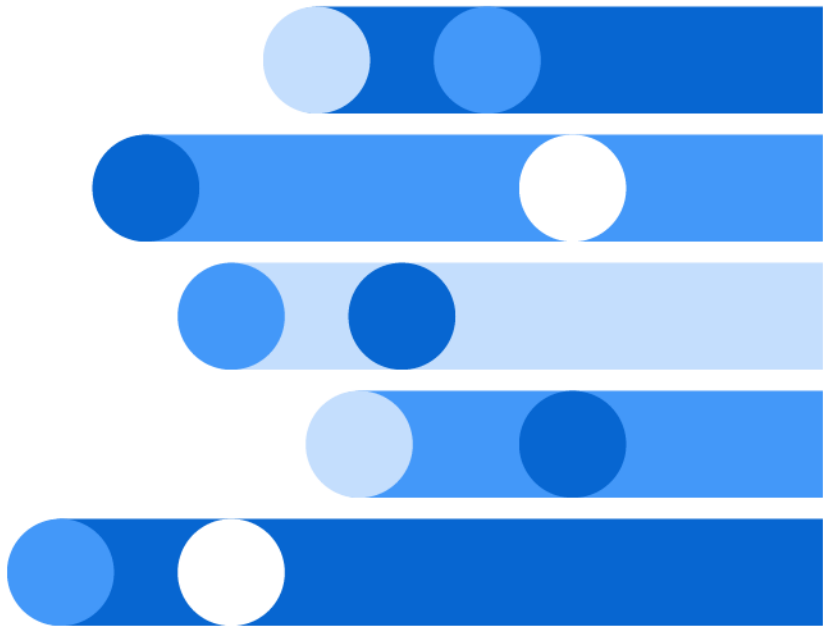
SAS® Viya® Platform: Machine Learning Python API Guide

2025.03*



* This document might apply to additional versions of the software. Open this document in SAS Help Center and click on the version in the banner to see all available versions.

SAS® Documentation
March 19, 2025

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2025. *SAS® Viya® Platform: Machine Learning Python API Guide*. Cary, NC: SAS Institute Inc.

SAS® Viya® Platform: Machine Learning Python API Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

March 2025

v_001-P1:ecepymlpg

Contents

1

Data Processing

Classes

Class Name	Description
Cardinality	The Cardinality class determines a variable's cardinality or limited cardinality.

Cardinality

The Cardinality class determines a variable's cardinality or limited cardinality.

API: Cardinality

class sasviya.ml.cardinality.Cardinality*, level_order='ASC', level_threshold=20, verbose=0

The cardinality of a variable is the number of its distinct values, and the limited cardinality of a variable is the number of its distinct values that do not exceed a specified threshold.

Parameters

level_order : str, optional
Specifies the order that variable levels are sorted. The level order affects how variable levels are sorted in the details output table as well as the list that is returned when calling the get_variable_levels() method. By default, “ASC”. Allowed values: “ASC”, “DESC”.

level_threshold : int, optional
Specifies the maximum number of levels of the variables to consider in the input data. When running summarize(), only up to the specified level_threshold of variable levels, or distinctive values will be considered. If the number of levels exceeds the level_threshold for a variable, the _MORE_ column in the summary output table will be ‘Y’. By default, 20. Allowed values: [5,254]

verbose : int, optional
Specifies the amount of information to print during summarize(). A verbosity level of 0 prints nothing; a verbosity level of 1 prints the cardinality summary table to the log; a verbosity level of 2 prints the cardinality summary and details tables to the log. By default, 0. Allowed values [0,1,2]

Attributes

details_ : Results
The details table that contains information about the model and its results.

variables_in_ : list(str)
The variables included in the summary.

n_variables_in_ : int
The number of variables.

level_threshold_in_ : int
The maximum number of variable levels.

level_order_in_ : str
The order that variable levels are sorted.

nominal_variables_ : list(str)
The names of nominal variables included.

interval_variables_ : list(str)
The names of interval variables included.

binary_variables_ : list(str)
The names of binary variables included.

id_variables_ : list(str)
The names of ID variables included.

Methods

Method Name	Description
get_details_table	Return the cardinality details table.
get_levels	Return the levels of a variable.
get_n_levels	Return the cardinality of a variable.
get_params	Get parameters for this model.
get_summary_table	Return the cardinality summary table.
set_params	Updates the parameters of this summarizer.
summarize	Summarize the cardinality of the input data.

get_details_table

```
get_details_table
```

Return the cardinality details table.

Returns

sasviya Table or pandas DataFrame or pyspark DataFrame

get_levels

```
get_levelsvar
```

Return the levels of a variable.

Parameters

variable : str
Specifies the variable name.

Returns

list(str)

get_n_levels

```
get_n_levelsvar
```

Return the cardinality of a variable.

Parameters

variable : str
Specifies the variable name.

Returns

int

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

get_summary_table

```
get_summary_table
```

Return the cardinality summary table.

Returns

sasviya Table or pandas DataFrame or pyspark DataFrame

set_params

```
set_params**params
```

Updates the parameters of this summarizer.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

summarize

```
summarizeX
```

Summarize the cardinality of the input data.

Parameters

X : array-like of shape (n_samples, n_features)
Specifies the input data to perform cardinality analysis on

Returns

self

2

Decomposition Models

Classes

Class Name	Description
PCA	Principal component analysis.

PCA

Principal component analysis.

API: PCA

class sasviya.ml.decomposition.PCAn_components=None, *, svd_solver='auto', iterated_power=1, random_state=None, scale=True

A class that applies a principal component analysis technique for examining relationships among several quantitative variables. The class provides an optimal way to reduce dimensionality by projecting the data onto a lower-dimensional orthogonal subspace that explains as much variation as possible in those variables.

You can use principal component analysis to reduce the number of variables in regression, clustering, and so on.

Parameters

n_components : int or None, optional

The number of principal components to extract. Must be less than min(n_samples, n_features). Defaults to min(n_samples, n_features) if not specified.

svd_solver : {"auto", "full", "randomized"}, optional
The solver used to calculate the principal components. If "auto" then "full" will be used unless *X* is larger than

500x500 and *n_components* is < 80% of the smallest dimension of the data.

iterated_power : int, optional
Number of solver iterations when *svd_solver* == "randomized", otherwise ignored.

random_state : int, optional
Seed to use for random number generation. Ignored if *svd_solver* is not "randomized"

scale : bool, optional
Specifies to scale variables when set to True. Interval inputs are always centered by their corresponding means. When the scale parameter is set to True, the method divides the inputs by their corresponding standard deviations.

