

scikit-learn-1

김 종 우



한양대학교



목차

- 개요
- 첫 번째 머신러닝 만들어보기
- 사이킷런의 기반 프레임워크
- Model Selection 모듈
- 데이터 전처리

개요

- 파이썬 머신러닝 라이브러리 중 가장 많이 사용되는 라이브러리

The screenshot displays the scikit-learn website. At the top, there's a navigation bar with links for 'Install', 'User Guide', 'API', 'Examples', and 'More'. Below this, the 'scikit-learn' logo is prominently displayed, followed by the tagline 'Machine Learning in Python'. To the right of the logo, a list of key features is provided: 'Simple and efficient tools for predictive data analysis', 'Accessible to everybody, and reusable in various contexts', 'Built on NumPy, SciPy, and matplotlib', and 'Open source, commercially usable - BSD license'. Below the main header, the website is organized into a grid of six categories, each with a title, description, applications, algorithms, and an 'Examples' link. The categories are: Classification, Regression, Clustering, Dimensionality reduction, Model selection, and Preprocessing. Each category includes a small image or plot illustrating its concept.

scikit-learn
Machine Learning in Python

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

[Getting Started](#) [Release Highlights for 0.23](#) [GitHub](#)

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

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

[Examples](#)

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

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

[Examples](#)

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

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

[Examples](#)

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...

[Examples](#)

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...

[Examples](#)

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.

Algorithms: preprocessing, feature extraction, and more...

[Examples](#)

첫 번째 머신러닝 만들어보기

- iris 데이터 집합



Iris setosa



Iris versicolor



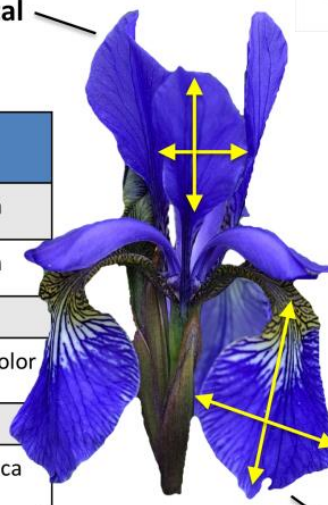
Iris virginica

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)



The diagram shows a detailed view of an Iris flower. A label 'Petal' points to the upper petals, and a label 'Sepal' points to the lower sepals. Yellow arrows indicate the measurements for petal length and width, and sepal length and width.

첫 번째 머신러닝 만들어보기

- 필요 모듈 import

```
from sklearn.datasets import load_iris  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.model_selection import train_test_split
```

첫 번째 머신러닝 만들어보기

- 데이터 가져오기

```
import pandas as pd
```

```
iris = load_iris()
```

```
iris_data = iris.data
```

```
iris_label = iris.target
```

```
print('iris target값:', iris_label)
```

```
print('iris target명:', iris.target_names)
```

```
iris_df = pd.DataFrame(data=iris_data, columns=iris.feature_names)
```

```
iris_df['label'] = iris.target
```

```
iris_df.head(3)
```

	sepal length (cm)	sepal width (cm)	...	petal width (cm)	label
0	5.1	3.5	...	0.2	0
1	4.9	3.0	...	0.2	0
2	4.7	3.2	...	0.2	0

Bunch

data

target

feature_names

target_names

DESCR

첫 번째 머신러닝 만들어보기

- 훈련용, 테스트용 데이터 분할
 - sklearn.model_selection 모듈

[illegible]

첫 번째 머신러닝 만들어보기

- 모형 객체 생성과 학습, 예측

```
dt_clf = DecisionTreeClassifier(random_state=11)
```

```
dt_clf.fit(X_train, y_train)
```

```
pred = dt_clf.predict(X_test)
```


