

사이킷런

scikit learn

오픈라이브러리

■ 기계 학습을 위한 라이브러리 #1: Scikit-Learn

- 다양한 머신러닝 알고리즘을 구현한 파이썬 라이브러리
- 심플하고 일관성 있는 API, 유용한 온라인 문서, 풍부한 예제
- 머신러닝을 위한 쉽고 효율적인 개발 라이브러리 제공
- 다양한 머신러닝 관련 알고리즘과 개발을 위한 프레임워크와 API제공
- 많은 사람들이 사용하며 다양한 환경에서 검증된 라이브러리

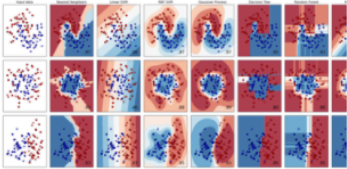
The screenshot shows the Scikit-Learn website at <https://scikit-learn.org/stable/>. The page features the Scikit-Learn logo, navigation links (Install, User Guide, API, Examples, More), and a search bar. The main content area highlights the library's features: simple and efficient tools for predictive data analysis, accessibility to everybody, built on NumPy, SciPy, and matplotlib, and being open source with a BSD license. Below this, three main categories are showcased with descriptive text, applications, algorithms, and representative visualizations.

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

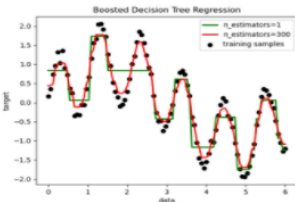


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...




Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...



<https://scikit-learn.org/stable/>

오픈라이브러리

- 기계 학습을 위한 라이브러리 #1: Scikit-Learn

Classification
Regression
Clustering
Semi-Supervised Learning
Feature Selection
Feature Extraction
Manifold Learning
Dimensionality Reduction
Kernel Approximation
Hyperparameter Optimization
Evaluation Metrics
Out-of-core learning
.....



오픈라이브러리

■ 기계 학습을 위한 라이브러리 #1: Scikit-Learn

모듈	설명
<code>sklearn.datasets</code>	내장된 예제 데이터 세트
<code>sklearn.preprocessing</code>	다양한 데이터 전처리 기능 제공 (변환, 정규화, 스케일링 등)
<code>sklearn.feature_selection</code>	특징(feature)을 선택할 수 있는 기능 제공
<code>sklearn.feature_extraction</code>	특징(feature) 추출에 사용
<code>sklearn.decomposition</code>	차원 축소 관련 알고리즘 지원 (PCA, NMF, Truncated SVD 등)
<code>sklearn.model_selection</code>	교차 검증을 위해 데이터를 학습/테스트용으로 분리, 최적 파라미터를 추출하는 API 제공 (GridSearch 등)
<code>sklearn.metrics</code>	분류, 회귀, 클러스터링, Pairwise에 대한 다양한 성능 측정 방법 제공 (Accuracy, Precision, Recall, ROC-AUC, RMSE 등)
<code>sklearn.pipeline</code>	특징 처리 등의 변환과 ML 알고리즘 학습, 예측 등을 묶어서 실행할 수 있는 유틸리티 제공
<code>sklearn.linear_model</code>	선형 회귀, 릿지(Ridge), 라쏘(Lasso), 로지스틱 회귀 등 회귀 관련 알고리즘과 SGD(Stochastic Gradient Descent) 알고리즘 제공
<code>sklearn.svm</code>	서포트 벡터 머신 알고리즘 제공
<code>sklearn.neighbors</code>	최근접 이웃 알고리즘 제공 (k-NN 등)
<code>sklearn.naive_bayes</code>	나이브 베이즈 알고리즘 제공 (가우시안 NB, 다항 분포 NB 등)
<code>sklearn.tree</code>	의사 결정 트리 알고리즘 제공
<code>sklearn.ensemble</code>	앙상블 알고리즘 제공 (Random Forest, AdaBoost, GradientBoost 등)
<code>sklearn.cluster</code>	비지도 클러스터링 알고리즘 제공 (k-Means, 계층형 클러스터링, DBSCAN 등)

오픈라이브러리

■ 기계 학습을 위한 라이브러리 #1: Scikit-Learn

The screenshot shows the Scikit-Learn website. The top navigation bar includes links for 'Install', 'User Guide', 'API', 'Examples', and 'More'. A search bar is on the right. The left sidebar has buttons for 'Prev', 'Up', and 'Next', and a section for 'scikit-learn 0.24.1' with a link to 'Other versions'. Below this is a yellow box asking to 'cite us' and a 'Getting Started' section with links to 'Fitting and predicting: estimator basics', 'Transformers and pre-processors', 'Pipelines: chaining pre-processors and estimators', 'Model evaluation', 'Automatic parameter searches', and 'Next steps'. The main content area has a 'Getting Started' header and a dropdown menu with options like 'Getting Started', 'Tutorial', 'What's new', 'Glossary', 'Development', 'FAQ', 'Support', 'Related packages', 'Roadmap', 'About us', 'GitHub', and 'Other Versions and Download'. The main text describes the purpose of the guide and provides a code example for using the RandomForestClassifier.

```
>>> from sklearn.ensemble import RandomForestClassifier
>>> clf = RandomForestClassifier(random_state=0)
>>> X = [[ 1,  2,  3], # 2 samples, 3 features
...      [11, 12, 13]]
>>> y = [0, 1] # classes of each sample
>>> clf.fit(X, y)
RandomForestClassifier(random_state=0)
```

The `fit` method generally accepts 2 inputs:

- The samples matrix (or design matrix) `X`. The size of `X` is typically `(n_samples, n_features)`, which means that samples are represented as rows and features are represented as columns.
- The target values `y` which are real numbers for regression tasks, or integers for classification (or any other discrete set of values). For unsupervised learning tasks, `y` does not need to be specified. `y` is usually 1d array where the `i` th entry corresponds to the target of the `i` th sample (row) of `X`.

오픈라이브러리

- Estimator API
 - 일관성
 - 모든 객체는 일관된 문서를 갖춘 제한된 메서드 집합에서 비롯된 공동 인터페이스 공유
 - 제한된 객체 계층 구조
 - 알고리즘만 파이썬 클래스에 의해 표현
 - 데이터 세트는 표준 포맷(Numpy 배열, Pandas DataFrame, Scipy 희소 행렬)으로 표현
 - 매개변수명은 표준 파이썬 문자열 사용
 - 합리적인 기본 값
 - 모델이 사용자 지정 파라미터를 필요로 할 때 라이브러리가 적절한 기본 값을 정의

오픈라이브러리

- API 사용 방법

- Scikit-Learn 에서 적절한 estimator 클래스를 임포트해서 모델의 클래스 선택
- 클래스를 원하는 값으로 인스턴스화해서 모델의 하이퍼 파라미터 선택
- 데이터를 특징 배열과 대상 벡터로 배치
- 모델 인스턴스의 fit() 메서드를 호출해 모델을 데이터에 적합
- 모델을 새 데이터에 대해서 적용
 - 지도 학습: 대체로 predict() 메서드를 사용해 알려지지 않은 데이터에 대한 레이블 예측
 - 비지도 학습: 대체로 transform()이나 predict() 메서드를 사용해 데이터의 속성을 변환하거나 추론