Attributes

details_ : Results
Additional information about the model fitting process and the final results.

components_ : ndarray of shape (n_components, n_features)
Principal axes in feature space, representing the directions of maximum variance.

explained_variance_ : ndarray of shape (n_components,)
The variance explained by each of the selected components.

explained_variance_ratio_ : ndarray of shape (n_components,)
Percentage of variance explained by each of the selected components.

feature_names_in_ : list of str
Number of features used in the model. Equal to len(feature_names_in_).

mean_ : ndarray of shape (n_features,)
Per-feature empirical mean, estimated from training data.

n_components_ : int
The number of principal components extracted.

n_features_in_ : int
Number of features used in the model. Equal to len(feature_names_in_).

n_samples_ : int
Number of samples in the training data.

singular_values_ : ndarray of shape (n_components,)
The singular values corresponding to each of the selected components.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model with X.
fit_transform	Fits the model and applies dimensionality reduction on X.
get_covariance	Gets the covariance of the data.
get_feature_names_out	Gets column names for the transformer's output.
get_params	Get parameters for this model.
inverse_transform	Transforms data back to its original space.
save	Saves the model to a file.
set_params	Updates the parameters of this model.
transform	Reduces the dimensionality of input data.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be “all” or “none”. Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots=”none” on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {“light”, “dark”, “opal”, “midnight”, “raven”, “htmlencore”, “highcontrast”, “grayscale”, monochrome”}
Overrides the default theme used in the graph. The default is “light”. The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {“svg”, “png”}
Overrides the default output type. In Jupyter notebooks, the default output is “svg”, which contains image map information for tool tips. Otherwise, the default is “png”, which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword (“none”, “all”), a single table, or list of tables. The default is “all”, which keeps all tables. The “none” keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the “exclude_tables” option, then the “include_tables” option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword (“none”, “graph_tables”, “all”), a single table, or list of tables. The default is “none”, so that all tables are shown. The “graph_tables” keyword drops all tables that were used to produce a graph. The “all” keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the “include_tables” option, then the “include_tables” option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the “include_tables” or “exclude_tables” option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the “order” option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXy=None
```

Fits the model with X.

Parameters

X : array-like of shape (n_samples, n_features)
Training data, where *n_samples* is the number of samples and *n_features* is the number of features.

y : any
Ignored (for API consistency only).

Returns

self : object
Returns the instance itself.

fit_transform

```
fit_transformXy=None
```

Fits the model and applies dimensionality reduction on X.

Parameters

X : array-like of shape (n_samples, n_features)
Training data, where *n_samples* is the number of samples and *n_features* is the number of features.

y : any
Ignored (for API consistency only).

Returns

array-like
X reduced to (n_samples, n_components).

get_covariance

```
get_covariance
```

Gets the covariance of the data.

Returns

ndarray of shape (n_features, n_features).
The covariance matrix.

get_feature_names_out

```
get_feature_names_out
```

Gets column names for the transformer's output.

Returns

ndarray of str
Column names

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

inverse_transform

```
inverse_transformX
```

Transforms data back to its original space.

In other words, return an input *X_original* whose transform would be X.

Parameters

X : array-like of shape (n_samples, n_components)
Transformed data, where *n_samples* is the number of samples and *n_components* is the number of components.

Returns

X_original : array-like of shape (n_samples, n_features)
Original data, where *n_samples* is the number of samples and *n_features* is the number of features.

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

transform

```
transformX
```

Reduces the dimensionality of input data.

Parameters

X : array-like of shape (n_samples, n_features)
Data to be projected onto the first principal components.

Returns

array-like
X reduced to (n_samples, n_components).

3

Gaussian Mixture Models

Classes

Class Name	Description
GaussianMixtureModel	The GaussianMixtureModel class fits a probabilistic GMM clustering model to data and predicts cluster assignments for new values.

GaussianMixtureModel

The GaussianMixtureModel class fits a probabilistic GMM clustering model to data and predicts cluster assignments for new values.

API: GaussianMixtureModel

class
sasviya.ml.gaussianmixturemodel.GaussianMixtureModel.GaussianMixtureModel*, n_components=100, covariance_type='diagonal', max_iter=100, weight_concentration_prior=1, tol=0.01, random_state=None, verbose=0

Parameters

n_components : int, optional
The number of mixture components. By default, n_components=100.

covariance_type : {"diagonal|diag", "full"}, optional
specifies the covariance matrix type of the Gaussian mixtures. By default, covariance_type = "diagonal".

max_iter : int, optional
Specifies the maximum number of variational Bayesian(VB) iterations to perform. By default, max_iter=100.

weight_concentration_prior : float, optional
The Dirichlet concentration of each component on the weight distribution. By default, weight_concentration_prior=1.

tol : float, optional
The convergence threshold of VB algorithm. By default, tol=0.01.

random_state : float, optional
Specifies the seed to use for random number generation for initialization.

verbose : int, optional
Specifies the amount of information to print about the fitted model.

Attributes

details_ : Results
Additional information about the model fitting process and the final results.

cluster_summary_ : ndarray of shape (n_components, n_features+3)
The cluster size, neighbor, and center of each mixture component.

cluster_info_ : ndarray of shape (n_components+2, n_features*2 + 2)
The cluster weight, mean, variance of each mixture component.

cluster_covariance_ : ndarray of shape (n_components*n_features, n_features + 1)
The covariance of each mixture component.

labels_ : ndarray of shape (n_samples,)
The labels of each observation.

converged_ : bool
True when convergence was reached in fit()

n_iter_ : int
The number of model fit iterations performed.

n_features_in_ : int
The number of input features seen by the model.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
fit_predict	Fits the model on the provided training data and predicts cluster assignments
get_params	Get parameters for this model.
predict	Predicts the closest cluster for each given input feature.
predict_proba	Predicts the cluster probabilities for each given input feature.
save	Saves the model to a file.
score	Scores clustering results on data X, based on the learned model.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be "all" or "none". Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword ("none", "all"), a single table, or list of tables. The default is "all", which keeps all tables. The "none" keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the "exclude_tables" option, then the "include_tables" option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword ("none", "graph_tables", "all"), a single table, or list of tables. The default is "none", so that all tables are shown. The "graph_tables" keyword drops all tables that were used to produce a graph. The "all" keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the "include_tables" option, then the "include_tables" option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the

"include_tables" or "exclude_tables" option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the "order" option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXy=None
```

Fits the model on the training data.

Parameters

X : array-like of shape (n_samples, n_features)
Specifies the training data, where *n_samples* is the number of samples and *n_features* is the number of features.

y : any
Ignored (included for API consistency only).

Returns

self : object
Returns the instance itself.

fit_predict

```
fit_predictXy=None
```

Fits the model on the provided training data and predicts cluster assignments

Equivalent to calling fit(X) followed by predict(X).

Parameters

X : array-like
Specifies the samples to fit and predict the closest cluster for, with shape (n_samples, n_features).

y : any
Ignored (included for API consistency only).

Returns

type(X)
Predicted index of the cluster for each input in the shape (n_samples,).

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

Predicts the closest cluster for each given input feature.

Parameters

X : array-like
Specifies the samples to predict, with shape (n_samples, n_features).

Returns

type(X)
Predicted index of the cluster for each input in the shape (n_samples,).

predict_proba

```
predict_probaX
```

Predicts the cluster probabilities for each given input feature.

