sklearn.model_selection.GridSearchCV

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*)[source]

Parameters

estimator- 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- Dictionary with parameters names (`str`) as keys and lists of parameter settings to try as values

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

n_jobs- Number of cores to use in parallel for the grid search

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

cv- Cross validation strategy

pre_dispatch- Controls the number of jobs that get dispatched during parallel execution.

return_train_score- Whether to return the training scores or not

Attributes

cv_results_- A dict with keys as column headers and values as columns, that can be imported into a pandas `DataFrame`.

best_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_- Mean cross-validated score of the best_estimator

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

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

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

multimetric_- Whether or not the scorers compute several metrics

Methods

`decision_function`(X)	Call decision_function on the estimator with the best found parameters.
`fit`(X[, y, groups])	Run fit with all sets of parameters.
`get_params`([deep])	Get parameters for this estimator.
`inverse_transform`(Xt)	Call inverse_transform on the estimator with the best found params.
`predict`(X)	Call predict on the estimator with the best found parameters.
`predict_log_proba`(X)	Call predict_log_proba on the estimator with the best found parameters.
`predict_proba`(X)	Call predict_proba on the estimator with the best found parameters.
`score`(X[, y])	Returns the score on the given data, if the estimator has been refit.
`score_samples`(X)	Call score_samples on the estimator with the best found parameters.
`set_params`(**params)	Set the parameters of this estimator.
`transform`(X)	Call transform on the estimator with the best found parameters.

How GridSearchCV works?

i. GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model.

ii. GridSearchCV is a function that comes in Scikit-learn's(or SK-learn) model_selection package.

iii. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.

iv. We pass predefined values for hyperparameters to the GridSearchCV function.

v. This is done by defining a dictionary in which we mention a particular hyperparameter along with the values it can take.

vi. GridSearchCV tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using the Cross-Validation method.

vii. Hence after using this function we get accuracy/loss for every combination of hyperparameters and we can choose the one with the best performance.