

기계학습의 구분

학습 방법의 구분

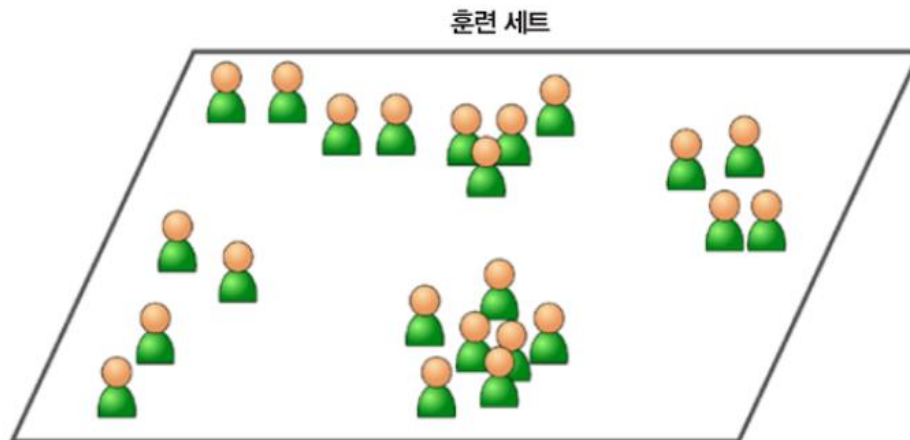
- 지도 학습 (Supervised Learning)
 - 데이터에 정답이 존재하여, 이 정답을 기반으로 학습한다.
 - 이 정답을 **레이블**(label) 또는 **클래스**(class)라고 한다.
 - 지도 학습의 종류
 - 회귀 분석
 - 분류 (의사결정 나무, 랜덤 포레스트, SVM 등)
 - 대부분의 기존 신경망



※ 그림 출처: 핸드온 머신러닝, p39, 한빛미디어

학습 방법의 구분

- 비지도 학습 (Unsupervised Learning)
 - 데이터에 정답이 없어서, 아무런 도움 없이 학습한다.
 - 비지도 학습의 종류
 - 군집화
 - PCA
 - 패턴 탐사 (연관규칙, 빈발항목집합 등)



※ 그림 출처: 핸드온 머신러닝, p41, 한빛미디어

사이킷런 (Scikit-learn)

사이킷런 (Scikit-learn)


- 사이킷런

- 대표적인 파이썬 머신러닝 라이브러리로서, 회귀 분석을 비롯하여 다양한 분석 알고리즘과 기능들을 제공한다.
- 웹사이트 <https://scikit-learn.org> 에서 관련 정보 및 문서, 예제 등을 확인할 수 있다.



사이킷런 (Scikit-learn)

- 사이킷런

- 아나콘다를 설치하면 자동적으로 사이킷런까지 설치되므로 별도의 추가 설치 또는 설정 없이 사용할 수 있다.
- 만약 별도로 재설치하고자 할 경우에는 Anaconda Prompt  에서 다음 명령어 중 하나를 입력하여 설치한다.
 - `pip install scikit-learn`
 - `conda install scikit-learn`
- 사이킷런을 불러올 때의 모듈명은 **sklearn**이다.

```
1  import sklearn
2
3  help(sklearn)
```

Help on package sklearn:

사이킷런 (Scikit-learn)

- 주요 모듈들의 개요 (1)

구분	모듈명	설명
예제 데이터	<code>sklearn.datasets</code>	사이킷런에 내장되어 있는 예제 데이터들
전처리 및 특성 처리	<code>sklearn.preprocessing</code>	데이터의 정규화, 스케일링 등 전처리 및 가공 기능
	<code>sklearn.feature_selection</code>	분석 수행과 관련된 특성 값들에 대한 처리 기능
데이터 분리 및 검증	<code>sklearn.model_selection</code>	학습용/검증용 데이터 분리, 매개변수 조정 기능
성능 평가	<code>sklearn.metrics</code>	분석 결과에 대한 성능 측정 및 평가 기능

사이킷런 (Scikit-learn)

- 주요 모듈들의 개요 (2)

구분	모듈명	설명
분석 기법 (알고리즘)	<code>sklearn.linear_model</code>	회귀 분석 (선형 회귀, 릿지, 라쏘, 로지스틱 회귀 등)
	<code>sklearn.svm</code>	서포트 벡터 머신
	<code>sklearn.tree</code>	의사 결정 나무
	<code>sklearn.ensemble</code>	앙상블 기법 (랜덤 포레스트, 에이다 부스트 등)
	<code>sklearn.cluster</code>	군집화 (K-means, DBSCAN 등)

사이킷런 (Scikit-learn)

- 내장 예제 데이터
 - 사이킷런 내에는 분석을 수행할 때 사용할 수 있는 기본적인 예제 데이터 집합들이 들어 있다.
 - 이 데이터들은 크게 세 가지로 분류할 수 있다.

종류	의미	함수 접두어
Toy dataset	크기가 작고 간단한 샘플 데이터를 불러온다.	load_
Real dataset	레코드가 더 많은 실제 데이터를 다운로드하여 사용한다.	fetch_
Generated dataset	사용자가 원하는 특성에 맞도록 데이터를 생성한다.	make_

사이킷런 (Scikit-learn)

- 내장 예제 데이터

- 내장 데이터는 대부분 사전 또는 그와 유사한 형태로 저장되어 있으며, 각 키들의 의미는 다음과 같다.