Parameters

X : array-like
Specifies the samples to predict, with shape (n_samples, n_features).

Returns

type(X)
Predicted probabilities of the cluster for each input in the shape (n_samples, n_components).

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy=None
```

Scores clustering results on data X, based on the learned model.

Parameters

X : array-like of shape (n_samples, n_features)
Specifies new data, where *n_samples* is the number of samples and *n_features* is the number of features.

y : any
Ignored (included for API consistency only).

Returns

type(X)
Index of the cluster that each observation belongs to in addition to the probability that it belongs to the cluster.

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

4

Linear Models

Classes

Class Name	Description
ElasticNet	Linear Regression with a mix of L1 and L2 regularization.
Lasso	Linear Regression with L1 regularization.
LinearRegression	Ordinary Least Squares Linear Regression.
LogisticRegression	Logistic Regression classifier.
Ridge	Linear Regression with L2 regularization.

ElasticNet

Linear Regression with a mix of L1 and L2 regularization.

API: ElasticNet

class sasviya.ml.linear_model.ElasticNetalpha=None, *, l1_ratio=0.5, fit_intercept=True, tol=1e-08, solver='lbfgs', verbose=0

Parameters

alpha : float, optional
Specifies the strength of the regularization applied. A constant value used to scale the regularization term of the cost function. If None is specified, the regularization term is computed from the data. "Alpha" is also referred to as "lambda" in fit results.

l1_ratio : float, optional
Specifies the elastic net mixing parameter. Specifies the ratio of L1:L2 regularization. If *l1_ratio* = 0, only L2 regularization is used and if *l1_ratio* = 1, only L1 regularization is used.

fit_intercept : bool, optional
Specifies whether to include a bias term in the model.

solver : str, optional
Specifies the optimization algorithm to use. Valid options are: "admm", "bfgs", "lbfgs", and "nlp".

tol : float optional
Specifies the tolerance threshold for early stopping. Model fitting stops when the size of the gradient is less than *tol*.

verbose : int
Specifies the amount of information to print about the fitted model.

Attributes

coef_ : ndarray of shape (n_parameters,)
Model coefficients. The number of parameters may be greater than the number of features if additional parameters are introduced to handle missing values, or less than the number of features if variable selection is used.

details_ : Results
Additional information about the model fitting process and the final results.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

intercept_ : ndarray of shape (, n_intercepts).
Bias term(s) added to the model.

n_features_in_ : int
The number of input features seen by the model.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predict the results of a data set based on the model.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be "all" or "none". Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword ("none", "all"), a single table, or list of tables. The default is "all", which keeps all tables. The "none" keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the "exclude_tables" option, then the "include_tables" option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword ("none", "graph_tables", "all"), a single table, or list of tables. The default is "none", so that all tables are shown. The "graph_tables" keyword drops all tables that were used to produce a graph. The "all" keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the "include_tables" option, then the "include_tables" option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the

"include_tables" or "exclude_tables" option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the "order" option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Names of columns in *X* that should be treated as categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

Lasso

Linear Regression with L1 regularization.

API: Lasso

class sasviya.ml.linear_model.Lasso*, fit_intercept=True, verbose=0

Parameters

fit_intercept : bool, optional
Specifies whether to include a bias term in the model.

verbose : int
Specifies the amount of information to print about the fitted model.

Attributes

coef_ : ndarray of shape (n_parameters,)
Model coefficients. The number of parameters may be greater than the number of features if additional parameters are introduced to handle missing values, or less than the number of features if variable selection is used.

details_ : Results
Additional information about the model fitting process and the final results.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

intercept_ : ndarray of shape (, n_intercepts).
Bias term(s) added to the model.

n_features_in_ : int
The number of input features seen by the model.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predict the results of a data set based on the model.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be "all" or "none". Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword ("none", "all"), a single table, or list of tables. The default is "all", which keeps all tables. The "none" keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the "exclude_tables" option, then the "include_tables" option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword ("none", "graph_tables", "all"), a single table, or list of tables. The default is "none", so that all tables are shown. The "graph_tables" keyword drops all tables that were used to produce a graph. The "all" keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the "include_tables" option, then the "include_tables" option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the

"include_tables" or "exclude_tables" option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the "order" option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Names of columns in *X* that should be treated as categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

LinearRegression

Ordinary Least Squares Linear Regression.

API: LinearRegression

class sasviya.ml.linear_model.LinearRegression*, fit_intercept=True, verbose=0

Parameters

fit_intercept : bool, optional
Specifies whether to include a bias term in the model.

verbose : int
Specifies the amount of information to print about the fitted model.

Attributes

coef_ : pandas.DataFrame of shape (n_parameters,)
Model coefficients. The number of parameters may be greater than the number of features if additional parameters are introduced to handle missing values, or less than the number of features if variable selection is used.

details_ : Results
Additional information about the model fitting process and the final results.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

intercept_ : pandas.DataFrame of shape (, n_intercepts).
Bias term(s) added to the model.

n_features_in_ : int
The number of input features seen by the model.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predict the results of a data set based on the model.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be "all" or "none". Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword ("none", "all"), a single table, or list of tables. The default is "all", which keeps all tables. The "none" keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the "exclude_tables" option, then the "include_tables" option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword ("none", "graph_tables", "all"), a single table, or list of tables. The default is "none", so that all tables are shown. The "graph_tables" keyword drops all tables that were used to produce a graph. The "all" keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the "include_tables" option, then the "include_tables" option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the

"include_tables" or "exclude_tables" option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the "order" option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Names of columns in *X* that should be treated as categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

LogisticRegression

Logistic Regression classifier.

API: LogisticRegression

class sasviya.ml.linear_model.LogisticRegression*, tol=1e-08, fit_intercept=True, solver='nrridg', selection=None, max_iter=50, max_time=None, verbose=0

Parameters

tol : float, optional
Specifies the relative gradient convergence criterion.

fit_intercept : bool, optional
Specifies whether to include a bias term in the model.

max_iter : float, optional
Specifies the maximum number of training iterations.

max_time : float, optional
Specifies the maximum training time, in seconds.

solver : str, optional
Specifies the optimization algorithm to use. Valid options are: “congra” - conjugate gradient method “dbldog” - double-dogleg method “lbfgs” - limited-memory BFGS solver “newrap” - Newton-Raphson method with line search and ridging “nmsimp” - Nelder-Mead simplex method “nrridg” - Newton-Raphson method with ridging “quanew | duquanew” - dual quasi-Newton optimization.

selection : str, optional
Specifies the variable selection method to use. Valid options are: “backward”, “forward”, “lasso”, and “stepwise”.

verbose : int: optional
Specifies the information to print about the fitted model: 0: None 1: Iteration history 2: Iteration history and fit statistics

Attributes

classes_ : ndarray of shape (n_classes,)
Class labels seen during model fitting.

coef_ : pandas.DataFrame of shape (n_parameters,)
Model coefficients. The number of parameters may be greater than the number of features if additional parameters are introduced to handle missing values, or less than the number of features if variable selection was used.

details_ : Results
Additional information about the model fitting process and the final results.

intercept_ : pandas.DataFrame of shape (n_classes-1,)
Bias term(s) added to the model.

n_iter_ : int
The number of model fit iterations performed.

n_features_in_ : int
The number of input features seen by the model.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
decision_function	Computes the decision function given input features.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the provided training data.
get_params	Get parameters for this model.

Method Name	Description
predict	Predicts class labels given input features.
predict_proba	Estimates probabilities for each class label.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be “all” or “none”. Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots=”none” on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {“light”, “dark”, “opal”, “midnight”, “raven”, “htmlencore”, “highcontrast”, “grayscale”, monochrome”}
Overrides the default theme used in the graph. The default is “light”. The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {“svg”, “png”}
Overrides the default output type. In Jupyter notebooks, the default output is “svg”, which contains image map information for tool tips. Otherwise, the default is “png”, which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword (“none”, “all”), a single table, or list of tables. The default is “all”, which keeps all tables. The “none” keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the “exclude_tables” option, then the “include_tables” option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword (“none”, “graph_tables”, “all”), a single table, or list of tables. The default is “none”, so that all tables are shown. The “graph_tables” keyword drops all tables that were used to produce a graph. The “all” keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the “include_tables” option, then the “include_tables” option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the “include_tables” or “exclude_tables” option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the “order” option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

decision_function

