10 Hyperparameter Tuning

Created Date: 2022-11-08

> Metadata 🧰
>
> - Title: Advanced Scikit Learn Applications
> - Author: Andrew Jones
> - Reference: Data Science Infinity

> Links & Tags 📎
>
> - Index: Course Note Index
> - Atomic Tag: #datascience
> - Subatomic Tags: #machinelearning #mltuning #hyperparameters

---

Grid Search for Hyperparameter Tuning

Jupyter Notebook: Hyperparameter Tuning

> Grid Search is a technique used to isolate the optimal hyperparameters of a model, which results in the most accurate predictions.

> Hyperparameters are the model settings we apply before training our model. Examples include;
>
> - k value in KNN
> - Max Depth in Decision Trees
> - Number of Trees in Random Forest

- Scikit allows us to test combinations of hyperparameters, assess the cross-validated accuracy, and apply the optimal value for the hyperparameter settings
- Gris Search allows us to specify multiple hyperparameter values and build models for all combinations of the values
    - Cross-validation is applied to determine accuracy scores for each model and we can choose the combination that produces the highest accuracy score

Applying Grid Search (Random Forest Regression)

- Replace estimator with appropriate model objects
- Replace param_grid with appropriate model parameters
- Replace scoring with desired scoring method

```
# instantiate grid search object
gscv = GridSearchCV(
    estimator = RandomForestRegressor(),
    param_grid = {
        'n_estimators': [10, 50, 100, 500],
        'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, None]
    },
    cv = 5,
    scoring = 'r2',
    n_jobs = -1
)


# fit to data
gscv.fit(X_train, y_train)


# return best CV score (mean)
gscv.best_score_
```

```
# return optimal parameters
gscv.best_params_


# instantiate regressor object with optimal parameters
regressor = gscv.best_estimator_
regressor
```