DEVYANI BEOHAR

J068

GridSearchCV

Code:

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*)

estimator *object.*

This is assumed to implement the scikit-learn estimator interface.

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.

scoringstr, callable, list, tuple or dict, default=None

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

n_jobsint, default=None

Number of jobs to run in parallel.

refitbool, str, or callable, default=True

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

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

Determines the cross-validation splitting strategy.

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.

pre_dispatchint, or str, default=n_jobs

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

error_score‘raise’ or numeric, default=np.nan

Value to assign to the score if an error occurs in estimator fitting.

return_train_scorebool, default=False

If False, the cv_results_ attribute will not include training scores.

It is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. GridSearchCV is a function that comes in Scikit-learn’s(or SK-learn) model_selection package.So an important point here to note is that we need to have Scikit-learn library installed on the computer. 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.

We pass predefined values for hyperparameters to the GridSearchCV function. We do this by defining a dictionary in which we mention a particular hyperparameter along with the values it can take.

GridSearchCV tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using the Cross-Validation method. Hence after using this function we get accuracy/loss for every combination of hyperparameters and we can choose the one with the best performance.