```
decision_functionX
```

Computes the decision function given input features.

Parameters

X : array-like
Specifies training features in the shape (n_samples, n_features).

Returns

array-like
Estimated posterior probability for the target class. This is (n_samples, n_classes) for the multi-class case, and (n_samples,) with just self.classes_[1] returned for the binary case.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the provided training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Names of columns in *X* that should be treated as categorical. The target is always treated as categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

Predicts class labels given input features.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Predicted class labels in the shape (n_samples,).

predict_proba

```
predict_probaX
```

Estimates probabilities for each class label.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Estimated class label probabilities in the shape (n_samples, n_classes).

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

Ridge

Linear Regression with L2 regularization.

API: Ridge

class sasviya.ml.linear_model.Ridgealpha=None, *, fit_intercept=True, tol=1e-08, solver='lbfgs', verbose=0

Parameters

alpha : float, optional
Specifies the strength of the regularization applied. A constant value used to scale the regularization term of the cost function. If None is specified, the regularization term is computed from the data. "Alpha" is also referred to as "lambda" in fit results.

fit_intercept : bool, optional
Specifies whether to include a bias term in the model.

solver : str, optional
Specifies the optimization algorithm to use. Valid options are: "admm", "bfgs", "lbfgs", and "nlp".

tol : float, optional
Specifies the tolerance threshold for early stopping. Model fitting stops when the size of the gradient is less than *tol*.

verbose : int
Specifies the amount of information to print about the fitted model.

Attributes

coef_ : ndarray of shape (n_parameters,)
Model coefficients. The number of parameters may be greater than the number of features if additional parameters are introduced to handle missing values, or less than the number of features if variable selection is used.

details_ : Results
Additional information about the model fitting process and the final results.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

intercept_ : ndarray of shape (, n_intercepts).
Bias term(s) added to the model.

n_features_in_ : int
The number of input features seen by the model.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predict the results of a data set based on the model.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list

Request plots. This can be a single plot or list of plots. This request can also be “all” or “none”. Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You

will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword ("none", "all"), a single table, or list of tables. The default is "all", which keeps all tables. The "none" keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the "exclude_tables" option, then the "include_tables" option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword ("none", "graph_tables", "all"), a single table, or list of tables. The default is "none", so that all tables are shown. The "graph_tables" keyword drops all tables that were used to produce a graph. The "all" keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the "include_tables" option, then the "include_tables" option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the "include_tables" or "exclude_tables" option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the "order" option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Names of columns in *X* that should be treated as categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

5

Support Vector Models

Classes

Class Name	Description
SVC	Support Vector Classification model.
SVR	Support Vector Regression model.

SVC

Support Vector Classification model.

API: SVC

class sasviya.ml.svm.SVC*, max_iter=25, C=1.0, kernel='linear', degree=2, sigma='auto', coef0=-1, method=None, tol=None, random_state=1, scale=True, verbose=0

Parameters

max_iter : int, optional
Specifies the maximum number of training iterations.

C : float, optional
Specifies the cost regularization parameter.

kernel : {“linear”, “poly”, “polynomial”, “rbf”, “sigmoid”}, optional
Specifies the type of kernel to use.

degree : int, optional
Specifies the degree of the polynomial kernel. Ignored if *kernel* is not “polynomial”.

sigma : “auto”, “scale” or float, optional
Specifies the kernel parameter for the RBF and Sigmoid kernels.

coef0 : float, optional
Specifies the second parameter for the Sigmoid kernel.

method : {“activeset”, “cd”, “ipoint”} or None, optional

Specifies the optimization method used to fit the model.

- Active-Set can be used with all kernels and is the default for the RBF and Sigmoid kernels, as well as polynomial kernels of more than 3 degree.
- Interior Point can be used with the linear and polynomial kernels and is the default for linear kernels and polynomial kernels of degree 3 or less.
- Coordinate Descent can only be used with the linear kernel.

tol : float, optional
Specifies the tolerance threshold for early stopping.

random_state : int, optional
Specifies the seed to use for random number generation.

scale : bool, optional
When set to True, scales the numeric variables prior to fitting.

verbose : int, optional
Specifies the amount of information to print about the fitted model.

Attributes

classes_ : ndarray of shape (2,)
Class labels seen during model fitting.

details_ : Results
Additional information about the model fitting process and the final results.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

n_features_in_ : int
The number of input features seen by the model.

n_iter_ : int
The number of training iterations. Only present when the Interior Point algorithm is used.

n_support_ : int
The number of support vectors. Not present if *method* is “cd”.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
decision_function	Computes the decision function given input features.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predicts class labels given input features.
predict_proba	Estimates probabilities for each class label.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepa
th=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=No
ne
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be “all” or “none”. Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots=”none” on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {“light”, “dark”, “opal”, “midnight”, “raven”, “htmlencore”, “highcontrast”, “grayscale”, monochrome”}
Overrides the default theme used in the graph. The default is “light”. The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {“svg”, “png”}
Overrides the default output type. In Jupyter notebooks, the default output is “svg”, which contains image map information for tool tips. Otherwise, the default is “png”, which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword (“none”, “all”), a single table, or list of tables. The default is “all”, which keeps all tables. The “none” keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the “exclude_tables” option, then the “include_tables” option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword (“none”, “graph_tables”, “all”), a single table, or list of tables. The default is “none”, so that all tables are shown. The “graph_tables” keyword drops all tables that were used to produce a graph. The “all” keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the “include_tables” option, then the “include_tables” option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the “include_tables” or “exclude_tables” option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the “order” option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

decision_function

