

Export a model to Python or C++

Export a model to JSON

Object importances

> Training parameters

✓ Python package

Quick start

> CatBoost

✓ CatBoostClassifier

Overview

fit

predict

predict_proba

Attributes

calc_leaf_indexes

calc_feature_statistics

compare

copy

eval_metrics

get_all_params

CatBoostClassifier

```
class CatBoostClassifier(iterations=None,
                          learning_rate=None,
                          depth=None,
                          l2_leaf_reg=None,
                          model_size_reg=None,
                          rsm=None,
                          loss_function=None,
                          border_count=None,
                          feature_border_type=None,
                          per_float_feature_quantization=None,
                          input_borders=None,
                          output_borders=None,
                          fold_permutation_block=None,
                          od_pval=None,
                          od_wait=None,
                          od_type=None,
                          nan_mode=None,
                          counter_calc_method=None,
                          leaf_estimation_iterations=None,
                          leaf_estimation_method=None,
                          thread_count=None,
                          random_seed=None,
                          use_best_model=None,
                          verbose=None,
                          logging_level=None,
                          metric_period=None,
                          ctr_leaf_count_limit=None,
                          store_all_simple_ctr=None,
                          max_ctr_complexity=None,
                          has_time=None,
                          allow_const_label=None,
                          classes_count=None,
                          class_weights=None,
                          auto_class_weights=None,
                          one_hot_max_size=None,
                          random_strength=None,
                          name=None,
                          ignored_features=None,
                          train_dir=None,
                          custom_loss=None,
                          custom_metric=None,
                          eval_metric=None,
                          bagging_temperature=None,
                          save_snapshot=None,
                          snapshot_file=None,
                          snapshot_interval=None,
```



In this article:

- get_params
- get_scale_and_bias
- get_test_eval
- grid_search
- is_fitted
- load_model
- plot_predictions
- plot_tree
- randomized_search
- save_borders
- save_model
- score
- select_features
- set_feature_names
- set_params
- set_scale_and_bias**
- shrink
- staged_predict
- staged_predict_proba

Export a model to Python or C++

Export a model to JSON

Object importances

> Training parameters

✓ Python package

Quick start

> CatBoost

✓ CatBoostClassifier

Overview

fit

predict

predict_proba

Attributes

calc_leaf_indexes

calc_feature_statistics

compare

copy

eval_metrics

get_all_params

```
fold_len_multiplier=None,
used_ram_limit=None,
gpu_ram_part=None,
allow_writing_files=None,
final_ctr_computation_mode=None,
approx_on_full_history=None,
boosting_type=None,
simple_ctr=None,
combinations_ctr=None,
per_feature_ctr=None,
task_type=None,
device_config=None,
devices=None,
bootstrap_type=None,
subsample=None,
sampling_unit=None,
dev_score_calc_obj_block_size=None,
max_depth=None,
n_estimators=None,
num_boost_round=None,
num_trees=None,
colsample_bylevel=None,
random_state=None,
reg_lambda=None,
objective=None,
eta=None,
max_bin=None,
scale_pos_weight=None,
gpu_cat_features_storage=None,
data_partition=None,
metadata=None,
early_stopping_rounds=None,
cat_features=None,
grow_policy=None,
min_data_in_leaf=None,
min_child_samples=None,
max_leaves=None,
num_leaves=None,
score_function=None,
leaf_estimation_backtracking=None,
ctr_history_unit=None,
monotone_constraints=None,
feature_weights=None,
penalties_coefficient=None,
first_feature_use_penalties=None,
model_shrink_rate=None,
model_shrink_mode=None,
langevin=None,
diffusion_temperature=None,
posterior_sampling=None,
boost_from_average=None,
text_features=None,
tokenizers=None,
dictionaries=None,
```



In this article:

get_params

get_scale_and_bias

get_test_eval

grid_search

is_fitted

load_model

plot_predictions

plot_tree

randomized_search

save_borders

save_model

score

select_features

set_feature_names

set_params

set_scale_and_bias

shrink

staged_predict

staged_predict_proba

- Export a model to Python or C++
- Export a model to JSON
- Object importances
- > Training parameters
- ✓ Python package
 - Quick start
 - > CatBoost
 - ✓ CatBoostClassifier
 - Overview
 - fit
 - predict
 - predict_proba
 - Attributes
 - calc_leaf_indexes
 - calc_feature_statistics
 - compare
 - copy
 - eval_metrics
 - get_all_params

```
feature_calcers=None,  
text_processing=None)
```



Purpose

Training and applying models for the classification problems. Provides compatibility with the scikit-learn tools.

The default optimized objective depends on various conditions:

- Logloss — The target has only two different values or the `target_border` parameter is not None.
- MultiClass — The target has more than two different values and the `border_count` parameter is None.

Parameters

metadata

Description

The key-value string pairs to store in the model's metadata storage after the training.

