GridSearchCV API

Aryaman Gautam

J001

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

class 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)

Parameters :

- estimatorestimator object.

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

Ex

- param_griddict 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 a values.

See Specifying multiple metrics for evaluation for an example.

- n_jobs int, default=None

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

- refit bool, str, or callable, default=True

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

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

- cv int, cross-validation generator or an iterable, default=None

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,
- CV splitter
- An iterable yielding (train, test) splits as arrays of indices.

- verbose : int

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.

- pre_dispatch int, or str, default=n_jobs

Controls the number of jobs that get dispatched during parallel execution. Reducing this number can be useful to avoid an explosion of memory consumption when more jobs get dispatched than CPUs can process. This parameter can be:

- None,
- An int, giving the exact number of total jobs that are spawned
- A str, giving an expression as a function of n_jobs, as in '2*n_jobs'

- 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.

- return_train_score bool, default=False

Attributes :

- cv_results_ dict of numpy (masked) ndarrays
- best_estimator_ estimator
- best_score_ float
- best_params_ dict
- best_index_ int
- scorer_ function or a dict
- n_splits_ int
- refit_time_ float
- multimetric_ bool