```
decision_functionX
```

Computes the decision function given input features.

Parameters

X : array-like
Specifies training features in the shape (n_samples, n_features).

Returns

array-like
Estimated posterior probability for the target class. This is (n_samples, n_classes) for the multi-class case, and (n_samples,) with just self.classes_[1] returned for the binary case.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Specifies the names of columns in *X* that should be treated as nominal values. For classifier models, the target is assumed to be categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

Predicts class labels given input features.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Predicted class labels in the shape (n_samples,).

predict_proba

```
predict_probaX
```

Estimates probabilities for each class label.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Estimated class label probabilities in the shape (n_samples, n_classes).

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

SVR

Support Vector Regression model.

API: SVR

```
class sasviya.ml.svm.SVR*, max_iter=25, C=1.0, kernel='linear', degree=2,
tol=None, epsilon=0.01, random_state=1, scale=True, verbose=0
```

Parameters

max_iter : int, optional
Specifies the maximum number of training iterations.

C : float, optional
Specifies the cost regularization parameter.

kernel : {“linear”, “poly”, “polynomial”}, optional
Specifies the type of kernel to use.

degree : int, optional
Specifies the degree of the polynomial kernel. Ignored if *kernel* is not “polynomial”.

tol : float, optional
Specifies the tolerance threshold for early stopping.

epsilon : float, optional
Specifies the epsilon-tube distance. Points that have a predicted value within this distance from their actual value are not counted towards the training loss.

random_state : int, optional
Specifies the seed to use for random number generation.

scale : bool, optional
When set to True, scales the numeric variables prior to fitting.

verbose : int, optional
Specifies the amount of information to print about the fitted model.

Attributes

details_ : Results
Additional information about the model fitting process and the final results.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

n_features_in_ : int
The number of input features seen by the model.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predict the results of a data set based on the model.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list

Request plots. This can be a single plot or list of plots. This request can also be “all” or “none”. Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You

will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword ("none", "all"), a single table, or list of tables. The default is "all", which keeps all tables. The "none" keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the "exclude_tables" option, then the "include_tables" option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword ("none", "graph_tables", "all"), a single table, or list of tables. The default is "none", so that all tables are shown. The "graph_tables" keyword drops all tables that were used to produce a graph. The "all" keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the "include_tables" option, then the "include_tables" option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the "include_tables" or "exclude_tables" option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the "order" option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Specifies the names of columns in *X* that should be treated as nominal values. For classifier models, the target is assumed to be categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

6

Tree Models

Classes

Class Name	Description
DecisionTreeClassifier	A decision tree classification model.
DecisionTreeRegressor	A decision tree regression model.
ForestClassifier	A random forest classification model.
ForestRegressor	A random forest regression model.
GradientBoostingClassifier	Gradient boosting classifier.
GradientBoostingRegressor	Gradient boosting regression model.

DecisionTreeClassifier

A decision tree classification model.

API: DecisionTreeClassifier

class sasviya.ml.tree.DecisionTreeClassifiercriterion='IGR', max_depth=10, min_samples_leaf=5, ccp_alpha=0, verbose=0

Parameters

criterion : {"chaid", "chisquare", "entropy|gain", "igr|gainratio", "gini"}
Specifies the split criterion for each tree node.

max_depth : int, optional
Specifies the maximum depth of the tree.

min_samples_leaf : int, optional
Specifies the minimum number of observations on each node.

ccp_alpha : float, optional
Specifies the value to use for minimal cost-complexity pruning for regression trees.

verbose : int, optional
Specifies the amount of information to print about the fitted model.

Attributes

details_ : Results
Additional information about the model fitting process and the final results.

n_features_in_ : int
The number of input features seen by the model. This can be a subset of the input columns if variable selection is used.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

n_features_used_ : int
The number of input features used by the model.

target_name_in_ : str
Column name of the target variable.

n_nodes_ : int
The number of tree nodes.

max_n_branches_ : int
The maximum number of branches of the tree.

depth_ : int
The depth of the tree.

n_leaves_ : int
The number of leaves of the tree.

n_bins_ : int
The number of bins for numeric variables.

min_n_leaves_ : int
The minimum size of leaves.

max_n_leaves_ : int
The maximum size of leaves.

n_obs_ : int
The number of observations used by the model.

classes_ : list of str
The class labels.

n_classes_ : int
The number of classes.

feature_importances_ : pandas.DataFrame()
Ranking of features by importance.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
decision_function	Computes the decision function given input features.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predicts class labels given input features.
predict_proba	Estimates probabilities for each class label.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.

Method Name	Description
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be “all” or “none”. Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots=”none” on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {“light”, “dark”, “opal”, “midnight”, “raven”, “htmlencore”, “highcontrast”, “grayscale”, monochrome”}
Overrides the default theme used in the graph. The default is “light”. The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {“svg”, “png”}
Overrides the default output type. In Jupyter notebooks, the default output is “svg”, which contains image map information for tool tips. Otherwise, the default is “png”, which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword (“none”, “all”), a single table, or list of tables. The default is “all”, which

keeps all tables. The “none” keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the “exclude_tables” option, then the “include_tables” option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword (“none”, “graph_tables”, “all”), a single table, or list of tables. The default is “none”, so that all tables are shown. The “graph_tables” keyword drops all tables that were used to produce a graph. The “all” keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the “include_tables” option, then the “include_tables” option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the “include_tables” or “exclude_tables” option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the “order” option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

decision_function

```
decision_functionX
```

Computes the decision function given input features.

Parameters

X : array-like
Specifies training features in the shape (n_samples, n_features).

Returns

array-like
Estimated posterior probability for the target class. This is (n_samples, n_classes) for the multi-class case, and (n_samples,) with just self.classes_[1] returned for the binary case.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=Noneweight=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies target labels in the shape (n_samples,).

nominals : list of str, optional
Specifies the names of columns in *X* that should be treated as categorical. For classifier models, the target is assumed to be categorical.

weight : str, optional
Specifies the name of a column in *X* that contains the weight of each observation.

Returns

——-

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

Predicts class labels given input features.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Predicted class labels in the shape (n_samples,).

predict_proba

```
predict_probaX
```

Estimates probabilities for each class label.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Estimated class label probabilities in the shape (n_samples, n_classes).

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

DecisionTreeRegressor

A decision tree regression model.

API: DecisionTreeRegressor

