is used for validation data. If this field is not set, validation\_fraction is used instead. Note that setting validation\_split\_expr does not impact training rows. If you want all non-validation rows to be in the training set, make sure to set training\_split\_expr accordingly. values: valid Eureqa Expression, as a string | None |
| string | EUREQA\_weight\_expr | Eureqa Weight Expression. Weights each row when evaluating the error of that row. The exact meaning of the weight depends on the specific value of error\_metric. Typically the per-row error is multiplied by the weight before being combined with other per-row weights into the aggregate weight. If this field is left as the empty string, Eureqa falls back to DataRobot’s default behavior. values: valid Eureqa Expression, as a string | None |
| select | XGB\_base\_margin\_initialize | If True, the intercept is initialized to the log odds of the target. values: [False, True] | True |
| floatgrid | XGB\_colsample\_bylevel | Subsample the features before each split in a tree. values: [0.1,1] | 1.0 |
| floatgrid | XGB\_colsample\_bytree | Subsample ratio of columns when constructing each tree. By default, the value of colsample\_bytree for XGBoost classes is 1.0. However, based on the training data, DataRobot may choose a different initial value for this parameter. values: [0,1] | 0.2 |
| int | XGB\_interval | Sets the interval for early stopping values: [2, 500] | None |
| floatgrid | XGB\_learning\_rate | Shrinks the contribution of each tree by learning\_rate. There is a trade-off between learning\_rate (lr) and n\_estimators(n). values: [5e-4,1] | 0.05 |
| select | XGB\_loss | loss function to be optimized. ‘ls’ refers to least squares regression. values: ['ls', 'labs', 'poisson', 'tweedie', 'gamma'] | None |
| int | XGB\_max\_bin | This is only used if ‘hist’ is specified as tree\_method. Maximum number of discrete bins to bucket continuous features. Increasing this number improves the optimality of splits at the cost of higher computation time. values: [16, 2048] | 256 |
| floatgrid | XGB\_max\_delta\_step | Maximum delta step we allow each tree’s weight estimation to be. If the value is set to 0, it means there is no constraint. If it is set to a positive value, it can help making the update step more conservative. Usually this parameter is not needed, but it might help in logistic regression when class is extremely imbalanced. Set it to value of 1-10 might help control the update values: [0,100] | 0.0 |
| intgrid | XGB\_max\_depth | maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree. Tune this parameter for best performance; the best value depends on the interaction of the input variables. Deeper the tree the more variable interactions the model can capture. For frozen models on larger sample sizes than parent model we increase the value of max\_depth to retain similar accuracy. values: [1, 16] | 3 |
| floatgrid | XGB\_min\_child\_weight | Minimum sum of instance weight(hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than min\_child\_weight, then the building process will give up further partitioning. In linear regression mode, this simply corresponds to minimum number of instances needed to be in each node. The larger, the more conservative the algorithm will be. values: [0.01,float(1e5)] | 5.0 |
| floatgrid | XGB\_min\_split\_loss | Minimum loss reduction required to make a further partition on a leaf node of the tree. the larger, the more conservative the algorithm will be. values: [0,1e5] | 0.01 |
| float | XGB\_missing\_value | The float value that should be treated as a missing value. When mono\_up or mono\_down are set, missing value will be set to -9999.0. values: [float(-1e5),float(1e5)] | None |
| string | XGB\_mono\_down | The id of the featurelist that defines the set of features with a monotonically decreasing relationship to the target. | None |
| string | XGB\_mono\_up | The id of the featurelist that defines the set of features with a monotonically increasing relationship to the target. | None |
| int | XGB\_n\_estimators | The number of boosting stages to perform. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance. values: [1,20000] | 530 |
| intgrid | XGB\_num\_parallel\_tree | Number of parallel trees created in each boosting stage. When this value is greater than 1, the model becomes a gradient-boosted random forest with (num\_parallel\_tree \* n\_estimators) trees. values: [1,16] | 1 |
| intgrid | XGB\_random\_state | The seed used in the random number generator 'values': [0, int(1e9)] | 1234 |
| multi | XGB\_reg\_alpha | L1 regularization term on weights, increase this value will make model more conservative. values: {'floatgrid': [0, 1e6], 'select': ['auto']} | 0.0 |
| multi | XGB\_reg\_lambda | L2 regularization term on weights, increase this value will make model more conservative. values: {'floatgrid': [0, 1e6], 'select': ['auto']} | 1.0 |
| float | XGB\_scale\_pos\_weight | Scaling factor for examples in the positive class. values: [0,float(1e9)] | 1.0 |
| int | XGB\_smooth\_interval | Sets the minimum interval for early stopping values: [2, 1000] | None |
| floatgrid | XGB\_subsample | subsample ratio of the training instance. Setting it to 0.5 means that XGBoost randomly collected half of the data instances to grow trees and this will prevent overfitting. | 1.0 |
| select | XGB\_tree\_method | The tree construction algorithm to be used. ‘auto’: Use heuristic to choose faster one. For small to medium dataset(<4M rows), exact greedy will be used. For very large-dataset(>=4M rows), approximate algorithm will be chosen. ‘exact’:Exact greedy algorithm. ‘approx’:Approximate greedy algorithm using sketching and histogram. ‘hist’: Fast histogram optimized approximate greedy algorithm. It uses some performance improvements such as bins caching. values: ['auto', 'exact', 'approx', 'hist'] | auto |
| float | XGB\_tweedie\_p | The power parameter of the tweedie distribution. Only applicable when Tweedie loss is used. values: [1, 2] | 1.5 |
| int | feature\_interaction\_max\_features | Specifies the max number of one vs all interactions to include in the pairwise calculations. For example, the default value of 50 will yield (50 ^ 2 - 50) / 2 pairwise interactions if feature\_interaction\_threshold is zero values: [0, int(1e5)] | 50 |
| int | feature\_interaction\_sampling | Specifies the extent (in number of rows) of downsampling used in the interaction calculations values': [1000, int(1e6)] | 2500 |
| float | feature\_interaction\_threshold | Specifies the minimum value a one vs all interaction strength must have to be include a given feature in the pairwise calculations. Zero includes all and one excludes all. values: [0.0, 1.0] | 0.1 |
| int | feature\_selection\_max\_features | The maximum number of features to include. ``values: {'int': [1, int(1e10)], 'select': ['no\_limit']}` | no\_limit |
| select | feature\_selection\_method | Method used to select features values: ['per\_variable', 'cumulative', 'no\_selection'] | no\_selection |
| int | feature\_selection\_min\_features | The minimum number of features to include. values: [1, int(1e5)] | 1 |
| float | feature\_selection\_threshold | The threshold used to select features. For cumulative method, threshold is typically close to 1 in order to retain more signal (ex:0.98). For per\_variable method, threshold would be close to 0 in order to discard only features with low signal. values: [0.0001, 1] | 0.001 |
| select | highdim\_modeling | Whether to include high cardinality and text features. values: [False, True] | True |
| int | subsample | Number of rows to sample for fitting the Eureqa model. values: [1000, 1e7] | 10000 |

5.3 Literature Review and References

* Suits, Daniel B. “Use of dummy variables in regression equations.” Journal of the American Statistical Association 52.280 (1957): 548-551. http://www.jstor.org/stable/2281705?seq=1
* [1] Acuna, Edgar, and Caroline Rodriguez. “The treatment of missing values and its effect on classifier accuracy.” Classification, Clustering, and Data Mining Applications. Springer Berlin Heidelberg, 2004. 639-647. https://link.springer.com/chapter/10.1007/978-3-642-17103-1\_60
* [2] Feelders, Ad. “Handling missing data in trees: Surrogate splits or statistical imputation?” Principles of Data Mining and Knowledge Discovery. Springer Berlin Heidelberg, 1999. 329-334. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.36.7991&rep=rep1&type=pdf
* Schmidt, Michael, and Hod Lipson. “Distilling free-form natural laws from experimental data.” Science 324.5923 (2009): 81-85. https://www.science.org/doi/10.1126/science.1165893

5.4 Alternative Model Frameworks and Theories Considered

As stated by regulatory guidance, comparison with alternative theories and approaches provides guidance for final model selection and is a fundamental component of a sound modeling process.

DataRobot develops dozens of alternative models, exposes the details of how these models were built and how they perform, and enables the user to select the best model for the particular business problem being addressed.

During the model development process, DataRobot considered the following alternative models. The final model was selected based on model performance as well as an analysis of model diagnostics and expert business judgment.

The performance metric used for this project was RMSE. The model types considered during the model selection process included the following models, which are sorted by the Holdout score.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model Name | Backtest Score | All Backtests Score | Holdout Score | Training Length |
| Ridge Regressor with Forecast Distance Modeling | 98881.2855 | N/A | 88263.5701 | 12 years 1 day |
| eXtreme Gradient Boosting on ElasticNet Predictions | 98325.2681 | N/A | 96057.9317 | 12 years 1 day |
| Performance Clustered Elastic Net Regressor with Forecast Distance Modeling | 98857.9199 | N/A | 100060.2791 | 12 years 1 day |
| Ridge Regressor using Linearly Decaying Weights with Forecast Distance Modeling | 96503.3455 | 110528.5028 | 100185.0696 | 12 years 1 day |
| Per Series Elastic Net Regressor with Forecast Distance Modeling | 98733.11 | N/A | 101399.4144 | 12 years 1 day |
| Temporal Hierarchical Model with Elastic Net and XGBoost | 72261.3096 | 132793.0576 | 102716.7823 | 12 years 1 day |
| Performance Clustered eXtreme Gradient Boosting on Elastic Net Predictions | 97677.3276 | N/A | 103220.2297 | 12 years 1 day |
| Light Gradient Boosting on ElasticNet Predictions (learning rate =0.3) | 114581.9116 | N/A | 104235.1283 | 12 years 1 day |
| Light Gradient Boosting on ElasticNet Predictions | 114775.5346 | N/A | 105582.3192 | 12 years 1 day |
| eXtreme Gradient Boosting on ElasticNet Predictions (learning rate =0.3) | 122324.4737 | N/A | 106579.1853 | 12 years 1 day |
| Baseline Predictions Using Most Recent Value | 98202.616 | N/A | 108809.6366 | 12 years 1 day |
| Eureqa Regressor (Quick Search: 250 Generations) | 80776.3867 | 121896.8084 | 108843.1004 | 12 years 1 day |
| eXtreme Gradient Boosted Trees Regressor with Early Stopping | 96617.3156 | N/A | 109436.6812 | 12 years 1 day |
| eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3) | 111122.5133 | N/A | 117616.9367 | 12 years 1 day |
| Performance Clustered eXtreme Gradient Boosted Trees Regressor | 109841.0653 | N/A | 118514.6529 | 12 years 1 day |
| Zero-Inflated Light Gradient Boosted Trees Regressor with Early Stopping (Gamma Loss) | 146134.3075 | N/A | 130558.3161 | 12 years 1 day |
| Ridge Regressor with Forecast Distance Modeling and Series Scaling | 189597.5312 | N/A | 227751.7932 | 12 years 1 day |
| Zero-Inflated eXtreme Gradient Boosted Trees Regressor with Early Stopping (Poisson Loss) | 144310.1679 | N/A | 247762.3273 | 12 years 1 day |
| Keras Slim Residual Neural Network Regressor using Training Schedule (1 Layer: 64 Units) | 326830.8929 | N/A | 358642.9651 | 12 years 1 day |

5.5 Variable Selection

The model's variable selection process includes a balance of quantitative analysis and key domain knowledge about the underlying business problem (i.e., expert judgment). The subsections below describe:

* DataRobot Quantitative Analysis: Key components related to variable selection that are automated by DataRobot.
* DataRobot automatically creates and selects time series features in the modeling data and will automatically detect whether or not a project's target value is stationary (that is, whether the statistical properties of the target are constant over time). If the target is not stationary, DataRobot attempts to make it stationary by applying a differencing strategy prior to modeling. This improves the accuracy and robustness of the underlying models.
* Expert Judgment and Variable Selection: Summary of the expert judgment used during the variable selection process.
* Final Model Variables: Final feature list chosen.