첫 번째 머신러닝 만들어보기

- 성능 평가

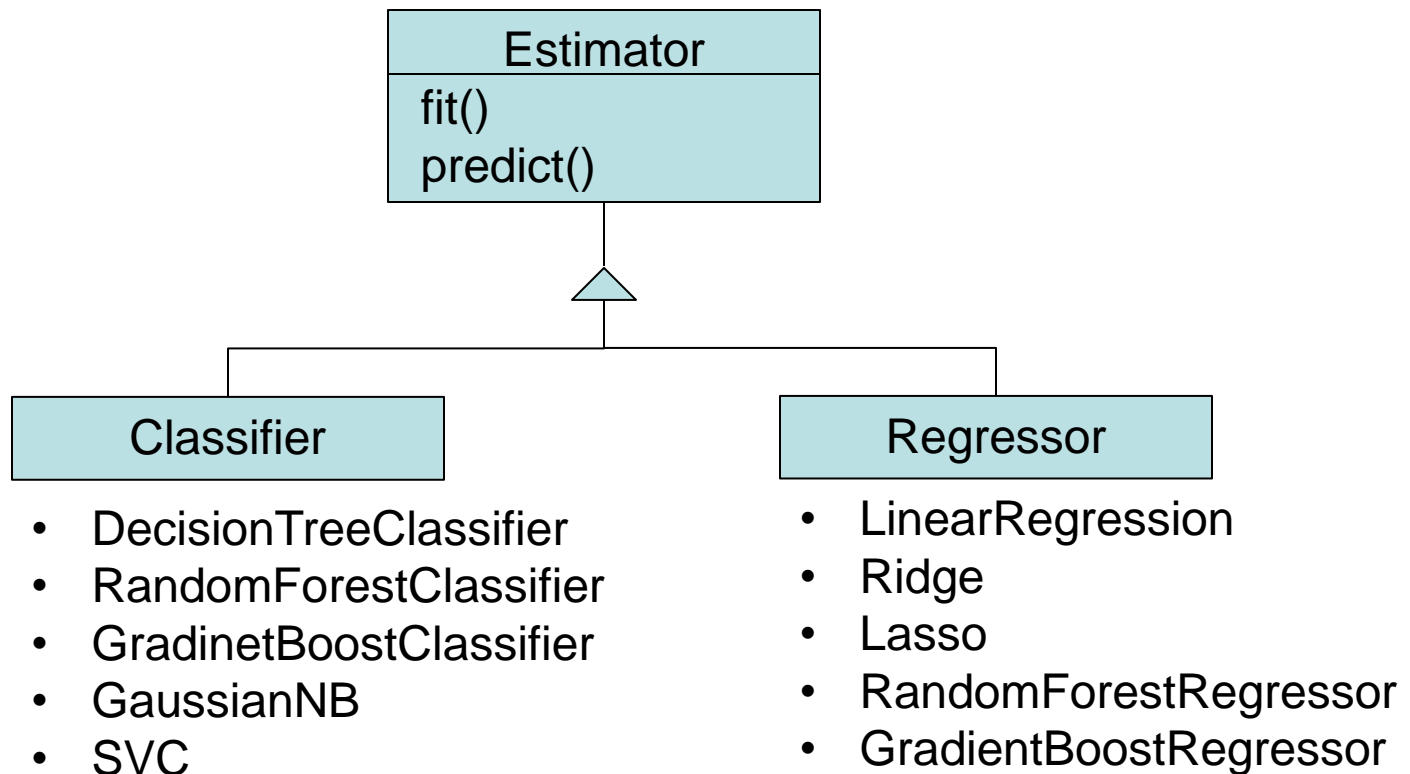
```
from sklearn.metrics import accuracy_score  
print('예측 정확도: {0:.4f}'.format(accuracy_score(y_test, pred)))
```

예측 정확도: 0.9333

사이킷런의 기반 프레임워크

- Estimator

- 지도 학습의 모든 알고리즘의 부모 클래스



사이킷런의 기반 프레임워크

- 분류나 회귀 연습용 예제 데이터
 - `datasets.load_boston()`
 - `datasets.load_breast_cancer()`
 - `datasets.load_diabetes()`
 - `datasets.load_digits()`
 - `datasets.load_iris()`