```
class sasviya.ml.tree.DecisionTreeRegressorcriterion='RSS', max_depth=10,
min_samples_leaf=5, ccp_alpha=0, verbose=0
```

Parameters

criterion : {“chaid”, “ftest”, “variance|rss”}
Specifies the criterion used to measure the quality of a split in each tree node.

max_depth : int, optional
Specifies the maximum depth of any tree.

min_samples_leaf : int, optional
Specifies the minimum number of observations at each node.

ccp_alpha : float, optional
Specifies the alpha value to use for minimal cost-complexity pruning for regression trees.

verbose : int, optional
Specifies the amount of information to print about the fitted model.

Attributes

details_ : Results
Additional information about the model fitting process and the final results.

n_features_in_ : int
The number of input features seen by the model.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

n_features_used_ : int
The number of input features used by the model.

target_name_in_ : str
Column name of the target variable.

n_nodes_ : int
The number of tree nodes.

max_n_branches_ : int
The maximum number of branches of the tree.

depth_ : int
The depth of the tree.

n_leaves_ : int
The number of leaves of the tree.

n_bins_ : int
The number of bins for numeric variables.

min_n_leaves_ : int
The minimum size of leaves.

max_n_leaves_ : int
The maximum size of leaves.

n_obs_ : int
The number of observations used by the model.

feature_importances_ : pandas.DataFrame()
Ranking of features by importance.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predict the results of a data set based on the model.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be "all" or "none". Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword ("none", "all"), a single table, or list of tables. The default is "all", which keeps all tables. The "none" keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the "exclude_tables" option, then the "include_tables" option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword ("none", "graph_tables", "all"), a single table, or list of tables. The default is "none", so that all tables are shown. The "graph_tables" keyword drops all tables that were used to produce a graph. The "all" keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the "include_tables" option, then the "include_tables" option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the

"include_tables" or "exclude_tables" option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the "order" option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=Noneweight=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies target labels in the shape (n_samples,).

nominals : list of str, optional
Specifies the names of columns in *X* that should be treated as categorical. For classifier models, the target is assumed to be categorical.

weight : str, optional
Specifies the name of a column in *X* that contains the weight of each observation.

Returns

——-

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

ForestClassifier

A random forest classification model.

API: ForestClassifier

class sasviya.ml.tree.ForestClassifiern_estimators=100, *, bootstrap=0.6, criterion='IGR', max_depth=20, max_features=None, min_samples_leaf=5, n_bins=50, oob_score=True, random_state=None, verbose=0

Parameters

n_estimators : int, optional
Specifies the number of trees to create.

bootstrap : float, optional
Specifies the fraction of the data to use for the bootstrap sample.

criterion : {"chaid", "chisquare", "entropy|gain", "igr|gainratio", "gini"}
Specifies the criterion used to measure the quality of a split in each tree node.

max_depth : int, optional
Specifies the maximum depth of any tree.

max_features : int, optional
Specifies the number of input variables to consider when splitting a node (also referred to as "M" in model output). The variables are selected at random from the input variables for each tree. By default, max_feature is equal to the the square root of the number of input variables.

min_samples_leaf : int, optional
Specifies the minimum number of observations at each node.

n_bins : int, optional
Specifies the number of bins to use for numeric variables.

oob_score : bool, optional
Specifies whether to compute out-of-bag error when training the model.

random_state : int, optional
Specifies the seed to use for random number generation.

verbose : int, optional
Specifies the amount of information to print about the fitted model.

Attributes

classes_ : list of str with length n_classes
Class labels seen during model fitting.

details_ : Results
Additional information about the model fitting process and the final results.

feature_importances_ : DataFrame
Feature importances calculated during model fitting, scaled to [0, 1] range.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

n_classes_ : int
Number of target classes.

n_features_in_ : int
The number of input features seen by the model.

oob_score_ : float
The out-of-bag error observed during model fitting. Computed only if the *oob_score* parameter is True.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
decision_function	Computes the decision function given input features.

Method Name	Description
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predicts class labels given input features.
predict_proba	Estimates probabilities for each class label.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be “all” or “none”. Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots=”none” on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {“light”, “dark”, “opal”, “midnight”, “raven”, “htmlencore”, “highcontrast”, “grayscale”, monochrome”}
Overrides the default theme used in the graph. The default is “light”. The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {“svg”, “png”}
Overrides the default output type. In Jupyter notebooks, the default output is “svg”, which contains image map information for tool tips. Otherwise, the default is “png”, which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword (“none”, “all”), a single table, or list of tables. The default is “all”, which keeps all tables. The “none” keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the “exclude_tables” option, then the “include_tables” option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword (“none”, “graph_tables”, “all”), a single table, or list of tables. The default is “none”, so that all tables are shown. The “graph_tables” keyword drops all tables that were used to produce a graph. The “all” keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the “include_tables” option, then the “include_tables” option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the “include_tables” or “exclude_tables” option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the “order” option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

decision_function

```
decision_functionX
```

Computes the decision function given input features.

Parameters

X : array-like
Specifies training features in the shape (n_samples, n_features).

Returns

array-like
Estimated posterior probability for the target class. This is (n_samples, n_classes) for the multi-class case, and (n_samples,) with just self.classes_[1] returned for the binary case.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Speifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Specifies the names of columns in *X* that should be treated as categorical. For classifier models, the target is assumed to be categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

Predicts class labels given input features.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Predicted class labels in the shape (n_samples,).

predict_proba

```
predict_probaX
```

Estimates probabilities for each class label.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Estimated class label probabilities in the shape (n_samples, n_classes).

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

ForestRegressor

A random forest regression model.

API: ForestRegressor

class sasviya.ml.tree.ForestRegressorn_estimators=100, *, bootstrap=0.6, criterion='RSS', max_depth=20, max_features=None, min_samples_leaf=5, n_bins=50, oob_score=True, random_state=None, verbose=0

Parameters

n_estimators : int, optional
Specifies the number of trees to create.

bootstrap : float, optional
Specifies the fraction of the data to use for the bootstrap sample.

criterion : {"chaid", "ftest", "variance|rss"}
Specifies the criterion used to measure the quality of a split in each tree node.

max_depth : int, optional
Specifies the maximum depth of any tree.

max_features : int, optional
Specifies the number of input variables to consider when splitting a node (also referred to as "M" in model output). The variables are selected at random from the input variables for each tree. By default, max_features is equal to the square root of the number of input variables.

min_samples_leaf : int, optional
Specifies the minimum number of observations at each node.

n_bins : int, optional
Specifies the number of bins to use for numeric variables.

oob_score : bool, optional
Specifies whether to compute out-of-bag error when training the model.

random_state : int, optional
Specifies the seed to use for random number generation.

verbose : int, optional
Specifies the amount of information to print about the fitted model.

Attributes

details_ : Results
Additional information about the model fitting process and the final results.

feature_importances_ : DataFrame
Feature importances calculated during model fitting, scaled to [0, 1] range.

feature_names_in_ : list of str
The column names of the input features seen during fitting.

n_features_in_ : int
The number of input features seen by the model.

oob_score_ : float
The out-of-bag error observed during model fitting. Computed only if the *oob_score* parameter is True.

target_name_in_ : str
Column name of the target variable.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predict the results of a data set based on the model.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepa
th=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=No
ne
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be “all” or “none”. Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots=”none” on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {“light”, “dark”, “opal”, “midnight”, “raven”, “htmlencore”, “highcontrast”, “grayscale”, monochrome”}
Overrides the default theme used in the graph. The default is “light”. The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {“svg”, “png”}
Overrides the default output type. In Jupyter notebooks, the default output is “svg”, which contains image map information for tool tips. Otherwise, the default is “png”, which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword (“none”, “all”), a single table, or list of tables. The default is “all”, which keeps all tables. The “none” keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the “exclude_tables” option, then the “include_tables” option has precedence.

exclude_tables : str or str list

Excludes tables from the display results. This can be a keyword (“none”, “graph_tables”, “all”), a single table, or list of tables. The default is “none”, so that all tables are shown. The “graph_tables” keyword drops all tables that were used to produce a graph. The “all” keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the “include_tables” option, then the “include_tables” option has precedence.

order : str or str list

Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the “include_tables” or “exclude_tables” option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the “order” option is ignored.

Returns

CustomResults

Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results

a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Speifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Specifies the names of columns in *X* that should be treated as categorical. For classifier models, the target is assumed to be categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

GradientBoostingClassifier

Gradient boosting classifier.

API: GradientBoostingClassifier