5.5.1 DataRobot Quantitative Analysis

A feature list is a defined set of features (variables) that DataRobot can use for modeling. DataRobot automatically creates three feature lists (described below) for each project. Users, however, can create customized feature lists that contain a subset of the total feature set, and use the new list to train new, alternative models. The default feature lists are described below:

* Informative Features (default): Features that pass a "reasonableness" check that determines whether they contain useful information. For example, DataRobot excludes features it determines are low information, such as a column containing all ones, duplicate columns, or a feature with too few values. The Informative Features list is sorted by each feature's correlation with the target variable.
* Raw Features: All features (variables) in the dataset, including those excluded from the Informative Features list.
* Univariate Selection: Features that meet a certain threshold for non-linear correlation with the selected target. DataRobot calculates, for each entry in the Informative Features list, the feature's individual relationship against the target.

Users also have the option to create user-defined feature transformations, which can then be included in a feature list for model exploration and to determine relative feature importance. Importance is measured using the information content of the variable; the calculation is done independently for each feature in the dataset. Features are then ranked on the Project Data from most to least important. This score represents a measure of predictive power using only that variable to predict the target. The score is measured using the project's accuracy metric that is defined by either the user (e.g., RMSE) or the default assigned by DataRobot.

5.5.2 Expert Judgement and Variable Selection

This section should include additional detail regarding the variable selection process and any expert judgment used during feature selection.

5.5.3 Final Model Variables

Below are two tables. The first contains a list of the final set of model feature variables, as well as summary statistics for the Eureqa Generalized Additive Model (250 Generations) 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).

5.5.3.1 Model Features and Summary Statistics

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Feature Name | Var Type | Unique | Missing | Mean | Std Dev | Median | Min | Max | Target Leakage |
| %key\_company | Categorical | 3 | 0 | N/A | N/A | N/A | N/A | N/A | N/A |
| platform no\_ | Categorical | 4 | 0 | N/A | N/A | N/A | N/A | N/A | N/A |
| sales date | Date | 239 | 0 | 2015-07-14 | 1939.37 days | 2015-09-01 | 2005-01-01 | 2024-11-01 | N/A |
| plant country | Categorical | 3 | 0 | N/A | N/A | N/A | N/A | N/A | N/A |
| %key\_DataRobot | Categorical | 6 | 0 | N/A | N/A | N/A | N/A | N/A | N/A |
| total net eur | Numeric | 1040 | 0 | 135200.93 | 208777.76 | 38162.98 | 0.0 | 1205714.3 | N/A |
| First sales date | Date | 4 | 0 | 2006-09-10 | 815.75 days | 2005-01-01 | 2005-01-01 | 2011-01-01 | N/A |
| last sales date | Date | 238 | 192 | 2015-04-30 | 1924.76 days | 2015-03-01 | 2005-01-01 | 2024-10-01 | N/A |
| revenue last 12 months | Numeric | 1055 | 192 | 1818288.39 | 2456675.58 | 810398.23 | 0.0 | 11324415.49 | N/A |
| plant country unemployment rate | Numeric | 141 | 0 | 9.093 | 3.703 | 8.7 | 2.9 | 18.3 | N/A |
| plant country registration | Numeric | 492 | 540 | 68550.17 | 114313.705 | 8810.0 | 4.72 | 427111.0 | N/A |
| plant country Freight | Numeric | 219 | 1018 | 8635.61 | 12113.37 | 27.67 | 17.62 | 30410.0 | N/A |
| plant country fuel | Numeric | 28 | 800 | 0.504 | 0.058 | 0.5 | 0.26 | 0.63 | N/A |
| plant country fatalities | Numeric | 228 | 546 | 116.47 | 116.35 | 64.0 | 0.16 | 583.0 | N/A |
| plant country Order books | Numeric | 362 | 0 | -14.6 | 17.11 | -11.4 | -80.0 | 29.8 | N/A |
| plant country Business situation | Numeric | 448 | 0 | 8.089 | 20.46 | 5.8 | -67.0 | 55.8 | N/A |
| plant country Finished good stocks | Numeric | 214 | 0 | 2.99 | 8.42 | 1.65 | -16.4 | 30.0 | N/A |
| plant country selling prices | Numeric | 293 | 0 | 5.48 | 13.34 | 4.6 | -53.0 | 71.9 | N/A |
| plant country Demand evolution | Numeric | 338 | 0 | 13.17 | 12.56 | 14.35 | -61.3 | 54.5 | N/A |
| plant country Production | Numeric | 366 | 0 | 8.23 | 13.86 | 8.4 | -78.8 | 60.7 | N/A |
| plant country household savings rate | Numeric | 131 | 648 | 11.78 | 5.12 | 9.9 | 2.38 | 27.33 | N/A |
| plant country economic situation | Numeric | 421 | 0 | -23.21 | 17.66 | -20.3 | -72.7 | 17.3 | N/A |
| plant country consumer prices | Numeric | 430 | 0 | 28.073 | 17.708 | 28.0 | -16.9 | 81.9 | N/A |
| plant country consumer price index | Numeric | 665 | 4 | 2.42 | 2.99 | 1.83 | -1.66 | 15.404 | N/A |
| plant country short\_term interest rate | Numeric | 261 | 0 | 0.905 | 1.59 | 0.21 | -0.58 | 5.11 | N/A |
| Number months until series project EOP | Numeric | 239 | 590 | 1012.58 | 66.15 | 1011.0 | 901.0 | 1139.0 | N/A |
| Number months since series project SOP | Numeric | 239 | 590 | 30.42 | 66.15 | 32.0 | -96.0 | 142.0 | N/A |
| Number months since first sale | Numeric | 239 | 0 | 106.103 | 63.62 | 104.0 | 0.0 | 238.0 | N/A |
| Number months since last sale | Numeric | 10 | 192 | 1.18 | 1.46 | 1.0 | 1.0 | 36.0 | N/A |
| First sale | Categorical | 2 | 0 | N/A | N/A | N/A | N/A | N/A | N/A |
| number months with revenue | Numeric | 238 | 0 | 80.12 | 64.601 | 73.0 | 0.0 | 237.0 | N/A |
| accumulated revenue | Numeric | 1063 | 0 | 10457402.83 | 15942477.75 | 4835235.016 | 0.0 | 74575373.57 | N/A |
| covid | Categorical | 2 | 0 | N/A | N/A | N/A | N/A | N/A | N/A |

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.

