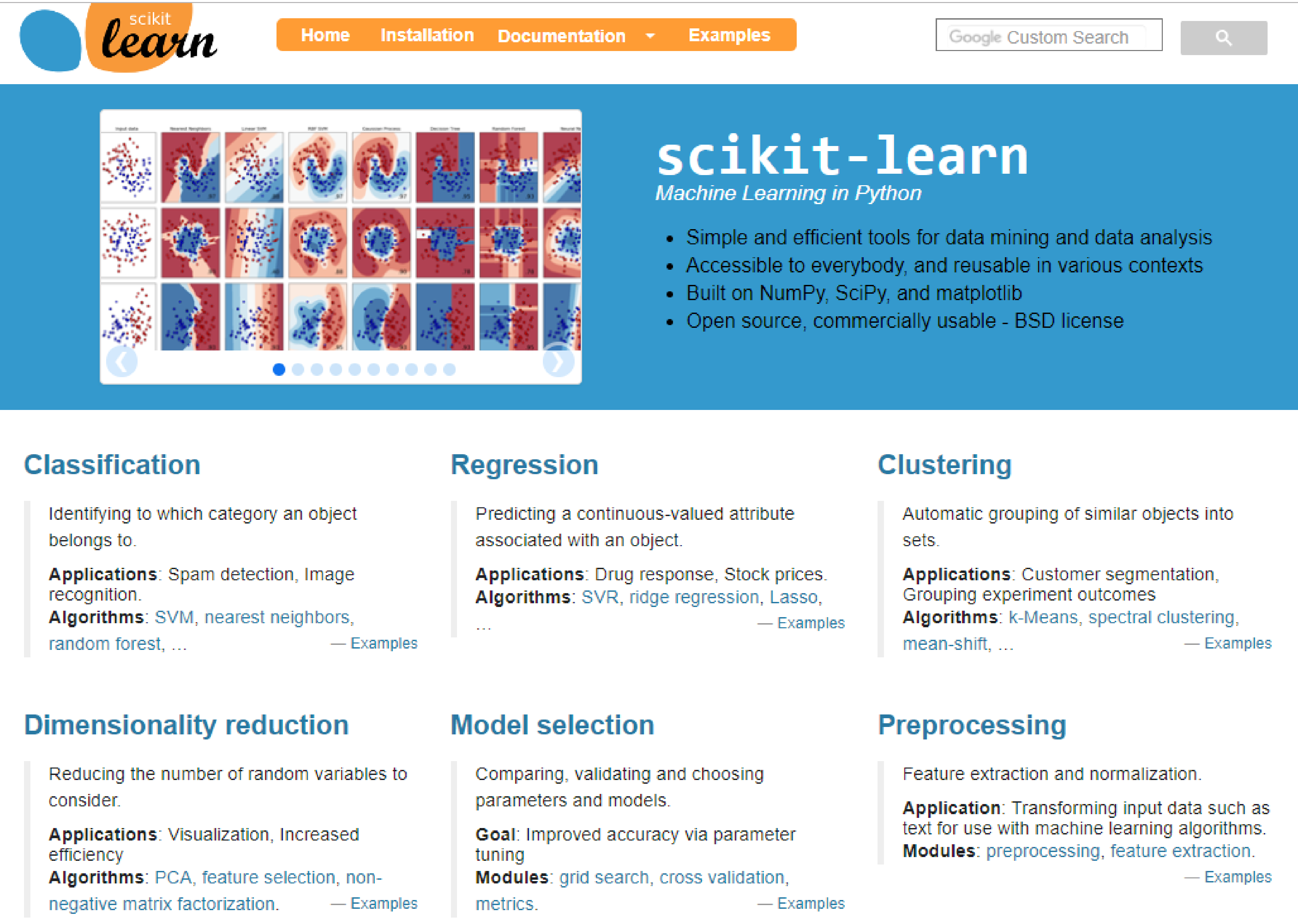
TA Session 2

2018.10.15

Wonjong Rhee, Hyunghun Cho, Daeyoung Choi
Seoul National University
Graduate School of Convergence Science and Technology
Applied Data Science Lab.