```
class sasviya.ml.tree.GradientBoostingClassifier*, min_samples_leaf=5,
n_bins=50, n_estimators=100, max_depth=4, max_features=None,
learning_rate=0.1, subsample=0.5, l1_penalty=0, l2_penalty=1,
random_state=None, verbose=0, n_iter_no_change=0, tol=0,
calc_feature_importances=False, validation_fraction=0
```

Parameters

n_bins: int, optional
Specifies the number of bins to use for numeric variables. By default, 50.

min_samples_leaf : int, optional
Specifies the minimum number of observations on each node. By default, 5.

n_estimators : int, optional
Specifies the number of trees to create. By default, 100.

max_depth : int, optional
Specifies the maximum depth of the tree. By default, 4.

max_features : float, optional
Specifies the number of input variables to consider for splitting on a node. By default, max_features is set to the number of input variables.

learning_rate : float, optional
Specifies the learning rate of each tree. By default, 0.1.

subsample : float, optional
Specifies the fraction of the data to use for building each tree. By default, 0.5.

l1_penalty : float, optional
Specifies the L1 norm regularization on prediction. By default, 0.

l2_penalty : float, optional
Specifies the L2 norm regularization on prediction. By default, 1.

random_state : float, optional
Specifies the seed for the random number generator. By default, the random number stream is based on the computer clock. If you want a reproducible random number sequence between runs, specify a value that is greater than zero.

verbose : int, optional
Specifies the amount of information to display about the fitted model.

calc_feature_importances : bool, optional
Specifies whether to calculate feature importances.

validation_fraction: float, optional
Specifies the fraction of training samples to use for validation. A validation_fraction greater than zero enables early stopping mode.

n_iter_no_change : int, optional
Specifies the maximum number of training iterations for early stopping. Ignored unless a validation fraction is specified.

tol : float , optional
Specifies the tolerance value for early stopping as the change relative to the previous iteration. Ignored unless a validation fraction is specified.

Attributes

details_ : Results
Information about the model fitting process and the final results.

n_estimators_ : int
Number of trees in the fitted model. The number of trees could be fewer than requested due to early stopping.

avg_n_leaves_ : float
Average number of leaves among trees in the fitted model.

min_n_leaves_ : int
Minimum number of leaves among trees in the fitted model.

max_n_leaves_ : int
Maximum number of leaves among trees in the fitted model.

max_n_nodes_ : int
Maximum number of nodes among trees in the fitted model.

min_n_nodes_ : int
Minimum number of nodes among trees in the fitted model.

max_depth_ : int
Maximum depth among trees in the fitted model.

min_depth_ : int
Minimum depth among trees in the fitted model.

min_samples_leaf_ : int
Minimum leaf size among trees in the fitted model.

max_samples_leaf_ : int
Maximum leaf size among trees in the fitted model.

n_features_in_ : int
The number of input features.

feature_names_in_ : list of str
The column names of the input features.

n_features_used_ : int
The number of features used by the fitted model.

target_name_in_ : str
Column name of the target variable.

classes_ : list of str
The class labels.

n_classes_ : int
Number of classes of the target variable.

feature_importances_ : pandas.DataFrame()
Ranking of features by importance.

distribution_ : str
Distribution function of the fitted model.

validation_scores_ : pandas.DataFrame()
Misclassification rate for the validation sample by iteration.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
decision_function	Computes the decision function given input features.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predicts class labels given input features.
predict_proba	Estimates probabilities for each class label.
save	Saves the model to a file.
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be "all" or "none". Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword ("none", "all"), a single table, or list of tables. The default is "all", which keeps all tables. The "none" keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the "exclude_tables" option, then the "include_tables" option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword ("none", "graph_tables", "all"), a single table, or list of tables. The default is "none", so that all tables are shown. The "graph_tables" keyword drops all tables that were used to produce a graph. The "all" keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the "include_tables" option, then the "include_tables" option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the

"include_tables" or "exclude_tables" option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the "order" option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

decision_function

```
decision_functionX
```

Computes the decision function given input features.

Parameters

X : array-like
Specifies training features in the shape (n_samples, n_features).

Returns

array-like
Estimated posterior probability for the target class. This is (n_samples, n_classes) for the multi-class case, and (n_samples,) with just self.classes_[1] returned for the binary case.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Specifies the names of columns in *X* that should be treated as categorical. For classifier models, the target is assumed to be categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

Predicts class labels given input features.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Predicted class labels in the shape (n_samples,).

predict_proba