5.5.3.2 Data Quality Handling Report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature Name | Var Type | Missing Count | Missing Percentage | Imputation Name | Imputation Description |
| plant country Freight (12 month max) | Numeric | 6480 | 83 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 28082 |
| plant country Freight (12 month mean) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 8385.2617 |
| plant country Freight (12 month min) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 21.95 |
| plant country Freight (12 month std) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 12458.879 |
| plant country Freight (1st lag) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 27.2 |
| plant country Freight (24 month max) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 28290 |
| plant country Freight (24 month mean) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 8804.7346 |
| plant country Freight (24 month min) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 20.85 |
| plant country Freight (24 month std) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 12422.334 |
| plant country Freight (6 month max) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 27110 |
| plant country Freight (6 month mean) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 8025.865 |
| plant country Freight (6 month min) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 23.41 |
| plant country Freight (6 month std) | Numeric | 6480 | 83 | Missing Values Imputed | Imputed value: 12891.695 |
| plant country fuel (12 month max) | Numeric | 5175 | 66 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 0.53 |
| plant country fuel (12 month mean) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.4758 |
| plant country fuel (12 month min) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.42 |
| plant country fuel (12 month std) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.0332 |
| plant country fuel (1st lag) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.48 |
| plant country fuel (24 month max) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.55 |
| plant country fuel (24 month mean) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.4771 |
| plant country fuel (24 month min) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.41 |
| plant country fuel (24 month std) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.0331 |
| plant country fuel (6 month max) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.51 |
| plant country fuel (6 month mean) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.4767 |
| plant country fuel (6 month min) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.44 |
| plant country fuel (6 month std) | Numeric | 5175 | 66 | Missing Values Imputed | Imputed value: 0.0299 |
| Number months since series project SOP (12 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 35 |
| Number months since series project SOP (12 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 29.5 |
| Number months since series project SOP (12 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 24 |
| Number months since series project SOP (12 month std) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 3.6056 |
| Number months since series project SOP (24 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 23.5 |
| Number months since series project SOP (24 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 12 |
| Number months since series project SOP (24 month std) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 7.0711 |
| Number months since series project SOP (6 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 32.5 |
| Number months since series project SOP (6 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 30 |
| Number months since series project SOP (6 month std) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1.8708 |
| Number months until series project EOP (12 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1019 |
| Number months until series project EOP (12 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1013.5 |
| Number months until series project EOP (12 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1008 |
| Number months until series project EOP (24 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1031 |
| Number months until series project EOP (24 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1019.5 |
| Number months until series project EOP (6 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1013 |
| Number months until series project EOP (6 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1010.5 |
| plant country fatalities (1st lag) | Numeric | 3870 | 50 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 62 |
| plant country household savings rate (12 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 11.32 |
| plant country household savings rate (12 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 9.9942 |
| plant country household savings rate (12 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 8.37 |
| plant country household savings rate (12 month std) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 0.8496 |
| plant country household savings rate (1st lag) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 9.9 |
| plant country household savings rate (24 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 13.32 |
| plant country household savings rate (24 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 9.4879 |
| plant country household savings rate (24 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 7.28 |
| plant country household savings rate (24 month std) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 1.2606 |
| plant country household savings rate (6 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 10.77 |
| plant country household savings rate (6 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 10.0033 |
| plant country household savings rate (6 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 9.09 |
| plant country household savings rate (6 month std) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 0.5819 |
| plant country registration (12 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 24851 |
| plant country registration (12 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 8622.5825 |
| plant country registration (12 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 12.79 |
| plant country registration (12 month std) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 9686.9672 |
| plant country registration (1st lag) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 9624 |
| plant country registration (6 month max) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 23369 |
| plant country registration (6 month mean) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 9643.4767 |
| plant country registration (6 month min) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 14.51 |
| plant country registration (6 month std) | Numeric | 3870 | 50 | Missing Values Imputed | Imputed value: 9007.6136 |
| plant country fatalities (6 month std) | Numeric | 3867 | 50 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 12.9321 |
| plant country fatalities (6 month max) | Numeric | 3864 | 50 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 79 |
| plant country fatalities (6 month mean) | Numeric | 3864 | 50 | Missing Values Imputed | Imputed value: 64.5 |
| plant country fatalities (6 month min) | Numeric | 3864 | 50 | Missing Values Imputed | Imputed value: 41 |
| plant country fatalities (12 month std) | Numeric | 3834 | 49 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 13.2216 |
| plant country registration (24 month std) | Numeric | 3834 | 49 | Missing Values Imputed | Imputed value: 9272.238 |
| plant country fatalities (12 month max) | Numeric | 3825 | 49 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 79 |
| plant country fatalities (12 month mean) | Numeric | 3825 | 49 | Missing Values Imputed | Imputed value: 63.1667 |
| plant country fatalities (12 month min) | Numeric | 3825 | 49 | Missing Values Imputed | Imputed value: 33 |
| plant country registration (24 month max) | Numeric | 3825 | 49 | Missing Values Imputed | Imputed value: 25980 |
| plant country registration (24 month mean) | Numeric | 3825 | 49 | Missing Values Imputed | Imputed value: 8326.6121 |
| plant country registration (24 month min) | Numeric | 3825 | 49 | Missing Values Imputed | Imputed value: 11.94 |
| plant country fatalities (24 month std) | Numeric | 3726 | 48 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 13.2433 |
| plant country fatalities (24 month max) | Numeric | 3717 | 48 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 95 |
| plant country fatalities (24 month mean) | Numeric | 3717 | 48 | Missing Values Imputed | Imputed value: 64.8333 |
| plant country fatalities (24 month min) | Numeric | 3717 | 48 | Missing Values Imputed | Imputed value: 31 |
| last sales date (Month) (24 month most\_frequent) | Categorical | 2232 | 29 | One-Hot Encoding | Missing indicator treated as feature |
| last sales date (Month) (12 month most\_frequent) | Categorical | 1785 | 23 | One-Hot Encoding | Missing indicator treated as feature |
| last sales date (Day of Week) (24 month most\_frequent) | Categorical | 1704 | 22 | One-Hot Encoding | Missing indicator treated as feature |
| last sales date (Day of Week) (12 month most\_frequent) | Categorical | 1555 | 20 | One-Hot Encoding | Missing indicator treated as feature |
| last sales date (Month) (6 month most\_frequent) | Categorical | 1523 | 20 | One-Hot Encoding | Missing indicator treated as feature |
| last sales date (Day of Week) (6 month most\_frequent) | Categorical | 1379 | 18 | One-Hot Encoding | Missing indicator treated as feature |
| last sales date (Day of Week) (1st lag) | Categorical | 1195 | 15 | One-Hot Encoding | Missing indicator treated as feature |
| last sales date (Month) (1st lag) | Categorical | 1195 | 15 | One-Hot Encoding | Missing indicator treated as feature |
| Number months since last sale (1st lag) | Numeric | 1195 | 15 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 1 |
| last sales date (Day of Month) (1st lag) | Numeric | 1195 | 15 | Missing Values Imputed | Imputed value: 0 |
| last sales date (Year) (1st lag) | Numeric | 1195 | 15 | Missing Values Imputed | Imputed value: 2015 |
| revenue last 12 months (1st lag) | Numeric | 1195 | 15 | Missing Values Imputed | Imputed value: 760084.12 |
| Number months since last sale (6 month std) | Numeric | 839 | 11 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 0 |
| last sales date (Year) (6 month std) | Numeric | 839 | 11 | Missing Values Imputed | Imputed value: 0 |
| revenue last 12 months (6 month std) | Numeric | 839 | 11 | Missing Values Imputed | Imputed value: 30439.35 |
| Number months since last sale (12 month std) | Numeric | 666 | 9 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 0 |
| last sales date (Year) (12 month std) | Numeric | 666 | 9 | Missing Values Imputed | Imputed value: 0.4523 |
| revenue last 12 months (12 month std) | Numeric | 666 | 9 | Missing Values Imputed | Imputed value: 49639.674 |
| Number months since last sale (6 month max) | Numeric | 525 | 7 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 1 |
| Number months since last sale (6 month mean) | Numeric | 525 | 7 | Missing Values Imputed | Imputed value: 1 |
| Number months since last sale (6 month min) | Numeric | 525 | 7 | Missing Values Imputed | Imputed value: 1 |
| last sales date (Year) (6 month max) | Numeric | 525 | 7 | Missing Values Imputed | Imputed value: 2015 |
| last sales date (Year) (6 month mean) | Numeric | 525 | 7 | Missing Values Imputed | Imputed value: 2015 |
| last sales date (Year) (6 month min) | Numeric | 525 | 7 | Missing Values Imputed | Imputed value: 2015 |
| revenue last 12 months (6 month max) | Numeric | 525 | 7 | Missing Values Imputed | Imputed value: 696487.48 |
| revenue last 12 months (6 month mean) | Numeric | 525 | 7 | Missing Values Imputed | Imputed value: 589826.08 |
| revenue last 12 months (6 month min) | Numeric | 525 | 7 | Missing Values Imputed | Imputed value: 519909.17 |
| Number months since last sale (24 month std) | Numeric | 432 | 6 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 0 |
| last sales date (Year) (24 month std) | Numeric | 432 | 6 | Missing Values Imputed | Imputed value: 0.669 |
| revenue last 12 months (24 month std) | Numeric | 432 | 6 | Missing Values Imputed | Imputed value: 91551.93 |
| Number months since last sale (12 month max) | Numeric | 373 | 5 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 1 |
| Number months since last sale (12 month mean) | Numeric | 373 | 5 | Missing Values Imputed | Imputed value: 1 |
| Number months since last sale (12 month min) | Numeric | 373 | 5 | Missing Values Imputed | Imputed value: 1 |
| last sales date (Year) (12 month max) | Numeric | 373 | 5 | Missing Values Imputed | Imputed value: 2015 |
| last sales date (Year) (12 month mean) | Numeric | 373 | 5 | Missing Values Imputed | Imputed value: 2014.75 |
| last sales date (Year) (12 month min) | Numeric | 373 | 5 | Missing Values Imputed | Imputed value: 2014 |
| revenue last 12 months (12 month max) | Numeric | 373 | 5 | Missing Values Imputed | Imputed value: 715463.42 |
| revenue last 12 months (12 month mean) | Numeric | 373 | 5 | Missing Values Imputed | Imputed value: 541616.29 |
| revenue last 12 months (12 month min) | Numeric | 373 | 5 | Missing Values Imputed | Imputed value: 359546.16 |
| Number months since last sale (24 month max) | Numeric | 192 | 2 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 1 |
| Number months since last sale (24 month mean) | Numeric | 192 | 2 | Missing Values Imputed | Imputed value: 1 |
| Number months since last sale (24 month min) | Numeric | 192 | 2 | Missing Values Imputed | Imputed value: 1 |
| last sales date (Year) (24 month max) | Numeric | 192 | 2 | Missing Values Imputed | Imputed value: 2015 |
| last sales date (Year) (24 month mean) | Numeric | 192 | 2 | Missing Values Imputed | Imputed value: 2014.4 |
| last sales date (Year) (24 month min) | Numeric | 192 | 2 | Missing Values Imputed | Imputed value: 2013 |
| revenue last 12 months (24 month max) | Numeric | 192 | 2 | Missing Values Imputed | Imputed value: 846237 |
| revenue last 12 months (24 month mean) | Numeric | 192 | 2 | Missing Values Imputed | Imputed value: 489768.52 |
| revenue last 12 months (24 month min) | Numeric | 192 | 2 | Missing Values Imputed | Imputed value: 167043.55 |
| total net eur (5th lag) (diff 24 month mean) | Numeric | 111 | 1 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: -333.7062 |
| total net eur (4th lag) (diff 24 month mean) | Numeric | 100 | 1 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: -348.0187 |
| total net eur (3rd lag) (diff 24 month mean) | Numeric | 90 | 1 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: -362.9417 |
| Number months since first sale (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 3.6056 |
| Number months since first sale (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 7.0711 |
| Number months since first sale (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 1.8708 |
| accumulated revenue (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 214295.66 |
| accumulated revenue (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 472165.44 |
| accumulated revenue (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 95434.531 |
| number months with revenue (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 3.6056 |
| number months with revenue (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 7.0711 |
| number months with revenue (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 2.1602 |
| plant country Business situation (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 5.7616 |
| plant country Business situation (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 7.2325 |
| plant country Business situation (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 4.4982 |
| plant country Demand evolution (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 6.0577 |
| plant country Demand evolution (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 8.1146 |
| plant country Demand evolution (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 4.4373 |
| plant country Finished good stocks (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 2.8276 |
| plant country Finished good stocks (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 3.6233 |
| plant country Finished good stocks (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 1.9552 |
| plant country Order books (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 5.4013 |
| plant country Order books (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 7.7277 |
| plant country Order books (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 3.5982 |
| plant country Production (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 9.0843 |
| plant country Production (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 11.1709 |
| plant country Production (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 6.7885 |
| plant country consumer price index (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.4019 |
| plant country consumer price index (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.5664 |
| plant country consumer price index (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.3089 |
| plant country consumer prices (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 6.0032 |
| plant country consumer prices (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 8.1171 |
| plant country consumer prices (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 4.3664 |
| plant country economic situation (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 5.5794 |
| plant country economic situation (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 8.3916 |
| plant country economic situation (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 4.2824 |
| plant country selling prices (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 6.5707 |
| plant country selling prices (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 7.6883 |
| plant country selling prices (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 5.1381 |
| plant country short\_term interest rate (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.0586 |
| plant country short\_term interest rate (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.1227 |
| plant country short\_term interest rate (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.0306 |
| plant country unemployment rate (12 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.3919 |
| plant country unemployment rate (24 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.7235 |
| plant country unemployment rate (6 month std) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: 0.1966 |
| total net eur (2nd lag) (diff 24 month mean) | Numeric | 81 | 1 | Missing Values Imputed | Imputed value: -328.2375 |
| %key\_company (1st lag) | Categorical | 72 | 1 | One-Hot Encoding | Missing indicator treated as feature |
| First sale (1st lag) | Categorical | 72 | 1 | One-Hot Encoding | Missing indicator treated as feature |
| First sales date (Day of Week) (12 month most\_frequent) | Categorical | 72 | 1 | One-Hot Encoding | Missing indicator treated as feature |
| First sales date (Month) (12 month most\_frequent) | Categorical | 72 | 1 | One-Hot Encoding | Missing indicator treated as feature |
| covid (1st lag) | Categorical | 72 | 1 | One-Hot Encoding | Missing indicator treated as feature |
| plant country (1st lag) | Categorical | 72 | 1 | One-Hot Encoding | Missing indicator treated as feature |
| platform no\_ (1st lag) | Categorical | 72 | 1 | One-Hot Encoding | Missing indicator treated as feature |
| First sale (12 month fraction equal yes) | Numeric | 72 | 1 | Missing Values Imputed | Missing indicator treated as feature, Imputed value: 0 |
| First sale (24 month fraction equal yes) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0 |
| First sale (6 month fraction equal yes) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0 |
| First sales date (Day of Week) (12 month entropy) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -1 |
| First sales date (Year) (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 2005 |
| Number months since first sale (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 108 |
| Number months since first sale (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 102.5 |
| Number months since first sale (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 97 |
| Number months since first sale (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 96.5 |
| Number months since first sale (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 85 |
| Number months since first sale (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 105.5 |
| Number months since first sale (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 103 |
| accumulated revenue (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 6220430.7 |
| accumulated revenue (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 4484770.7 |
| accumulated revenue (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 3485633.8 |
| accumulated revenue (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 4971102.7 |
| accumulated revenue (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 6690565.7 |
| accumulated revenue (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 4060527 |
| accumulated revenue (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1848235.5 |
| accumulated revenue (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 5781576.8 |
| accumulated revenue (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 4647517.1 |
| accumulated revenue (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 4144698.3 |
| covid (12 month fraction equal yes) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0 |
| covid (24 month fraction equal yes) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0 |
| covid (6 month fraction equal yes) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0 |
| last sales date (Day of Month) (12 month fraction equal 1\_0) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1 |
| last sales date (Day of Month) (24 month fraction equal 1\_0) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1 |
| last sales date (Day of Month) (6 month fraction equal 1\_0) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1 |
| last sales date (Day of Week) (12 month entropy) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.8637 |
| last sales date (Day of Week) (24 month entropy) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.9251 |
| last sales date (Day of Week) (6 month entropy) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.5607 |
| last sales date (Month) (12 month entropy) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 2.4849 |
| last sales date (Month) (24 month entropy) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 2.4849 |
| last sales date (Month) (6 month entropy) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.7918 |
| number months with revenue (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 91 |
| number months with revenue (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 71.5 |
| number months with revenue (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 64 |
| number months with revenue (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 81 |
| number months with revenue (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 93 |
| number months with revenue (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 65 |
| number months with revenue (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 47 |
| number months with revenue (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 88 |
| number months with revenue (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 74.1667 |
| number months with revenue (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 70 |
| plant country Business situation (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 10.6 |
| plant country Business situation (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 6.4 |
| plant country Business situation (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1 |
| plant country Business situation (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 6.4 |
| plant country Business situation (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 16.9 |
| plant country Business situation (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 5.9333 |
| plant country Business situation (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -4.7 |
| plant country Business situation (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9.3 |
| plant country Business situation (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 6.5167 |
| plant country Business situation (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 2.6 |
| plant country Demand evolution (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 26.4 |
| plant country Demand evolution (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 13.575 |
| plant country Demand evolution (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.1 |
| plant country Demand evolution (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 15.3 |
| plant country Demand evolution (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 28.3 |
| plant country Demand evolution (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 13.1375 |
| plant country Demand evolution (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -6.5 |
| plant country Demand evolution (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 23 |
| plant country Demand evolution (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 14.6667 |
| plant country Demand evolution (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 6.7 |
| plant country Finished good stocks (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 4.7 |
| plant country Finished good stocks (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0.8333 |
| plant country Finished good stocks (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -3.6 |
| plant country Finished good stocks (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1 |
| plant country Finished good stocks (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 7.5 |
| plant country Finished good stocks (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.0042 |
| plant country Finished good stocks (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -6 |
| plant country Finished good stocks (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 3.5 |
| plant country Finished good stocks (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0.8571 |
| plant country Finished good stocks (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -1.9 |
| plant country Order books (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -5 |
| plant country Order books (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -12.5917 |
| plant country Order books (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -21 |
| plant country Order books (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -11 |
| plant country Order books (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -2 |
| plant country Order books (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -16.4583 |
| plant country Order books (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -32 |
| plant country Order books (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -6.9 |
| plant country Order books (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -11.5714 |
| plant country Order books (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -17 |
| plant country Production (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 22.5 |
| plant country Production (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9.5083 |
| plant country Production (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -5.1 |
| plant country Production (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9 |
| plant country Production (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 26.6 |
| plant country Production (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9.5708 |
| plant country Production (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -12.9 |
| plant country Production (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 19.2 |
| plant country Production (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9.1667 |
| plant country Production (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.1 |
| plant country consumer price index (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.9597 |
| plant country consumer price index (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.1738 |
| plant country consumer price index (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0.4027 |
| plant country consumer price index (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.3309 |
| plant country consumer price index (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 2.436 |
| plant country consumer price index (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.1575 |
| plant country consumer price index (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -0.0998 |
| plant country consumer price index (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.6162 |
| plant country consumer price index (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.2511 |
| plant country consumer price index (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0.7374 |
| plant country consumer prices (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 37.8 |
| plant country consumer prices (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 26.9917 |
| plant country consumer prices (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 13.7 |
| plant country consumer prices (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 27.9 |
| plant country consumer prices (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 45.4 |
| plant country consumer prices (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 24.5875 |
| plant country consumer prices (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 5.1 |
| plant country consumer prices (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 34.3 |
| plant country consumer prices (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 27.8333 |
| plant country consumer prices (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 21.4 |
| plant country economic situation (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -6.9 |
| plant country economic situation (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -17.1333 |
| plant country economic situation (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -27.2 |
| plant country economic situation (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -16 |
| plant country economic situation (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -5.9 |
| plant country economic situation (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -19.0625 |
| plant country economic situation (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -42.1 |
| plant country economic situation (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -10.1 |
| plant country economic situation (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -15.6286 |
| plant country economic situation (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -22 |
| plant country selling prices (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 11 |
| plant country selling prices (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.4583 |
| plant country selling prices (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -10 |
| plant country selling prices (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 3 |
| plant country selling prices (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 15.3 |
| plant country selling prices (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.125 |
| plant country selling prices (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -16 |
| plant country selling prices (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9 |
| plant country selling prices (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 1.8333 |
| plant country selling prices (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -5 |
| plant country short\_term interest rate (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0.0482 |
| plant country short\_term interest rate (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -0.0368 |
| plant country short\_term interest rate (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -0.1462 |
| plant country short\_term interest rate (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -0.1462 |
| plant country short\_term interest rate (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0.3298 |
| plant country short\_term interest rate (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 0.077 |
| plant country short\_term interest rate (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -0.1462 |
| plant country short\_term interest rate (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -0.0277 |
| plant country short\_term interest rate (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -0.0797 |
| plant country short\_term interest rate (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -0.1462 |
| plant country unemployment rate (12 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 10.8 |
| plant country unemployment rate (12 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9.825 |
| plant country unemployment rate (12 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9 |
| plant country unemployment rate (1st lag) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9 |
| plant country unemployment rate (24 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 12.3 |
| plant country unemployment rate (24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 10.5333 |
| plant country unemployment rate (24 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 8.7 |
| plant country unemployment rate (6 month max) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9.7 |
| plant country unemployment rate (6 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9.3667 |
| plant country unemployment rate (6 month min) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 9 |
| total net eur (12 month max) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 13083.538 |
| total net eur (12 month mean) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -30.5333 |
| total net eur (12 month median) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -412.2729 |
| total net eur (12 month min) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -15683.624 |
| total net eur (1st lag) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -330.9517 |
| total net eur (24 month max) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 34010.936 |
| total net eur (24 month median) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -271.6979 |
| total net eur (24 month min) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -22844.609 |
| total net eur (6 month max) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: 3955.8146 |
| total net eur (6 month mean) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -77.8417 |
| total net eur (6 month median) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -345.3542 |
| total net eur (6 month min) (diff 24 month mean) | Numeric | 72 | 1 | Missing Values Imputed | Imputed value: -6755.42 |
| sales date (Day of Week) (actual) | Categorical | 0 | 0 | One-Hot Encoding | Missing values ignored |
| sales date (Month) (actual) | Categorical | 0 | 0 | One-Hot Encoding | Missing values ignored |
| sales date (Year) (actual) | Numeric | 0 | 0 | Missing Values Imputed | Imputed value: 2016 |