키	의미	자료형
data	데이터 집합의 특성(또는 독립변수) 값들이다.	ndarray
target	데이터 집합의 레이블(또는 종속변수) 값들이다.	
feature_names	특성(또는 독립변수)들의 이름이다.	ndarray 또는 list
target_names	레이블(또는 종속변수)들의 이름이다.	
DESCR	데이터에 대한 전체적인 설명이다.	str

사이킷런 (Scikit-learn)

- 내장 예제 데이터
 - 붓꽃 데이터 IRIS 살펴보기

```
1 import sklearn.datasets as d
2
3 iris = d.load_iris()
```

```
1 print("속성 :", iris.feature_names)
2 print("레이블 :", iris.target_names)
```

```
속성 : ['sepal length (cm)', 'sepal width (cm)',
        'petal length (cm)', 'petal width (cm)']
레이블 : ['setosa' 'versicolor' 'virginica']
```

```
1 print("레코드 수:", len(iris.data))
```

```
레코드 수: 150
```

사이킷런 (Scikit-learn)

- 학습/검증 데이터 분리
 - `model_selection` 모듈에 있는 `train_test_split` 함수를 이용하여 원본 데이터를 학습용과 검증용으로 분리한다.
 - 첫 번째 매개변수는 원본 데이터의 특성 집합이다.
 - 두 번째 매개변수는 원본 데이터의 레이블 집합이다.
 - 다음과 같은 옵션들을 추가할 수 있다.
 - `test_size` : 검증용 데이터의 크기를 결정한다. 기본값은 0.25(즉, 25%)이다.
 - `shuffle` : 데이터를 섞어서 분리할 것인지의 여부를 결정한다. 기본값은 True이다.
 - `random_state` : 분리할 때마다 동일한 집합으로 생성할 것인지 결정하는 정수 값이다.

사이킷런 (Scikit-learn)

- 학습/검증 데이터 분리
 - 붓꽃 데이터 IRIS 분리하기

```
1 import sklearn.model_selection as ms
2
3 X_train, X_test, y_train, y_test = #
4 ms.train_test_split(iris.data, iris.target, #
5                      test_size=0.3, random_state=42)
```

```
1 print("학습용 데이터 수: ", len(X_train))
2 print("검증용 데이터 수: ", len(X_test))
```

학습용 데이터 수: 105

검증용 데이터 수: 45

※ 분리된 결과는 학습용 데이터의 특성 집합, 검증용 데이터의 특성 집합, 학습용 데이터의 레이블 집합, 검증용 데이터의 레이블 집합을 순서대로 항목들로 가지는 1개의 튜플이다.

사이킷런 (Scikit-learn)

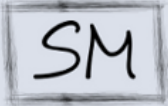
- 사이킷런의 분석 수행 절차

- ① 분석을 수행하기 위한 추정자(Estimator) 객체를 생성한다.
- ② 생성된 객체에 대하여 **fit** 메소드(함수)를 호출하여 학습을 수행한다.
- ③ 생성된 객체 또는 학습이 수행된 분석 결과에 대하여 **predict** 메소드(함수)를 호출하여 예측 결과를 도출한다.
- ④ 생성된 객체 또는 학습이 수행된 분석 결과에 대하여 적합한 성능 평가 지표를 도출한다.

스탯츠모델 (Statsmodels)

스탯츠모델 (Statsmodels)

- 스탯츠모델
 - 다양한 통계 검정 및 추정, 회귀 분석, 시계열 분석 기능을 제공하는 통계 분석 라이브러리이다.
 - 웹사이트 <https://www.statsmodels.org> 에서 관련 정보 및 문서, 예제 등을 확인할 수 있다.



StatsModels
Statistics in Python

[Install](#) | [Support](#) | [Bugs](#) | [Develop](#) | [Examples](#) | [FAQ](#) |

Download

This documentation is for the **v0.10.0** release. You can install it with pip:

```
pip install --upgrade  
--no-deps statsmodels
```

or conda:


```
conda install statsmodels
```

Welcome to Statsmodels's Documentation

statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct. The package is released under the open source Modified BSD (3-clause) license. The online documentation is hosted at [statsmodels.org](https://www.statsmodels.org).

스탯츠모델 (Statsmodels)

- 스탯츠모델

- 아나콘다를 설치하면 자동적으로 스탯츠모델이 설치되므로 별도의 추가 설치 또는 설정 없이 사용할 수 있다.
- 만약 별도로 재설치하고자 할 경우에는 Anaconda Prompt  에서 다음 명령어 중 하나를 입력하여 설치한다.
 - `pip install statsmodels`
 - `conda install statsmodels`
- 스탯츠모델을 불러올 때의 모듈명은 **statsmodels**이다.

```
1 import statsmodels
2
3 help(statsmodels)
```

Help on package statsmodels:

스탯츠모델 (Statsmodels)

- 스탯츠모델
 - 기존의 R Studio에서 제공하는 것과 동일한 명령어들을 이용하여 통계 분석과 시계열 분석, 검정 통계량 계산 등을 수행할 수 있다.
 - 따라서, 회귀 분석(OLS) 또는 주성분 추출(PCA) 등의 기법은 상황에 따라 사이킷런 외에 스탯츠모델을 이용하여 수행하는 것도 무방하다.
 - 분석 기법들은 서브 모듈 **api**를 이용한다.

```
1 import statsmodels.api as sm
2
3 help(sm)
```

Help on module statsmodels.api in statsmodels: