GridSearchCV API

Siddhanth Nair

J040

API:-

Exhaustive search over specified parameter values for an estimator.

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.

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)

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.
- **scoring: str, callable, list, tuple or dict, default=None**
  Strategy to evaluate the performance of the cross-validated model on the test set.

  A) If scoring represents a single score, one can use:
  i) a single string;
  ii) a callable that returns a single value.

  B) If scoring represents multiple scores, one can use:
  i) a list or tuple of unique strings;
  ii) a callable returning a dictionary where the keys are the metric names and the values are the metric scores;
  iii) a dictionary with metric names as keys and callables a values.

- **n_jobs: int, default=None**
  Number of jobs to run in parallel.
- **refit: bool, str, or callable, default=True**
  Refit an estimator using the best found parameters on the whole dataset.
- **cv: int, cross-validation generator or an iterable, default=None**
  Determines the cross-validation splitting strategy.
- **verbose: int**
  Controls the verbosity: the higher, the more messages.
  A) >1 : the computation time for each fold and parameter candidate is displayed;
  B) >2 : the score is also displayed;
  C) >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.
- **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_scorebool, default=False**
  If False, the cv_results_ attribute will not include training scores.

ATTRIBUTES:-

- **cv_results_:dict of numpy (masked) ndarrays**
  A dict with keys as column headers and values as columns, that can be imported into a pandas DataFrame.
- **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.
- **best_params_: dict**
  Parameter setting that gave the best results on the hold out data.
- **best_index_: int**
  The index (of the cv_results_ arrays) which corresponds to the best candidate parameter setting..
- **scorer_: function or a dict**
  Scorer function used on the held out data to choose the best parameters for the model.
- **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.

- **multimetric_: bool**
  Whether or not the scorers compute several metrics.