6 Model Performance and Stability

6.1 Model Validation Stability

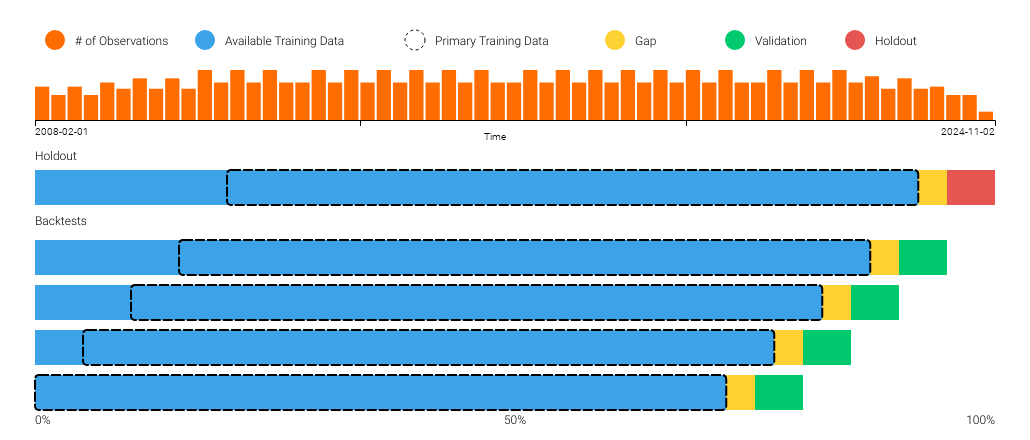
To find patterns in a dataset from which it can make predictions, an algorithm must first learn from a historical example – typically from a historical dataset that contains the output variable you want to predict. However, if a model is trained too closely on its training data then it may be overfit. 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.