사이킷런의 기반 프레임워크

- fetch 계열 명령
 - 패키지에 처음부터 저장되어 있지 않고 처음 호출 시 인터넷에서 다운로드. 최초 사용 시 인터넷 연결 필요
 - fetch_covtype(): 회귀분석용 토지 조사
 - fetch_20newsgroup(): 뉴스 그룹 텍스트 데
 - fetch_olivetti_faces(): 얼굴 이미지
 - fetch_lfw_people(): 얼굴 이미지
 - fetch_lfw_pairs(): 얼굴 이미지
 - fetch_rvc1(): 로이터 뉴스 말뭉치
 - fetch_mldata(): ML 웹사이트에서 다운로드

사이킷런의 기반 프레임워크

- 내장 데이터 집합
 - Bunch 객체
 - 딕셔너리 형태
 - Key = data, target, feature_names, target_name, DESCR

Bunch
data
target
feature_names
target_names
DESCR

사이킷런의 기반 프레임워크

- Bunch 객체

```
from sklearn.datasets import load_iris
```

```
iris_data = load_iris()  
print(type(iris_data))
```

<class 'sklearn.utils.Bunch'>

```
keys = iris_data.keys()  
print('붓꽃 데이터 세트의 키들:', keys)
```

붓꽃 데이터 세트의 키들: dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename'])

사이킷런의 기반 프레임워크

- Bunch 객체

```
print('\n feature_names 의 type:',type(iris_data.feature_names))  
print(' feature_names 의 shape:',len(iris_data.feature_names))  
print(iris_data.feature_names)
```

feature_names 의 type: <class 'list'>

feature_names 의 shape: 4

['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']

사이킷런의 기반 프레임워크

- Bunch 객체

```
print('\n target_names 의 type:',type(iris_data.target_names))  
print(' feature_names 의 shape:',len(iris_data.target_names))  
print(iris_data.target_names)
```

```
target_names 의 type: <class 'numpy.ndarray'>  
feature_names 의 shape: 3  
['setosa' 'versicolor' 'virginica']
```


사이킷런의 기반 프레임워크

- Bunch 객체

```
print('\n data 의 type:',type(iris_data.data))  
print(' data 의 shape:',iris_data.data.shape)  
print(iris_data['data'])
```

```
data 의 type: <class 'numpy.ndarray'>  
data 의 shape: (150, 4)  
[[5.1 3.5 1.4 0.2]  
 [4.9 3.  1.4 0.2]  
 .....]
```

[illegible]

Model Selection 모듈

- train_test_split()
 - 훈련/테스트 데이터 세트 분리

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

잘 못 된 예 !!

```
iris = load_iris()
dt_clf = DecisionTreeClassifier()
train_data = iris.data
train_label = iris.target
dt_clf.fit(train_data, train_label)
```

```
# 학습 데이터 셋으로 예측 수행
pred = dt_clf.predict(train_data)
print('예측 정확도:', accuracy_score(train_label, pred))
```

예측 정확도: 1.0

Model Selection 모듈

- `train_test_split()` 주요 인자
 - `test_size`
 - 디폴트는 0.25
 - `shuffle`
 - 디폴트는 True
 - `random_state`
 - 지정하지 않으면 수행할 때마다 다른 학습/테스트 데이터를 생성

Model Selection 모듈

- train_test_split()의 반환값은 튜플 형태

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
```

예측 정확도: 0.9556

```
dt_clf = DecisionTreeClassifier( )
iris_data = load_iris()
```

```
X_train, X_test, y_train, y_test = train_test_split(iris_data.data, iris_data.target,
                                                    test_size=0.3, random_state=121)
```

```
dt_clf.fit(X_train, y_train)
pred = dt_clf.predict(X_test)
print('예측 정확도: {0:.4f}'.format(accuracy_score(y_test, pred)))
```

정리

- 첫번째 머신러닝 만들기
- 사이킷런의 기반 프레임워크
- Model Selection 모듈