

Name: Manasi Amdekar
Roll No.: J003

GridSearchCV

Code:

```
sklearn.model_selection.GridSearchCV(estimator, param_grid, *, scoring=None, n_jobs=None, refit=True, cv=None, verbose=0, pre_dispatch='2*n_jobs', error_score=nan, return_train_score=False)
```

What is GridSearchCV?

GridSearchCV is a library function that is a member of sklearn's model_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

In addition to that, you can specify the number of times for the cross-validation for each set of hyperparameters.

Its important members are fit, predict.

GridSearchCV implements a "fit" and a "score" method. It also implements "score_samples", "predict", "predict_proba", "decision_function", "transform" and "inverse_transform" if they are implemented in the estimator used.

The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

For given values with the param_grid parameter, GridSearchCV exhaustively considers all parameter combinations. It has a successive halving counterpart, HalvingGridSearchCV, which can be much faster at finding a good parameter combination.

The GridSearchCV instance implements the usual estimator API: when "fitting" it on a dataset all the possible combinations of parameter values are evaluated and the best combination is retained.

Parameters

estimator estimator object.

This is assumed to implement the scikit-learn estimator interface. Either estimator needs to provide a score function, or scoring must be passed.

param_grid dict or list of dictionaries

Dictionary with parameters names (str) as keys and lists of parameter settings to try as values, or a list of such dictionaries, in which case the grids spanned by each dictionary in the list are explored. This enables searching over any sequence of parameter settings.

scoring str, callable, list, tuple or dict, default=None

Strategy to evaluate the performance of the cross-validated model on the test set.

If scoring represents a single score, one can use:

- a single string;
- a callable that returns a single value.

If scoring represents multiple scores, one can use:

- a list or tuple of unique strings;
- a callable returning a dictionary where the keys are the metric names and the values are the metric scores;
- a dictionary with metric names as keys and callables as values.

n_jobsint, default=None

Number of jobs to run in parallel. None means 1 unless in a [joblib.parallel_backend](#) context. - 1 means using all processors.

refitbool, str, or callable, default=True

Refit an estimator using the best found parameters on the whole dataset.

For multiple metric evaluation, this needs to be a str denoting the scorer that would be used to find the best parameters for refitting the estimator at the end.

Where there are considerations other than maximum score in choosing a best estimator, refit can be set to a function which returns the selected best_index_ given cv_results_. In that case, the best_estimator_ and best_params_ will be set according to the returned best_index_ while the best_score_ attribute will not be available.

The refitted estimator is made available at the best_estimator_ attribute and permits using predict directly on this GridSearchCV instance.

Also for multiple metric evaluation, the attributes best_index_, best_score_ and best_params_ will only be available if refit is set and all of them will be determined w.r.t this specific scorer.

cvint, cross-validation generator or an iterable, default=None

Determines the cross-validation splitting strategy. Possible inputs for cv are:

None, to use the default 5-fold cross validation,

integer, to specify the number of folds in a (Stratified)KFold,

verboseint

Controls the verbosity: the higher, the more messages.

>1 : the computation time for each fold and parameter candidate is displayed;

>2 : the score is also displayed;

>3 : the fold and candidate parameter indexes are also displayed together with the starting time of the computation.

error_score'raise' or numeric, default=np.nan

Value to assign to the score if an error occurs in estimator fitting. If set to 'raise', the error is raised. If a numeric value is given, FitFailedWarning is raised. This parameter does not affect the refit step, which will always raise the error.

return_train_scorebool, default=False

If False, the cv_results_ attribute will not include training scores. Computing training scores is used to get insights on how different parameter settings impact the overfitting/underfitting trade-off. However computing the scores on the training set can be computationally expensive and is not strictly required to select the parameters that yield the best generalization performance.

Attributes

best_estimator_estimator

Estimator that was chosen by the search, i.e. estimator which gave highest score (or smallest loss if specified) on the left out data. Not available if refit=False.

best_score_float

Mean cross-validated score of the best_estimator

For multi-metric evaluation, this is present only if refit is specified.

This attribute is not available if refit is a function.

best_params_dict

Parameter setting that gave the best results on the hold out data.

For multi-metric evaluation, this is present only if refit is specified.

best_index_int

The index (of the cv_results_ arrays) which corresponds to the best candidate parameter setting.

The dict at search.cv_results_['params'][search.best_index_] gives the parameter setting for the best model, that gives the highest mean score (search.best_score_).

For multi-metric evaluation, this is present only if refit is specified.

scorer_function or a dict

Scorer function used on the held out data to choose the best parameters for the model.

For multi-metric evaluation, this attribute holds the validated scoring dict which maps the scorer key to the scorer callable.

n_splits_int

The number of cross-validation splits (folds/iterations).

refit_time_float

Seconds used for refitting the best model on the whole dataset.

This is present only if refit is not False.