Out-of-time validation (OTV) allows the selection of specific time periods or durations to test the stability of your model rather than using random rows, creating “backtests” for your data. OTV ensures that there is no overfitting on prior periods of data on which the model was trained. Measuring backtest performance is a good way to assess model performance on new observations. In addition to the validation length partitioning (which is the length of time used for validation per each backtest), users can also adjust the gap length or the time between training and validation. In addition to the OTV 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:

* DataRobot used 4 backtests with a validation length of 10 months.
* A holdout fold with start date: 2024-01-02 and end date 2024-11-02 for additional testing. This dataset is used to verify that the final model performs well on data that has not been touched throughout the training process.

The following figure summarizes the CV process used by DataRobot, where the blue denotes available training data, green denotes the validation partition, and red denotes the holdout sample.



6.1.1 Data Partitioning Methodology

Time series projects, like OTV projects, use date/time partitioning, and all the workflow changes that apply to other date/time partitioned projects also apply to them. Unlike other projects, time series projects produce different types of models which forecast multiple future predictions instead of an individual prediction for each row.

DataRobot uses a general time series framework to configure how time series features are created and what future values the models will output. This framework consists of a Forecast Point (defining a time a prediction is being made), a Feature Derivation Window (a rolling window used to create features), and a Forecast Window (a rolling window of future values to predict).

Time series projects will automatically transform the dataset provided in order to apply this framework. During the transformation, DataRobot uses the Feature Derivation Window to derive time series features (such as lags and rolling statistics), and uses the Forecast Window to provide examples of forecasting different distances in the future (such as time shifts). After project creation, a new dataset and a new feature list are generated and used to train the models. This process is reapplied automatically at prediction time as well in order to generate future predictions based on the original data features.

6.2 Model Performance (Backtesting)

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

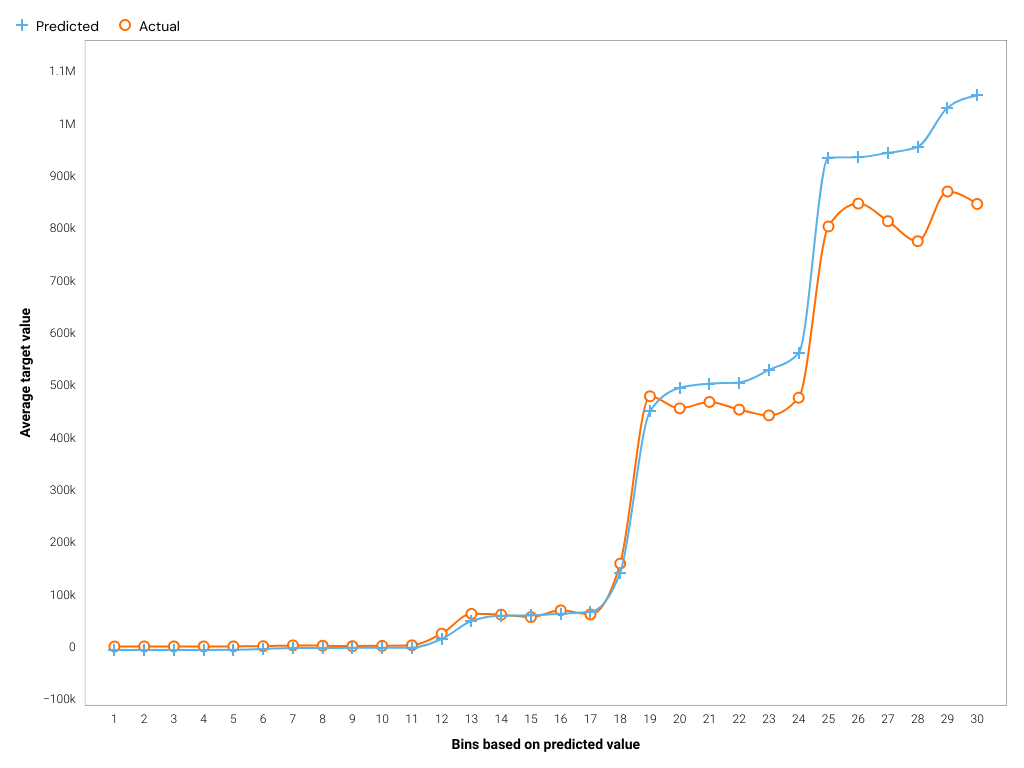
The performance metric used for this project was RMSE. The model performance results are presented below for out-of-time testing:

|  |  |
| --- | --- |
| Scoring Type | Score (RMSE) |
| backtesting\_scores | 93107.9069, 86615.7141, 50781.5342, 135827.2314 |
| holdout | 119972.9334 |
| validation | 93107.9069 |

6.3 Sensitivity Testing and Analysis

6.3.1 Lift Chart

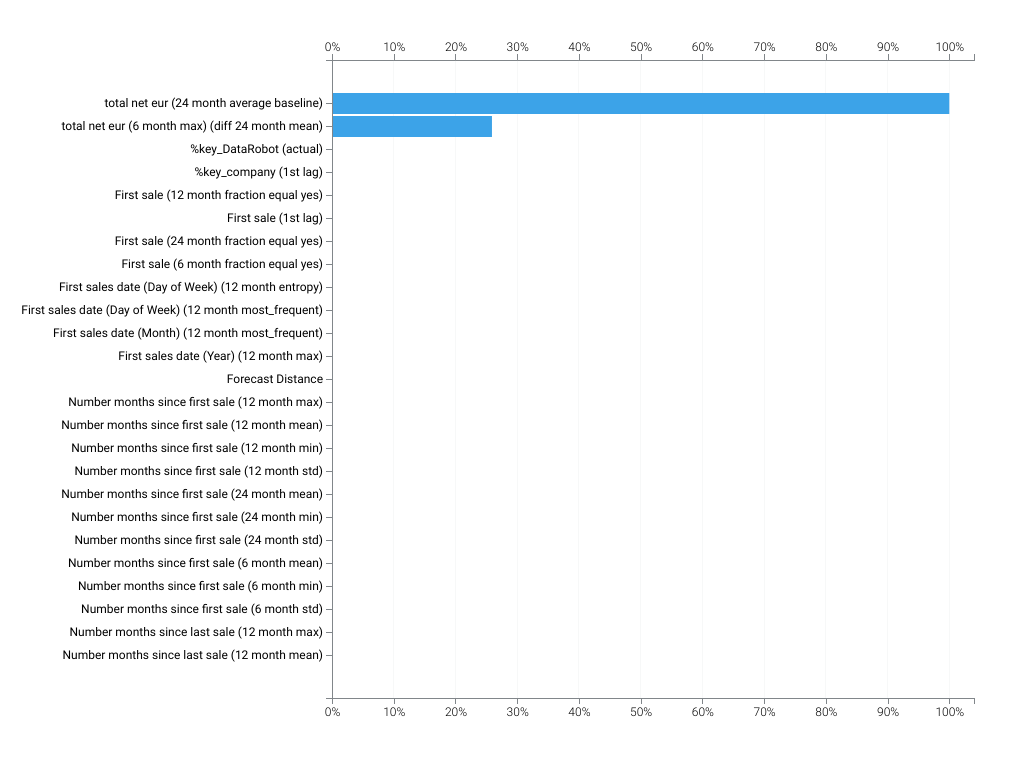
The Lift Chart sorts and groups numeric feature values into equal sized bins, depicting how well a model segments the target population and how capable it is of predicting the target. This helps the user to visualize model accuracy for each bin. The chart is sorted by predicted values -- lowest to highest predictions, for example -- which provides transparency to the model performance for different ranges of values of the target variable. Looking at the Lift Chart, the left side of the curve indicates where the model predicted a low score on one section of the population while the right side of the curve indicates where the model predicted a high score. The model Lift Chart is presented in the figure below.



The points on the Lift Chart indicate the average percentage in each bin. The "Predicted" blue line displays the average prediction score for the rows in that bin. The "Actual" orange line displays the actual percentage for the rows in that bin. In general, the steeper the Actual line is, and the more closely the Predicted line matches the actual line, the better the model. A close relationship between these two lines is indicative of the predictive accuracy of the model; a consistently increasing line is another good indicator of satisfactory model performance.

6.3.2 Key Relationships

Feature Impact, which is available for all model types, works by altering input data and observing the effect on a models score. This technique is sometimes called Permutation Importance. The Feature Impact for a given column measures how much worse a models error score would be if DataRobot made predictions after randomly shuffling that column (while leaving other columns unchanged). DataRobot normalizes the scores so that the value of the most important feature column is first and the other subsequent features are normalized to it.