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

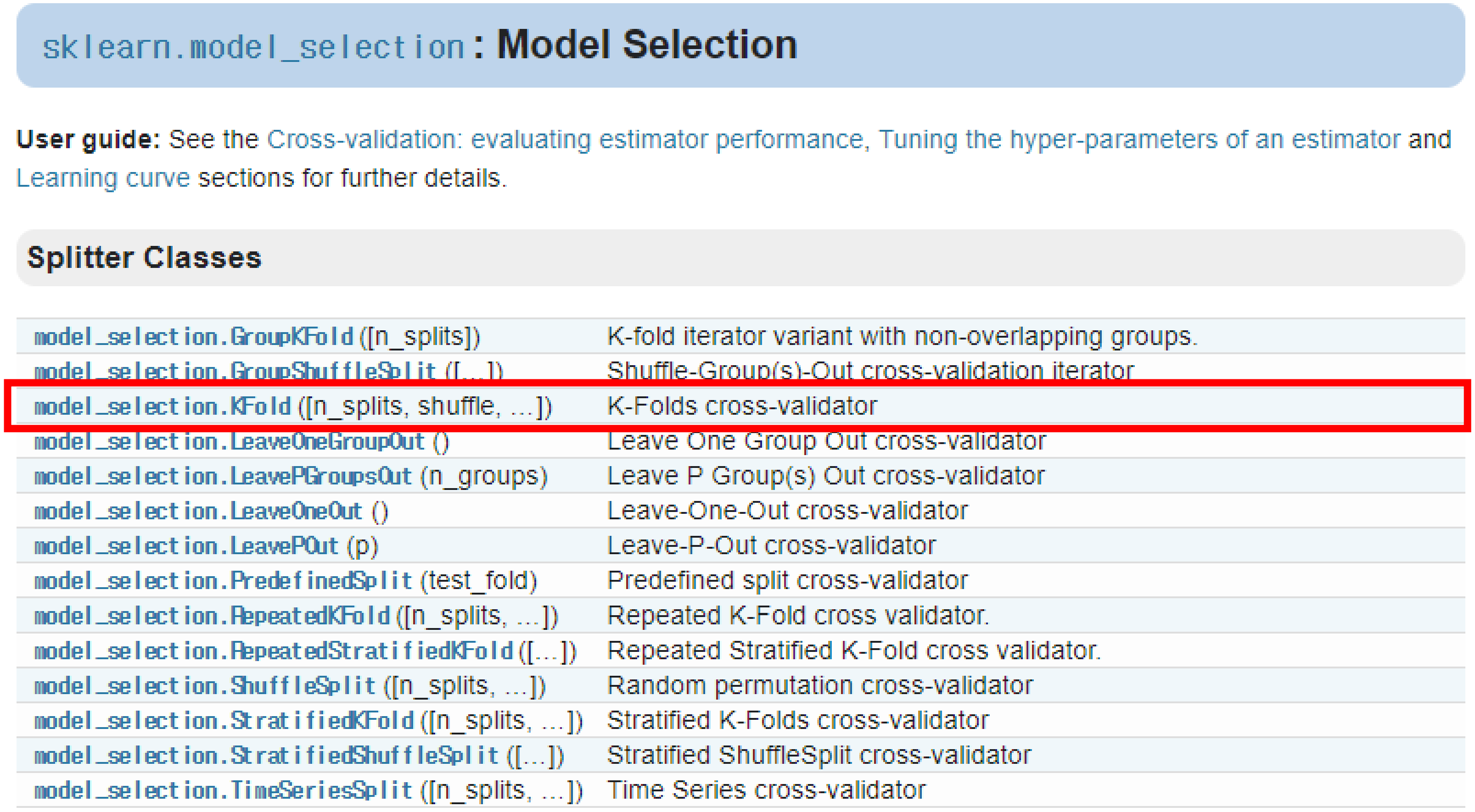
Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.
— Examples

