Machine Learning Assignment: GridSearchCV

Shehzeen Shuaeb Khan
J028, BTech Data Science, Sem 5

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- to implement the scikit-learn estimator interface; either the estimator needs to provide a score function, or a form of scoring must be passed;
- param_grid- a 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- acronym, 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 dictionary with keys as column headers and values as columns, that can be imported into a pandas dataframe;
- best_estimator_- an estimator that was chosen by the search, i.e., estimator which gave highest score (or smallest loss if specified) on the remaining 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_- a scorer function used on the "held out" data to choose the best parameters for the model;
- refit_time_- records 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 does GridSearchCV work?

- GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model that comes in Scikit-learn's(or SK-learn) model_selection package.
- This function loops through predefined hyperparameters and fits your estimator (model) to your training set so one can select the best parameters from the listed hyperparameters.
- Predefined values for hyperparameters are passed to the GridSearchCV function.
- This is done by defining a dictionary in which a particular hyperparameter along with the values it can take exists.
- GridSearchCV tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using the method of Cross-Validation.
- Therefore after using this function, we get accuracy/loss for every combination of hyperparameters and we can choose the one with the best performance.