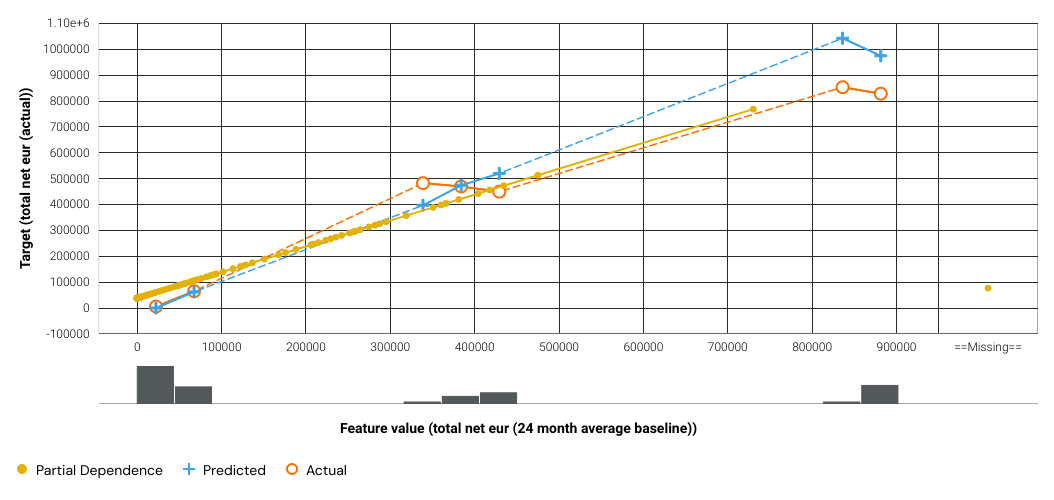


|  |  |  |
| --- | --- | --- |
| Feature Name | Impact Normalized | Impact Unnormalized |
| total net eur (24 month average baseline) | 1.0 | 133463.7513 |
| total net eur (6 month max) (diff 24 month mean) | 0.2591 | 34582.4803 |
| %key\_DataRobot (actual) | 0.0 | 0.0 |
| %key\_company (1st lag) | 0.0 | 0.0 |
| First sale (12 month fraction equal yes) | 0.0 | 0.0 |
| First sale (1st lag) | 0.0 | 0.0 |
| First sale (24 month fraction equal yes) | 0.0 | 0.0 |
| First sale (6 month fraction equal yes) | 0.0 | 0.0 |
| First sales date (Day of Week) (12 month entropy) | 0.0 | 0.0 |
| First sales date (Day of Week) (12 month most\_frequent) | 0.0 | 0.0 |
| First sales date (Month) (12 month most\_frequent) | 0.0 | 0.0 |
| First sales date (Year) (12 month max) | 0.0 | 0.0 |
| Number months since first sale (12 month max) | 0.0 | 0.0 |
| Number months since first sale (12 month mean) | 0.0 | 0.0 |
| Number months since first sale (12 month min) | 0.0 | 0.0 |
| Number months since first sale (12 month std) | 0.0 | 0.0 |
| Number months since first sale (24 month mean) | 0.0 | 0.0 |
| Number months since first sale (24 month min) | 0.0 | 0.0 |
| Number months since first sale (24 month std) | 0.0 | 0.0 |
| Number months since first sale (6 month mean) | 0.0 | 0.0 |
| Number months since first sale (6 month min) | 0.0 | 0.0 |
| Number months since first sale (6 month std) | 0.0 | 0.0 |
| Number months since last sale (12 month max) | 0.0 | 0.0 |
| Number months since last sale (12 month mean) | 0.0 | 0.0 |
| Number months since last sale (12 month min) | 0.0 | 0.0 |
| Number months since last sale (12 month std) | 0.0 | 0.0 |
| Number months since last sale (1st lag) | 0.0 | 0.0 |
| Number months since last sale (24 month max) | 0.0 | 0.0 |
| Number months since last sale (24 month mean) | 0.0 | 0.0 |
| Number months since last sale (24 month min) | 0.0 | 0.0 |
| Number months since last sale (24 month std) | 0.0 | 0.0 |
| Number months since last sale (6 month max) | 0.0 | 0.0 |
| Number months since last sale (6 month mean) | 0.0 | 0.0 |
| Number months since last sale (6 month min) | 0.0 | 0.0 |
| Number months since last sale (6 month std) | 0.0 | 0.0 |
| Number months since series project SOP (12 month max) | 0.0 | 0.0 |
| Number months since series project SOP (12 month mean) | 0.0 | 0.0 |
| Number months since series project SOP (12 month min) | 0.0 | 0.0 |
| Number months since series project SOP (12 month std) | 0.0 | 0.0 |
| Number months since series project SOP (24 month mean) | 0.0 | 0.0 |
| Number months since series project SOP (24 month min) | 0.0 | 0.0 |
| Number months since series project SOP (24 month std) | 0.0 | 0.0 |
| Number months since series project SOP (6 month mean) | 0.0 | 0.0 |
| Number months since series project SOP (6 month min) | 0.0 | 0.0 |
| Number months since series project SOP (6 month std) | 0.0 | 0.0 |
| Number months until series project EOP (12 month max) | 0.0 | 0.0 |
| Number months until series project EOP (12 month mean) | 0.0 | 0.0 |
| Number months until series project EOP (12 month min) | 0.0 | 0.0 |
| Number months until series project EOP (24 month max) | 0.0 | 0.0 |
| Number months until series project EOP (24 month mean) | 0.0 | 0.0 |
| Number months until series project EOP (6 month max) | 0.0 | 0.0 |
| Number months until series project EOP (6 month mean) | 0.0 | 0.0 |
| accumulated revenue (12 month max) | 0.0 | 0.0 |
| accumulated revenue (12 month mean) | 0.0 | 0.0 |
| accumulated revenue (12 month min) | 0.0 | 0.0 |
| accumulated revenue (12 month std) | 0.0 | 0.0 |
| accumulated revenue (1st lag) | 0.0 | 0.0 |
| accumulated revenue (24 month max) | 0.0 | 0.0 |
| accumulated revenue (24 month mean) | 0.0 | 0.0 |
| accumulated revenue (24 month min) | 0.0 | 0.0 |
| accumulated revenue (24 month std) | 0.0 | 0.0 |
| accumulated revenue (6 month max) | 0.0 | 0.0 |
| accumulated revenue (6 month mean) | 0.0 | 0.0 |
| accumulated revenue (6 month min) | 0.0 | 0.0 |
| accumulated revenue (6 month std) | 0.0 | 0.0 |
| covid (12 month fraction equal yes) | 0.0 | 0.0 |
| covid (1st lag) | 0.0 | 0.0 |
| covid (24 month fraction equal yes) | 0.0 | 0.0 |
| covid (6 month fraction equal yes) | 0.0 | 0.0 |
| last sales date (Day of Month) (12 month fraction equal 1\_0) | 0.0 | 0.0 |
| last sales date (Day of Month) (1st lag) | 0.0 | 0.0 |
| last sales date (Day of Month) (24 month fraction equal 1\_0) | 0.0 | 0.0 |
| last sales date (Day of Month) (6 month fraction equal 1\_0) | 0.0 | 0.0 |
| last sales date (Day of Week) (12 month entropy) | 0.0 | 0.0 |
| last sales date (Day of Week) (12 month most\_frequent) | 0.0 | 0.0 |
| last sales date (Day of Week) (1st lag) | 0.0 | 0.0 |
| last sales date (Day of Week) (24 month entropy) | 0.0 | 0.0 |
| last sales date (Day of Week) (24 month most\_frequent) | 0.0 | 0.0 |
| last sales date (Day of Week) (6 month entropy) | 0.0 | 0.0 |
| last sales date (Day of Week) (6 month most\_frequent) | 0.0 | 0.0 |
| last sales date (Month) (12 month entropy) | 0.0 | 0.0 |
| last sales date (Month) (12 month most\_frequent) | 0.0 | 0.0 |
| last sales date (Month) (1st lag) | 0.0 | 0.0 |
| last sales date (Month) (24 month entropy) | 0.0 | 0.0 |
| last sales date (Month) (24 month most\_frequent) | 0.0 | 0.0 |
| last sales date (Month) (6 month entropy) | 0.0 | 0.0 |
| last sales date (Month) (6 month most\_frequent) | 0.0 | 0.0 |
| last sales date (Year) (12 month max) | 0.0 | 0.0 |
| last sales date (Year) (12 month mean) | 0.0 | 0.0 |
| last sales date (Year) (12 month min) | 0.0 | 0.0 |
| last sales date (Year) (12 month std) | 0.0 | 0.0 |
| last sales date (Year) (1st lag) | 0.0 | 0.0 |
| last sales date (Year) (24 month max) | 0.0 | 0.0 |
| last sales date (Year) (24 month mean) | 0.0 | 0.0 |
| last sales date (Year) (24 month min) | 0.0 | 0.0 |
| last sales date (Year) (24 month std) | 0.0 | 0.0 |
| last sales date (Year) (6 month max) | 0.0 | 0.0 |
| last sales date (Year) (6 month mean) | 0.0 | 0.0 |
| last sales date (Year) (6 month min) | 0.0 | 0.0 |
| last sales date (Year) (6 month std) | 0.0 | 0.0 |
| number months with revenue (12 month max) | 0.0 | 0.0 |
| number months with revenue (12 month mean) | 0.0 | 0.0 |
| number months with revenue (12 month min) | 0.0 | 0.0 |
| number months with revenue (12 month std) | 0.0 | 0.0 |
| number months with revenue (1st lag) | 0.0 | 0.0 |
| number months with revenue (24 month max) | 0.0 | 0.0 |
| number months with revenue (24 month mean) | 0.0 | 0.0 |
| number months with revenue (24 month min) | 0.0 | 0.0 |
| number months with revenue (24 month std) | 0.0 | 0.0 |
| number months with revenue (6 month max) | 0.0 | 0.0 |
| number months with revenue (6 month mean) | 0.0 | 0.0 |
| number months with revenue (6 month min) | 0.0 | 0.0 |
| number months with revenue (6 month std) | 0.0 | 0.0 |
| plant country (1st lag) | 0.0 | 0.0 |
| plant country Business situation (12 month max) | 0.0 | 0.0 |
| plant country Business situation (12 month mean) | 0.0 | 0.0 |
| plant country Business situation (12 month min) | 0.0 | 0.0 |
| plant country Business situation (12 month std) | 0.0 | 0.0 |
| plant country Business situation (1st lag) | 0.0 | 0.0 |
| plant country Business situation (24 month max) | 0.0 | 0.0 |
| plant country Business situation (24 month mean) | 0.0 | 0.0 |
| plant country Business situation (24 month min) | 0.0 | 0.0 |
| plant country Business situation (24 month std) | 0.0 | 0.0 |
| plant country Business situation (6 month max) | 0.0 | 0.0 |
| plant country Business situation (6 month mean) | 0.0 | 0.0 |
| plant country Business situation (6 month min) | 0.0 | 0.0 |
| plant country Business situation (6 month std) | 0.0 | 0.0 |
| plant country Demand evolution (12 month max) | 0.0 | 0.0 |
| plant country Demand evolution (12 month mean) | 0.0 | 0.0 |
| plant country Demand evolution (12 month min) | 0.0 | 0.0 |
| plant country Demand evolution (12 month std) | 0.0 | 0.0 |
| plant country Demand evolution (1st lag) | 0.0 | 0.0 |
| plant country Demand evolution (24 month max) | 0.0 | 0.0 |
| plant country Demand evolution (24 month mean) | 0.0 | 0.0 |
| plant country Demand evolution (24 month min) | 0.0 | 0.0 |
| plant country Demand evolution (24 month std) | 0.0 | 0.0 |
| plant country Demand evolution (6 month max) | 0.0 | 0.0 |
| plant country Demand evolution (6 month mean) | 0.0 | 0.0 |
| plant country Demand evolution (6 month min) | 0.0 | 0.0 |
| plant country Demand evolution (6 month std) | 0.0 | 0.0 |
| plant country Finished good stocks (12 month max) | 0.0 | 0.0 |
| plant country Finished good stocks (12 month mean) | 0.0 | 0.0 |
| plant country Finished good stocks (12 month min) | 0.0 | 0.0 |
| plant country Finished good stocks (12 month std) | 0.0 | 0.0 |
| plant country Finished good stocks (1st lag) | 0.0 | 0.0 |
| plant country Finished good stocks (24 month max) | 0.0 | 0.0 |
| plant country Finished good stocks (24 month mean) | 0.0 | 0.0 |
| plant country Finished good stocks (24 month min) | 0.0 | 0.0 |
| plant country Finished good stocks (24 month std) | 0.0 | 0.0 |
| plant country Finished good stocks (6 month max) | 0.0 | 0.0 |
| plant country Finished good stocks (6 month mean) | 0.0 | 0.0 |
| plant country Finished good stocks (6 month min) | 0.0 | 0.0 |
| plant country Finished good stocks (6 month std) | 0.0 | 0.0 |
| plant country Freight (12 month max) | 0.0 | 0.0 |
| plant country Freight (12 month mean) | 0.0 | 0.0 |
| plant country Freight (12 month min) | 0.0 | 0.0 |
| plant country Freight (12 month std) | 0.0 | 0.0 |
| plant country Freight (1st lag) | 0.0 | 0.0 |
| plant country Freight (24 month max) | 0.0 | 0.0 |
| plant country Freight (24 month mean) | 0.0 | 0.0 |
| plant country Freight (24 month min) | 0.0 | 0.0 |
| plant country Freight (24 month std) | 0.0 | 0.0 |
| plant country Freight (6 month max) | 0.0 | 0.0 |
| plant country Freight (6 month mean) | 0.0 | 0.0 |
| plant country Freight (6 month min) | 0.0 | 0.0 |
| plant country Freight (6 month std) | 0.0 | 0.0 |
| plant country Order books (12 month max) | 0.0 | 0.0 |
| plant country Order books (12 month mean) | 0.0 | 0.0 |
| plant country Order books (12 month min) | 0.0 | 0.0 |
| plant country Order books (12 month std) | 0.0 | 0.0 |
| plant country Order books (1st lag) | 0.0 | 0.0 |
| plant country Order books (24 month max) | 0.0 | 0.0 |
| plant country Order books (24 month mean) | 0.0 | 0.0 |
| plant country Order books (24 month min) | 0.0 | 0.0 |
| plant country Order books (24 month std) | 0.0 | 0.0 |
| plant country Order books (6 month max) | 0.0 | 0.0 |
| plant country Order books (6 month mean) | 0.0 | 0.0 |