scikit-learn



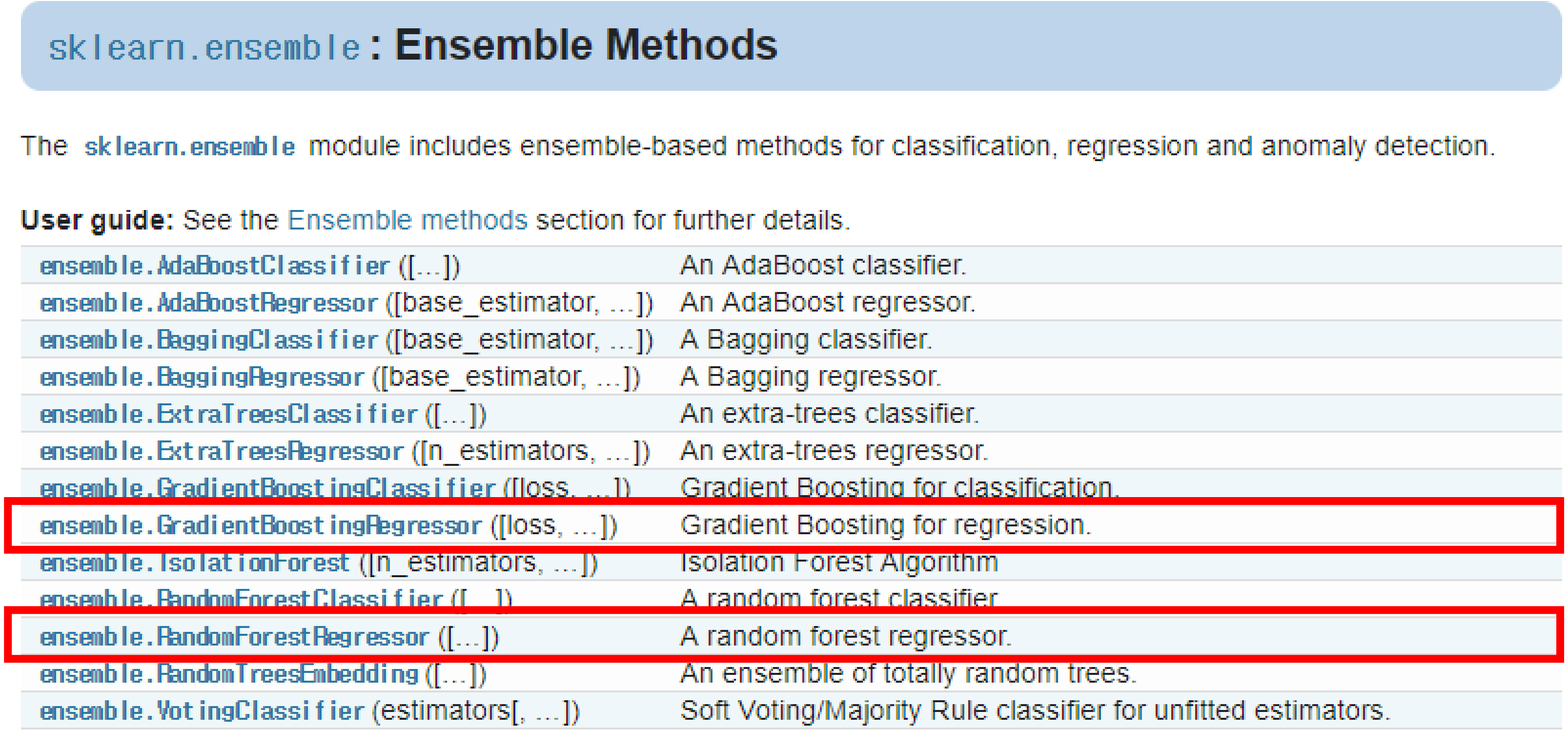
sklearn.model_selection : Model Selection

User guide: See the Cross-validation: evaluating estimator performance, Tuning the hyper-parameters of an estimator and Learning curve sections for further details.