Default value

None

cat_features

Description

A one-dimensional array of categorical columns indices (specified as integers) or names (specified as strings).

This array can contain both indices and names for different elements.

If any features in the `cat_features` parameter are specified as names instead of indices, feature names must be provided for the training dataset. Therefore, the type of the `X` parameter in the future calls of the `fit` function must be either `catboost.Pool` with defined feature names data or `pandas.DataFrame` with defined column names.



Note

- If this parameter is not None and the training dataset passed as the value of the `X` parameter to the `fit` function of this class has the `catboost.Pool` type, CatBoost checks the equivalence of the categorical features indices specification in this object and the one in the `catboost.Pool` object.
- If this parameter is not None, passing objects of the `catboost.FeaturesData` type as the `X` parameter to the `fit` function of this class is prohibited.

Default value

None (all features are either considered numerical or of other types if specified precisely)

In this article:

- get_params
- get_scale_and_bias
- get_test_eval
- grid_search
- is_fitted
- load_model
- plot_predictions
- plot_tree
- randomized_search
- save_borders
- save_model
- score
- select_features
- set_feature_names
- set_params
- set_scale_and_bias
- shrink
- staged_predict
- staged_predict_proba

Export a model to Python or C++

Export a model to JSON

Object importances

> Training parameters

✓ Python package

Quick start

> CatBoost

✓ CatBoostClassifier

Overview

fit

predict

predict_proba

Attributes

calc_leaf_indexes

calc_feature_statistics

compare

copy

eval_metrics

get_all_params

text_features

Description

A one-dimensional array of text columns indices (specified as integers) or names (specified as strings).

Use only if the data parameter is a two-dimensional feature matrix (has one of the following types: list, numpy.ndarray, pandas.DataFrame, pandas.Series).

If any elements in this array are specified as names instead of indices, names for all columns must be provided. To do this, either use the `feature_names` parameter of this constructor to explicitly specify them or pass a pandas.DataFrame with column names specified in the data parameter.

Default value

None (all features are either considered numerical or of other types if specified precisely)

See [Python package training parameters](#) for the full list of parameters.



Note

Some parameters duplicate the ones specified for the `fit` method. In these cases the values specified for the `fit` method take precedence.

Attributes

tree_count_

Return the number of trees in the model.

This number can differ from the value specified in the `--iterations` training parameter in the following cases:

- The training is stopped by the [overfitting detector](#).
- The `--use-best-model` training parameter is set to "True".

feature_importances_

Return the calculated [feature importances](#). The output data depends on the type of the model's loss function:

- Non-ranking loss functions — [PredictionValuesChange](#)
- Ranking loss functions — [LossFunctionChange](#)

random_seed_

The random seed used for training.

learning_rate_



In this article:

[get_params](#)
[get_scale_and_bias](#)
[get_test_eval](#)
[grid_search](#)
[is_fitted](#)
[load_model](#)
[plot_predictions](#)
[plot_tree](#)
[randomized_search](#)
[save_borders](#)
[save_model](#)
[score](#)
[select_features](#)
[set_feature_names](#)
[set_params](#)
[set_scale_and_bias](#)
[shrink](#)
[staged_predict](#)
[staged_predict_proba](#)

- Export a model to Python or C++
- Export a model to JSON
- Object importances
- > Training parameters
- ✓ Python package
 - Quick start
 - > CatBoost
 - ✓ CatBoostClassifier
 - Overview
 - fit
 - predict
 - predict_proba
 - Attributes
 - calc_leaf_indexes
 - calc_feature_statistics
 - compare
 - copy
 - eval_metrics
 - get_all_params

The learning rate used for training.

feature_names_

The names of features in the dataset.

evals_result_

Return the values of metrics calculated during the training.

best_score_

Return the best result for each metric calculated on each validation dataset.

best_iteration_

Return the identifier of the iteration with the best result of the evaluation metric or loss function on the last validation set.

classes_

Return the names of classes for classification models. An empty list is returned for all other models.

The order of classes in this list corresponds to the order of classes in resulting predictions.

Methods

fit

Train a model.

predict

Apply the model to the given dataset.

predict_proba

Apply the model to the given dataset to predict the probability that the object belongs to the given classes.

calc_leaf_indexes

Returns indexes of leafs to which objects from pool are mapped by model trees.

calc_feature_statistics

Calculate and plot a set of statistics for the chosen feature.