| plant country Order books (6 month min) | 0.0 | 0.0 |
| plant country Order books (6 month std) | 0.0 | 0.0 |
| plant country Production (12 month max) | 0.0 | 0.0 |
| plant country Production (12 month mean) | 0.0 | 0.0 |
| plant country Production (12 month min) | 0.0 | 0.0 |
| plant country Production (12 month std) | 0.0 | 0.0 |
| plant country Production (1st lag) | 0.0 | 0.0 |
| plant country Production (24 month max) | 0.0 | 0.0 |
| plant country Production (24 month mean) | 0.0 | 0.0 |
| plant country Production (24 month min) | 0.0 | 0.0 |
| plant country Production (24 month std) | 0.0 | 0.0 |
| plant country Production (6 month max) | 0.0 | 0.0 |
| plant country Production (6 month mean) | 0.0 | 0.0 |
| plant country Production (6 month min) | 0.0 | 0.0 |
| plant country Production (6 month std) | 0.0 | 0.0 |
| plant country consumer price index (12 month max) | 0.0 | 0.0 |
| plant country consumer price index (12 month mean) | 0.0 | 0.0 |
| plant country consumer price index (12 month min) | 0.0 | 0.0 |
| plant country consumer price index (12 month std) | 0.0 | 0.0 |
| plant country consumer price index (1st lag) | 0.0 | 0.0 |
| plant country consumer price index (24 month max) | 0.0 | 0.0 |
| plant country consumer price index (24 month mean) | 0.0 | 0.0 |
| plant country consumer price index (24 month min) | 0.0 | 0.0 |
| plant country consumer price index (24 month std) | 0.0 | 0.0 |
| plant country consumer price index (6 month max) | 0.0 | 0.0 |
| plant country consumer price index (6 month mean) | 0.0 | 0.0 |
| plant country consumer price index (6 month min) | 0.0 | 0.0 |
| plant country consumer price index (6 month std) | 0.0 | 0.0 |
| plant country consumer prices (12 month max) | 0.0 | 0.0 |
| plant country consumer prices (12 month mean) | 0.0 | 0.0 |
| plant country consumer prices (12 month min) | 0.0 | 0.0 |
| plant country consumer prices (12 month std) | 0.0 | 0.0 |
| plant country consumer prices (1st lag) | 0.0 | 0.0 |
| plant country consumer prices (24 month max) | 0.0 | 0.0 |
| plant country consumer prices (24 month mean) | 0.0 | 0.0 |
| plant country consumer prices (24 month min) | 0.0 | 0.0 |
| plant country consumer prices (24 month std) | 0.0 | 0.0 |
| plant country consumer prices (6 month max) | 0.0 | 0.0 |
| plant country consumer prices (6 month mean) | 0.0 | 0.0 |
| plant country consumer prices (6 month min) | 0.0 | 0.0 |
| plant country consumer prices (6 month std) | 0.0 | 0.0 |
| plant country economic situation (12 month max) | 0.0 | 0.0 |
| plant country economic situation (12 month mean) | 0.0 | 0.0 |
| plant country economic situation (12 month min) | 0.0 | 0.0 |
| plant country economic situation (12 month std) | 0.0 | 0.0 |
| plant country economic situation (1st lag) | 0.0 | 0.0 |
| plant country economic situation (24 month max) | 0.0 | 0.0 |
| plant country economic situation (24 month mean) | 0.0 | 0.0 |
| plant country economic situation (24 month min) | 0.0 | 0.0 |
| plant country economic situation (24 month std) | 0.0 | 0.0 |
| plant country economic situation (6 month max) | 0.0 | 0.0 |
| plant country economic situation (6 month mean) | 0.0 | 0.0 |
| plant country economic situation (6 month min) | 0.0 | 0.0 |
| plant country economic situation (6 month std) | 0.0 | 0.0 |
| plant country fatalities (12 month max) | 0.0 | 0.0 |
| plant country fatalities (12 month mean) | 0.0 | 0.0 |
| plant country fatalities (12 month min) | 0.0 | 0.0 |
| plant country fatalities (12 month std) | 0.0 | 0.0 |
| plant country fatalities (1st lag) | 0.0 | 0.0 |
| plant country fatalities (24 month max) | 0.0 | 0.0 |
| plant country fatalities (24 month mean) | 0.0 | 0.0 |
| plant country fatalities (24 month min) | 0.0 | 0.0 |
| plant country fatalities (24 month std) | 0.0 | 0.0 |
| plant country fatalities (6 month max) | 0.0 | 0.0 |
| plant country fatalities (6 month mean) | 0.0 | 0.0 |
| plant country fatalities (6 month min) | 0.0 | 0.0 |
| plant country fatalities (6 month std) | 0.0 | 0.0 |
| plant country fuel (12 month max) | 0.0 | 0.0 |
| plant country fuel (12 month mean) | 0.0 | 0.0 |
| plant country fuel (12 month min) | 0.0 | 0.0 |
| plant country fuel (12 month std) | 0.0 | 0.0 |
| plant country fuel (1st lag) | 0.0 | 0.0 |
| plant country fuel (24 month max) | 0.0 | 0.0 |
| plant country fuel (24 month mean) | 0.0 | 0.0 |
| plant country fuel (24 month min) | 0.0 | 0.0 |
| plant country fuel (24 month std) | 0.0 | 0.0 |
| plant country fuel (6 month max) | 0.0 | 0.0 |
| plant country fuel (6 month mean) | 0.0 | 0.0 |
| plant country fuel (6 month min) | 0.0 | 0.0 |
| plant country fuel (6 month std) | 0.0 | 0.0 |
| plant country household savings rate (12 month max) | 0.0 | 0.0 |
| plant country household savings rate (12 month mean) | 0.0 | 0.0 |
| plant country household savings rate (12 month min) | 0.0 | 0.0 |
| plant country household savings rate (12 month std) | 0.0 | 0.0 |
| plant country household savings rate (1st lag) | 0.0 | 0.0 |
| plant country household savings rate (24 month max) | 0.0 | 0.0 |
| plant country household savings rate (24 month mean) | 0.0 | 0.0 |
| plant country household savings rate (24 month min) | 0.0 | 0.0 |
| plant country household savings rate (24 month std) | 0.0 | 0.0 |
| plant country household savings rate (6 month max) | 0.0 | 0.0 |
| plant country household savings rate (6 month mean) | 0.0 | 0.0 |
| plant country household savings rate (6 month min) | 0.0 | 0.0 |
| plant country household savings rate (6 month std) | 0.0 | 0.0 |
| plant country registration (12 month max) | 0.0 | 0.0 |
| plant country registration (12 month mean) | 0.0 | 0.0 |
| plant country registration (12 month min) | 0.0 | 0.0 |
| plant country registration (12 month std) | 0.0 | 0.0 |
| plant country registration (1st lag) | 0.0 | 0.0 |
| plant country registration (24 month max) | 0.0 | 0.0 |
| plant country registration (24 month mean) | 0.0 | 0.0 |
| plant country registration (24 month min) | 0.0 | 0.0 |
| plant country registration (24 month std) | 0.0 | 0.0 |
| plant country registration (6 month max) | 0.0 | 0.0 |
| plant country registration (6 month mean) | 0.0 | 0.0 |
| plant country registration (6 month min) | 0.0 | 0.0 |
| plant country registration (6 month std) | 0.0 | 0.0 |
| plant country selling prices (12 month max) | 0.0 | 0.0 |
| plant country selling prices (12 month mean) | 0.0 | 0.0 |
| plant country selling prices (12 month min) | 0.0 | 0.0 |
| plant country selling prices (12 month std) | 0.0 | 0.0 |
| plant country selling prices (1st lag) | 0.0 | 0.0 |
| plant country selling prices (24 month max) | 0.0 | 0.0 |
| plant country selling prices (24 month mean) | 0.0 | 0.0 |
| plant country selling prices (24 month min) | 0.0 | 0.0 |
| plant country selling prices (24 month std) | 0.0 | 0.0 |
| plant country selling prices (6 month max) | 0.0 | 0.0 |
| plant country selling prices (6 month mean) | 0.0 | 0.0 |
| plant country selling prices (6 month min) | 0.0 | 0.0 |
| plant country selling prices (6 month std) | 0.0 | 0.0 |
| plant country short\_term interest rate (12 month max) | 0.0 | 0.0 |
| plant country short\_term interest rate (12 month mean) | 0.0 | 0.0 |
| plant country short\_term interest rate (12 month min) | 0.0 | 0.0 |
| plant country short\_term interest rate (12 month std) | 0.0 | 0.0 |
| plant country short\_term interest rate (1st lag) | 0.0 | 0.0 |
| plant country short\_term interest rate (24 month max) | 0.0 | 0.0 |
| plant country short\_term interest rate (24 month mean) | 0.0 | 0.0 |
| plant country short\_term interest rate (24 month min) | 0.0 | 0.0 |
| plant country short\_term interest rate (24 month std) | 0.0 | 0.0 |
| plant country short\_term interest rate (6 month max) | 0.0 | 0.0 |
| plant country short\_term interest rate (6 month mean) | 0.0 | 0.0 |
| plant country short\_term interest rate (6 month min) | 0.0 | 0.0 |
| plant country short\_term interest rate (6 month std) | 0.0 | 0.0 |
| plant country unemployment rate (12 month max) | 0.0 | 0.0 |
| plant country unemployment rate (12 month mean) | 0.0 | 0.0 |
| plant country unemployment rate (12 month min) | 0.0 | 0.0 |
| plant country unemployment rate (12 month std) | 0.0 | 0.0 |
| plant country unemployment rate (1st lag) | 0.0 | 0.0 |
| plant country unemployment rate (24 month max) | 0.0 | 0.0 |
| plant country unemployment rate (24 month mean) | 0.0 | 0.0 |
| plant country unemployment rate (24 month min) | 0.0 | 0.0 |
| plant country unemployment rate (24 month std) | 0.0 | 0.0 |
| plant country unemployment rate (6 month max) | 0.0 | 0.0 |
| plant country unemployment rate (6 month mean) | 0.0 | 0.0 |
| plant country unemployment rate (6 month min) | 0.0 | 0.0 |
| plant country unemployment rate (6 month std) | 0.0 | 0.0 |
| platform no\_ (1st lag) | 0.0 | 0.0 |
| revenue last 12 months (12 month max) | 0.0 | 0.0 |
| revenue last 12 months (12 month mean) | 0.0 | 0.0 |
| revenue last 12 months (12 month min) | 0.0 | 0.0 |
| revenue last 12 months (12 month std) | 0.0 | 0.0 |
| revenue last 12 months (1st lag) | 0.0 | 0.0 |
| revenue last 12 months (24 month max) | 0.0 | 0.0 |
| revenue last 12 months (24 month mean) | 0.0 | 0.0 |
| revenue last 12 months (24 month min) | 0.0 | 0.0 |
| revenue last 12 months (24 month std) | 0.0 | 0.0 |
| revenue last 12 months (6 month max) | 0.0 | 0.0 |
| revenue last 12 months (6 month mean) | 0.0 | 0.0 |
| revenue last 12 months (6 month min) | 0.0 | 0.0 |
| revenue last 12 months (6 month std) | 0.0 | 0.0 |
| sales date (Day of Week) (actual) | 0.0 | 0.0 |
| sales date (Month) (actual) | 0.0 | 0.0 |
| sales date (Year) (actual) | 0.0 | 0.0 |
| sales date (actual) | 0.0 | 0.0 |
| total net eur (12 month max) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (12 month mean) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (12 month median) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (12 month min) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (1st lag) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (24 month max) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (24 month median) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (24 month min) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (2nd lag) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (3rd lag) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (4th lag) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (5th lag) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (6 month mean) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (6 month median) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (6 month min) (diff 24 month mean) | 0.0 | 0.0 |
| total net eur (quarter aggregation) (actual) | 0.0 | 0.0 |