Splitter Classes

model_selection.GroupKFold ([n_splits])	K-fold iterator variant with non-overlapping groups.
model_selection.GroupShuffleSplit ([…])	Shuffle-Group(s)-Out cross-validation iterator
model_selection.KFold ([n_splits, shuffle, …])	K-Folds cross-validator
model_selection.LeaveOneGroupOut ()	Leave One Group Out cross-validator
model_selection.LeavePGroupsOut (n_groups)	Leave P Group(s) Out cross-validator
model_selection.LeaveOneOut ()	Leave-One-Out cross-validator
model_selection.LeavePOut (p)	Leave-P-Out cross-validator
model_selection.PredefinedSplit (test_fold)	Predefined split cross-validator
model_selection.RepeatedKFold ([n_splits, …])	Repeated K-Fold cross validator.
model_selection.RepeatedStratifiedKFold ([…])	Repeated Stratified K-Fold cross validator.
model_selection.ShuffleSplit ([n_splits, …])	Random permutation cross-validator
model_selection.StratifiedKFold ([n_splits, …])	Stratified K-Folds cross-validator
model_selection.StratifiedShuffleSplit ([…])	Stratified ShuffleSplit cross-validator
model_selection.TimeSeriesSplit ([n_splits, …])	Time Series cross-validator

scikit-learn



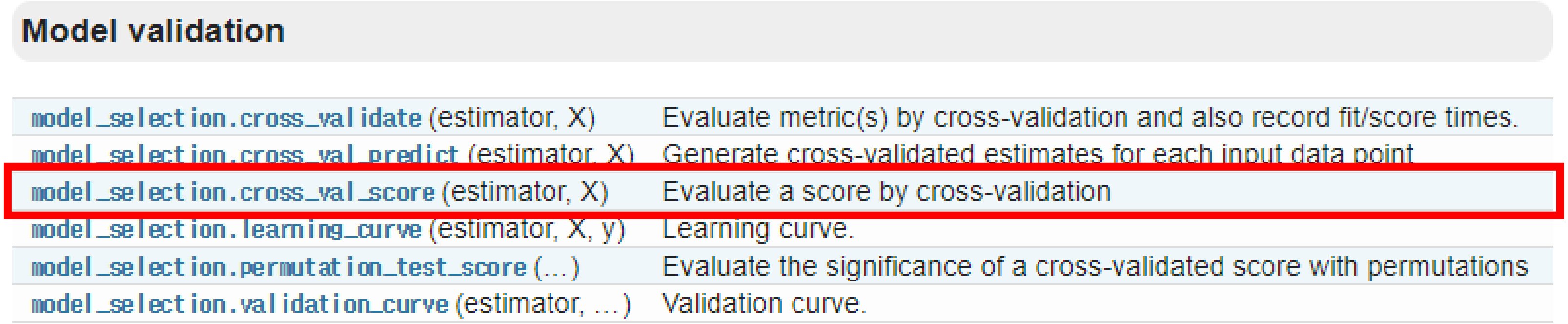
sklearn.ensemble: Ensemble Methods

The sklearn.ensemble module includes ensemble-based methods for classification, regression and anomaly detection.