```
predict_probaX
```

Estimates probabilities for each class label.

Parameters

X : array-like
Specifies the samples to predict class labels for, with shape (n_samples, n_features).

Returns

array-like
Estimated class label probabilities in the shape (n_samples, n_classes).

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self

GradientBoostingRegressor

Gradient boosting regression model.

API: GradientBoostingRegressor

class sasviya.ml.tree.GradientBoostingRegressor*, min_samples_leaf=5, n_bins=50, n_estimators=100, max_depth=4, max_features=None, learning_rate=0.1, subsample=0.5, l1_penalty=0, l2_penalty=1, random_state=None, verbose=0, n_iter_no_change=0, tol=0, calc_feature_importances=False, validation_fraction=0

Parameters

n_bins: int, optional
Specifies the number of bins to use for numeric variables. By default, 50.

min_samples_leaf : int, optional
Specifies the minimum number of observations on each node. By default, 5.

n_estimators : int, optional
Specifies the number of trees to create. By default, 100.

max_depth : int, optional
Specifies the maximum depth of the tree. By default, 4.

max_features : float, optional
Specifies the number of input variables to consider for splitting on a node. By default, the number of input variables is used.

learning_rate : float, optional
Specifies the learning rate of each tree. By default, 0.1.

subsample : float, optional
Specifies the fraction of the data to use for building each tree. By default, 0.5.

l1_penalty : float, optional
Specifies the L1 norm regularization on prediction. By default, 0.

l2_penalty : float, optional
Specifies the L2 norm regularization on prediction. By default, 1.

random_state : float, optional
Specifies the seed for the random number generator. By default, the random number stream is based on the computer clock. If you want a reproducible random number sequence between runs, specify a value that is greater than zero.

verbose : int, optional
Specifies the amount of information to display about the fitted model.

calc_feature_importances : bool, optional
Specifies whether to calculate feature importances.

validation_fraction: float, optional
Specifies the fraction of training sample to use for validation. A validation_fraction greater than zero enables early stopping mode.

n_iter_no_change : int, optional
Specifies the maximum number of training iterations for early stopping. Ignored unless a validation fraction is specified.

tol : float , optional
Specifies the tolerance value for early stopping as the change relative to the previous iteration. Ignored unless a validation fraction is specified.

Attributes

details_ : Results
Information about the model fitting process and the final results.

n_estimators_ : int
Number of trees in the fitted model. The number of trees could be fewer than requested due to early stopping.

avg_n_leaves_ : float
Average number of leaves among trees in the fitted model.

min_n_leaves_ : int
Minimum number of leaves among trees in the fitted model.

max_n_leaves_ : int
Maximum number of leaves among trees in the fitted model.

max_n_nodes_ : int
Maximum number of nodes among trees in the fitted model.

min_n_nodes_ : int
Minimum number of nodes among trees in the fitted model.

max_depth_ : int
Maximum depth among trees in the fitted model.

min_depth_ : int
Minimum depth among trees in the fitted model.

min_samples_leaf_ : int
Minimum leaf size among trees in the fitted model.

max_samples_leaf_ : int
Maximum leaf size among trees in the fitted model.

n_features_in_ : int
The number of input features.

feature_names_in_ : list of str
The column names of the input features.

n_features_used_ : int
The number of features used by the fitted model.

target_name_in_ : str
Column name of the target variable.

feature_importances_ : pandas.DataFrame()
Ranking of features by importance.

distribution_ : str
Distribution function of the fitted model.

validation_scores_ : pandas.DataFrame()
Average squared error for the validation sample by iteration.

Methods

Method Name	Description
customize_results	Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.
describe	Describes the type of model, its inputs and its output.
export	Exports the internal SAS analytic store from a fitted model.
fit	Fits the model on the training data.
get_params	Get parameters for this model.
predict	Predict the results of a data set based on the model.
save	Saves the model to a file.

Method Name	Description
score	Determine model accuracy by using data set X and targets y.
set_params	Updates the parameters of this model.

customize_results

```
customize_resultsplots=Nonewidth=Noneheight=Nonedpi=Nonetheme=Nonepath=Noneoutput_type=Noneinclude_tables=Noneexclude_tables=Noneorder=None
```

Generate plots for all actions that support graphics, as well as filter tables from the result list and create custom result orders.

Parameters

plots : str or str list
Request plots. This can be a single plot or list of plots. This request can also be "all" or "none". Parenthesized arguments are also allowed with each plot request. In addition, automatic plots from models can be disabled by setting the environment variable AUTO_GRAPHICS=false. However, this variable does not disable automatic plot generation through the customize_results() method. You will need to set plots="none" on the method to guaranteed that only tables are returned.

width : int
Overrides the default width of the graph in pixels.

height : int
Overrides the default height of the graph in pixels.

dpi : int
Overrides the default dots per inch of the graph. The default is 96 dpi.

theme : {"light", "dark", "opal", "midnight", "raven", "htmlencore", "highcontrast", "grayscale", monochrome"}
Overrides the default theme used in the graph. The default is "light". The session default can be set by setting the GRAPHICS_THEME environment variable.

path : str
If specified, the plot results are written to the specified path in addition to the notebook.

output_type : {"svg", "png"}
Overrides the default output type. In Jupyter notebooks, the default output is "svg", which contains image map information for tool tips. Otherwise, the default is "png", which contains no image map information.

include_tables : str or str list
Includes only the specified tables from the display results. This can be a keyword (“none”, “all”), a single table, or list of tables. The default is “all”, which keeps all tables. The “none” keyword drops all tables, so that the results contain only plots that were generated. You can include tables by name (case-sensitive), and wildcarding is supported. These names can seen in the top-left corner of all displayed results. If this option is specified with the “exclude_tables” option, then the “include_tables” option has precedence.

exclude_tables : str or str list
Excludes tables from the display results. This can be a keyword (“none”, “graph_tables”, “all”), a single table, or list of tables. The default is “none”, so that all tables are shown. The “graph_tables” keyword drops all tables that were used to produce a graph. The “all” keyword drops all tables, so that the results contain only plots that were generated. You can also exclude tables by name (case-sensitive), and wildcarding is supported. Those names can seen in the top-left corner of all displayed results. If this option is specified with the “include_tables” option, then the “include_tables” option has precedence.

order : str or str list
Orders plot and tabular results for display. Any result not specified in the option will be excluded in the new results. The names for the results can seen in the top-left corner of all displayed results. If this option is specified along with the “include_tables” or “exclude_tables” option, those options are applied BEFORE the ordering is done. If the ordered list contains no results, then the “order” option is ignored.

Returns

CustomResults
Returns CustomResults that are a combination of both model results and plot results. These results do not replace the original model results. Instead, The CustomResults are typically assigned to a variable for later usage or passed passed directly into the display() method for viewing in a Jupyter notebook cell.

describe

```
describe
```

Describes the type of model, its inputs and its output.

Returns

Results
a Mapping of various properties that describe aspects of the model.

export

```
exportfile=Nonereplace=False
```

Exports the internal SAS analytic store from a fitted model.

Generates an analytic store file that is common with other SAS products/PROCS.

Parameters

file : str, pathlib.Path, or file-like, optional
Specifies the location where the exported model will be saved. May be a path specifying a file or an existing file-like object. Must be writable in binary format.

replace : bool, optional
Specifies whether to overwrite *file* if it already exists. Ignored unless *file* specifies a file path.

Returns

bytes or None
The analytic store in binary form if *file* is not provided, otherwise None.

fit

```
fitXynominals=None
```

Fits the model on the training data.

Parameters

X : array-like
Specifies the training features in the shape (n_samples, n_features).

y : array-like
Specifies the target labels in the shape (n_samples,).

nominals : list of str, optional
Specifies the names of columns in *X* that should be treated as categorical. For classifier models, the target is assumed to be categorical.

Returns

self

get_params

```
get_paramsdeep=True
```

Get parameters for this model.

Parameters

deep : bool, default=True
Unused but provided for compatibility with scikit-learn.

Returns

dict
Parameter names and current values.

predict

```
predictX
```

save

```
savepath
```

Saves the model to a file.

Save uses pickle and can be loaded with sasviya.load_model.

Parameters

path : str or pathlib.Path
Specifies the location where model will be saved.

Returns

None

score

```
scoreXy
```

set_params

```
set_params**params
```

Updates the parameters of this model.

Parameters

****params : dict**
Specifies name:value pairs of the parameters to update.

Returns

self