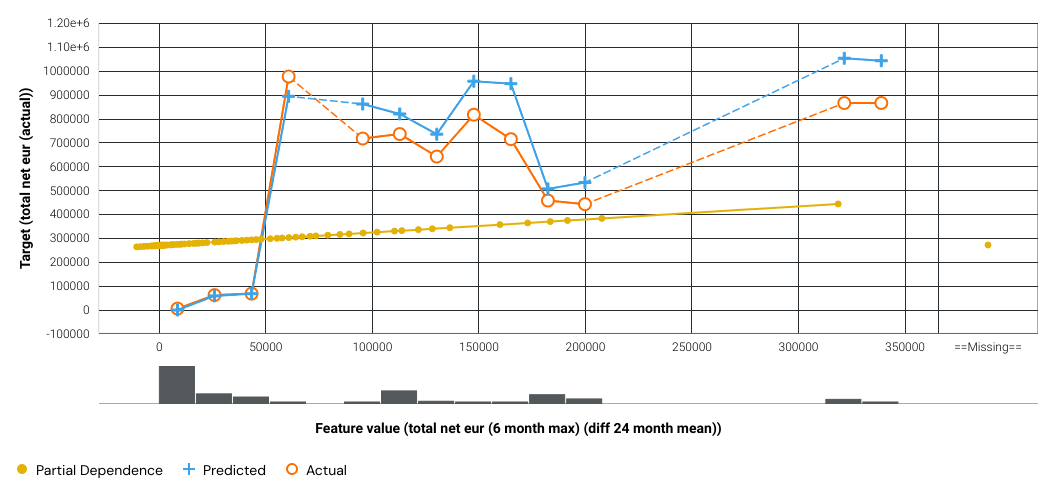
6.3.3 Sensitivity Analysis (Partial Dependence)

In the case of linear regression, users can gain considerable insight into the structure and interpretation of the model by examining its coefficients. For more complex models like support vector machines, random forests, or the blenders considered here, no comparably simple parametric description is available, making the interpretation of these models more difficult. To address this difficulty for his gradient boosting machine, Friedman (2001) proposed the use of partial dependence plots. Partial dependence plots show the average partial relationship between a set of predictors and the predicted response. The partial dependence plots below capture the top features in our model, as measured by Feature Impact.

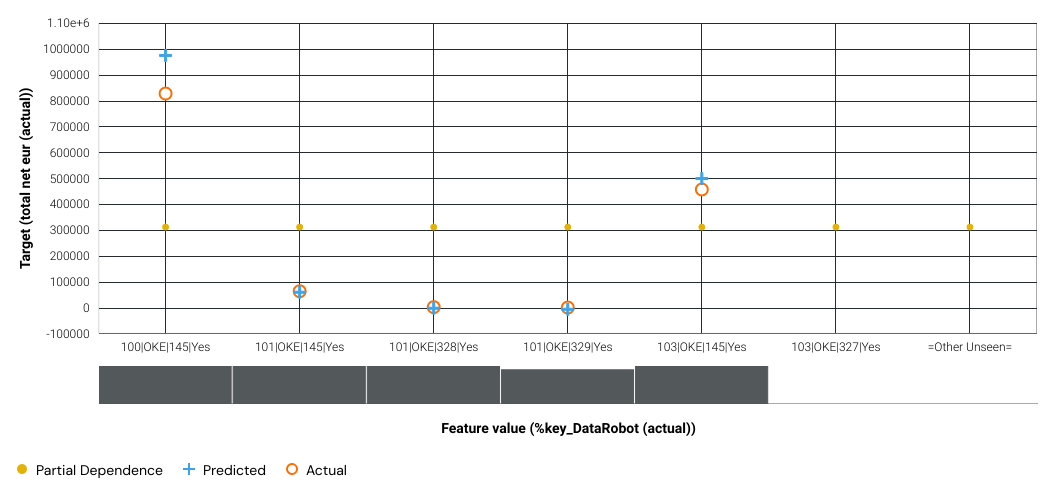
total net eur (24 month average baseline)



total net eur (6 month max) (diff 24 month mean)



%key\_DataRobot (actual)



The orange circles depict, for the selected feature, the average target value for the aggregated feature values. The blue crosses depict, for the selected feature, the average prediction for a specific value. From the graph you can see that DataRobot also averages the predicted feature values. Comparing the actual and predicted points can identify segments where model predictions differ from observed data. This typically occurs when the segment size is small. In those cases, for example, some models may predict closer to the overall average.

The yellow partial dependence data points depict the marginal effect of a feature on the target variable after accounting for the average effects of all other predictive features. It indicates how, holding all other variables constant, the value of this feature affects prediction. DataRobot holds constant the values of all columns in the sample except the feature of interest. The value of the feature of interest is then reassigned to each possible value, calculating the average predictions for the sample at each setting. These values help determine how the value of each feature affects the target. The shape of the yellow data points describes the model's view of the marginal relationship between the selected feature and the target.

6.3.4 Accuracy Over Time

The Accuracy Over Time chart is available using DataRobot’s data/time partitioning. It helps to visualize the predictive stability of a model by comparing how predictions vary over time with actual values. By default, the chart shows predicted and actual vs. time values for the training and validation data of the most recent (first) backtest. This is the backtest model DataRobot uses to deploy and make predictions. (In other words, the model for the validation set.). However, for multiseries projects, the chart presents an average of the series.

Bins within the Accuracy Over Time tab are equal width—that is, each bin spans the same time range—while bins in the Lift Chart are equal sized such that each bin contains the same number of rows. In the chart below, the blue line represents predicted values and the orange line represents actual values.

Accuracy Over Time chart has not been calculated for this model.

7 Model Implementation and Output Reporting

7.1 Version Control

DataRobot handles model and project version control automatically by tagging each model on the Leaderboard with a unique Model ID. The Model ID represents a single instance of a model type, feature list, sample size, and set of tuning parameter values. DataRobot also maintains unique Project IDs for each project, allowing accessibility to all models built for the project dataset. DataRobot's version control allows for reproducibility and traceability of the models it creates, which greatly increases the auditability of the model development process.

Users may also export Scoring Code for a DataRobot model in Java. You can download both a pre-compiled .jar file (with all dependencies included), plus the source code. Scoring Code is easy to deploy, test, and maintain on a variety of platforms, and you can inspect the generated Java code for complete transparency. DataRobot Scoring Code employs advanced features to ensure that predictions computed using generated Java code are the same as predictions computed inside DataRobot.