# [sklearn.model\_selection](https://scikit-learn.org/stable/modules/classes.html#module-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]](https://github.com/scikit-learn/scikit-learn/blob/2beed5584/sklearn/model_selection/_search.py#L975)

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**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.decision_function)(X) | Call decision\_function on the estimator with the best found parameters. |
| [**fit**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.fit)(X[, y, groups]) | Run fit with all sets of parameters. |
| [**get\_params**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.get_params)([deep]) | Get parameters for this estimator. |
| [**inverse\_transform**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.inverse_transform)(Xt) | Call inverse\_transform on the estimator with the best found params. |
| [**predict**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.predict)(X) | Call predict on the estimator with the best found parameters. |
| [**predict\_log\_proba**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.predict_log_proba)(X) | Call predict\_log\_proba on the estimator with the best found parameters. |
| [**predict\_proba**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.predict_proba)(X) | Call predict\_proba on the estimator with the best found parameters. |
| [**score**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.score)(X[, y]) | Returns the score on the given data, if the estimator has been refit. |
| [**score\_samples**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.score_samples)(X) | Call score\_samples on the estimator with the best found parameters. |
| [**set\_params**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.set_params)(\*\*params) | Set the parameters of this estimator. |
| [**transform**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.transform)(X) | Call transform on the estimator with the best found parameters. |

**How GridSearchCV works?**

1. GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model.
2. GridSearchCV is a function that comes in Scikit-learn’s(or SK-learn) model\_selection package.
3. 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.
4. We pass predefined values for hyperparameters to the GridSearchCV function.
5. This is done by defining a dictionary in which we mention a particular hyperparameter along with the values it can take.
6. GridSearchCV tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using the [Cross-Validation](https://www.mygreatlearning.com/blog/cross-validation/) method.
7. Hence after using this function we get accuracy/loss for every combination of hyperparameters and we can choose the one with the best performance.