In this article:

- get_params
- get_scale_and_bias
- get_test_eval
- grid_search
- is_fitted
- load_model
- plot_predictions
- plot_tree
- randomized_search
- save_borders
- save_model
- score
- select_features
- set_feature_names
- set_params
- set_scale_and_bias**
- shrink
- staged_predict
- staged_predict_proba

Export a model to Python or C++

Export a model to JSON

Object importances

> Training parameters

✓ Python package

Quick start

> CatBoost

✓ CatBoostClassifier

Overview

fit

predict

predict_proba

Attributes

calc_leaf_indexes

calc_feature_statistics

compare

copy

eval_metrics

get_all_params

compare

Draw train and evaluation metrics in [Jupyter Notebook](#) for two trained models.

copy

Copy the CatBoost object.

eval_metrics

Calculate the specified metrics for the specified dataset.

get_all_params

Return the values of all training parameters (including the ones that are not explicitly specified by users).

get_best_iteration

Return the identifier of the iteration with the best result of the evaluation metric or loss function on the last validation set.

get_best_score

Return the best result for each metric calculated on each validation dataset.

get_borders

Return the list of borders for numerical features.

get_evals_result

Return the values of metrics calculated during the training.

get_feature_importance

Calculate and return the [feature importances](#).

get_metadata

Return a proxy object with metadata from the model's internal key-value string storage.

get_object_importance

Calculate the effect of objects from the train dataset on the optimized metric values for the objects from the input dataset:

- Positive values reflect that the optimized metric increases.



In this article:

- get_params
- get_scale_and_bias
- get_test_eval
- grid_search
- is_fitted
- load_model
- plot_predictions
- plot_tree
- randomized_search
- save_borders
- save_model
- score
- select_features
- set_feature_names
- set_params
- set_scale_and_bias**
- shrink
- staged_predict
- staged_predict_proba

- Negative values reflect that the optimized metric decreases.

Export a model to Python or C++

Export a model to JSON

Object importances

> Training parameters

✓ Python package

Quick start

> CatBoost

✓ CatBoostClassifier

Overview

fit

predict

predict_proba

Attributes

calc_leaf_indexes

calc_feature_statistics

compare

copy

eval_metrics

get_all_params

get_param

Return the value of the given parameter if it is explicitly by the user before starting the training. If this parameter is used with the default value, this function returns None.

get_params

Return the values of training parameters that are explicitly specified by the user. If all parameters are used with their default values, this function returns an empty dict.

get_scale_and_bias

Return the scale and bias of the model.

These values affect the results of applying the model, since the model prediction results are calculated as follows:

$$\sum leaf_values \cdot scale + bias$$

get_test_eval

Return the formula values that were calculated for the objects from the validation dataset provided for training.

grid_search

A simple grid search over specified parameter values for a model.

is_fitted

Check whether the model is trained.

load_model

Load the model from a file.

plot_predictions

Sequentially vary the value of the specified features to put them into all buckets and calculate predictions for the input objects accordingly.

plot_tree

Visualize the CatBoost decision trees.

randomized_search

A simple randomized search on hyperparameters.



In this article:

get_params

get_scale_and_bias

get_test_eval

grid_search

is_fitted

load_model

plot_predictions

plot_tree

randomized_search

save_borders

save_model

score

select_features

set_feature_names

set_params

set_scale_and_bias

shrink

staged_predict

staged_predict_proba

Export a model to Python or C++

Export a model to JSON

Object importances

> Training parameters

✓ Python package

Quick start

> CatBoost

✓ CatBoostClassifier

Overview



CatBoost

predict_proba

Attributes

calc_leaf_indexes

calc_feature_statistics

compare

copy

eval_metrics

get_all_params

save_borders

Save the model borders to a file.

save_model

Save the model to a file.

score

Calculate the Accuracy [metric](#) for the objects in the given dataset.

select_features

set_feature_names

Set names for all features in the model.

set_params

Set the training parameters.

set_scale_and_bias

Set the scale and bias.

shrink

Shrink the model. Only trees with indices from the range `[ntree_start, ntree_end)` are kept.

staged_predict

Apply the model to the given dataset and calculate the results taking into consideration only the trees in the range `[0; i)`.

staged_predict_proba

Apply the model to the given dataset to predict the probability that the object belongs to the class and calculate the results taking into consideration only the trees in the range `[0; i)`.



In this article:

get_params
get_scale_and_bias
get_test_eval
grid_search
is_fitted
load_model



randomized_search
save_borders
save_model
score
select_features
set_feature_names
set_params
set_scale_and_bias
shrink
staged_predict
staged_predict_proba

Previous

Was the article helpful?

 Yes

 No

Next

← virtual_ensembles_predict

fit →



In this article:

- get_params
- get_scale_and_bias
- get_test_eval
- grid_search
- is_fitted
- load_model
- plot_predictions
- plot_tree
- randomized_search
- save_borders
- save_model
- score
- select_features
- set_feature_names
- set_params
- set_scale_and_bias**
- shrink
- staged_predict
- staged_predict_proba

Export a model to Python or C++

Export a model to JSON

Object importances

> Training parameters

✓ Python package

Quick start

> CatBoost

✓ CatBoostClassifier

Overview

fit

predict

predict_proba

Attributes

calc_leaf_indexes

calc_feature_statistics

compare

copy

eval_metrics

get_all_params