User guide: See the Ensemble methods section for further details.

ensemble.AdaBoostClassifier ([…])	An AdaBoost classifier.
ensemble.AdaBoostRegressor ([base_estimator, …])	An AdaBoost regressor.
ensemble.BaggingClassifier ([base_estimator, …])	A Bagging classifier.
ensemble.BaggingRegressor ([base_estimator, …])	A Bagging regressor.
ensemble.ExtraTreesClassifier ([…])	An extra-trees classifier.
ensemble.ExtraTreesRegressor ([n_estimators, …])	An extra-trees regressor.
ensemble.GradientBoostingClassifier ([loss, …])	Gradient Boosting for classification.
ensemble.GradientBoostingRegressor ([loss, …])	Gradient Boosting for regression.
ensemble.IsolationForest ([n_estimators, …])	Isolation Forest Algorithm
ensemble.RandomForestClassifier ([…])	A random forest classifier.
ensemble.RandomForestRegressor ([…])	A random forest regressor.
ensemble.RandomTreesEmbedding ([…])	An ensemble of totally random trees.
ensemble.VotingClassifier (estimators[, …])	Soft Voting/Majority Rule classifier for unfitted estimators.

scikit-learn



Model validation

model_selection.cross_validate (estimator, X)	Evaluate metric(s) by cross-validation and also record fit/score times.
model_selection.cross_val_predict (estimator, X)	Generate cross-validated estimates for each input data point
model_selection.cross_val_score (estimator, X)	Evaluate a score by cross-validation
model_selection.learning_curve (estimator, X, y)	Learning curve.
model_selection.permutation_test_score (...)	Evaluate the significance of a cross-validated score with permutations
model_selection.validation_curve (estimator, ...)	Validation curve.

Task 1: LOOCV and K-fold cross validation

- ISLR/Chapter 5.ipynb 을 실행하고 결과를 확인해 보세요.

Task 2: Tree based methods

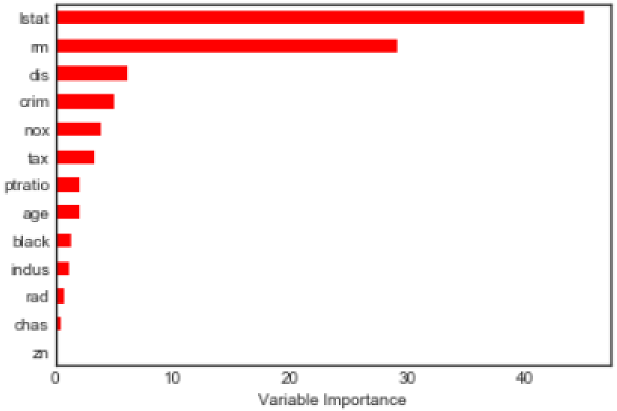
- ISLR/Chapter 8.ipynb 을 실행하고 결과를 확인해 보세요.

Task 3: combining Task 1 and 2

- 1. Chapter 8.ipynb에 있는 Boston dataset을 이용하여 Chapter 5.ipynb의 Figure 5.2와 Figure 5.4 (오른쪽 10-fold CV만) 를 재현하세요. 알고리즘은 RandomForest를 사용하며 10-fold CV, hyperparameter는 feature의 개수(1~10개)로 합니다.

- 2. 1번에 해당하는 ISLR의 text를 읽고 1번의 결과를 설명하시오. 책의 관련 내용(178p, 182p와 그 주변 page들) 을 읽고 이해하여 적으시면 됩니다.

- 3. Boston dataset으로 RandomForest 모델을 만든 후 다음과 같이 feature importance 그림을 그려보세요. Feature의 개수, tree의 개수, data split은 자유입니다.



- 4. (advanced) 1번을 GradientBoostingRegressor functio을 이용해 재현해보세요. Hyperparameter는 n_estimators로 합니다. (range는 자유)

유의사항

- Task는 구현 자체보다는 결과에 대한 분석과 이해를 중심으로 학습하시기 바랍니다.
- 제공된 2018_fall_session_2_task.ipynb 파일에 코드를 작성하고 설명을 단 뒤 ipynb 파일을 제출해주세요.
- 제출시간은 진행상황을 봐가며 정하도록 하겠습니다.
- 다음의 이메일로 choid@snu.ac.kr (최대영 조교) 로 제출하시고 보내실 때 반드시 조별 대표자 성명을 기록해주시기 바랍니다.