
파이썬 기반의
빅데이터 처리 및
분석 기술
Crayon

# 빅데이터 회귀 분석

## 2 Scikit Learn API 사용법

- Scikit Learn API 장점
- Scikit Learn 기본 사용법
- 간단한 회귀 분석

## 1. Scikit Learn

### 1) 개요

**“ 머신러닝 ”**

- 데이터로부터 모델을 만드는 학습 과정

## 1. Scikit Learn

# 1) 개요

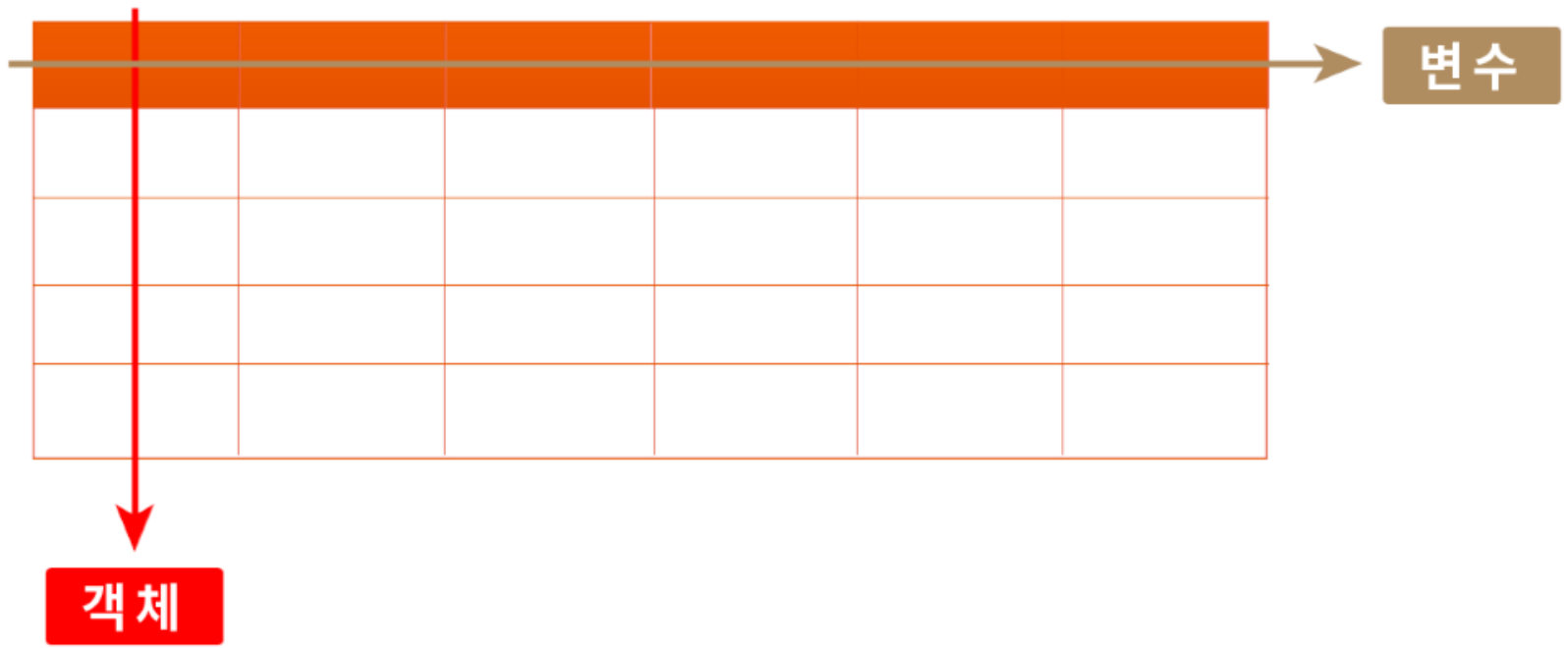
**“ 데이터 표현 방법 ”**

| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

## 1. Scikit Learn

### 1) 개요

"데이터 표현 방법"



## 1. Scikit Learn

# 1) 개요

"데이터 표현 방법"

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head(4)
```

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |

## 1. Scikit Learn

### 1) 개요

"데이터 표현 방법"

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head(4)
```

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |

꽃 한 개체

## 1. Scikit Learn

# 1) 개요

"데이터 표현 방법"

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head(4)
```

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |

꽃 한 개체

## 1. Scikit Learn

# 1) 개요

"데이터 표현 방법"

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head(4)
```

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |

꽃 한 개체

꽃의 품종

## 1. Scikit Learn

### 1) 개요

"데이터 표현 방법"

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head(4)
```

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |

종속 변수

꽃 한 개체

꽃의 품종

## 1. Scikit Learn

# 1) 개요

## " Scikit Learn API 설계 기본 원칙 "

| | |
|---|---|
| 일관성 | 모든 객체는 사용법이 일관된 공통 인터페이스를 공유 |
| 검사 | 모든 파라미터 값은 외부에서 확인 가능한 공개 속성임 |
| 제한된 계층 구조 | 알고리즘만 파이썬 클래스로 표현되고, 데이터는 표준 포맷 사용 |
| 구성 | 머신러닝 작업은 기본적으로 제공되는 기본 알고리즘의 시퀀스로 구성됨 |
| 합리적 기본 | 사용자 지정이 필요한 파라미터는 적절한 기본값으로 설정 |

- 서로 다른 API를 사용해도 공통 인터페이스를 사용
- 넘파이, 판다스와 같은 파이썬 표준 포맷을 사용
- 배우기 쉽고 응용이 편리하도록 개발

## 1. Scikit Learn

### 1) 개요

**" Scikit Learn API 사용 단계 "**

| | |
|---|---|
| 1 | Scikit Learn API에서 적절한 추정기 클래스를 임포트 해서 사용하고자 하는 모델을 선택한다. |
| 2 | 클래스로부터 인스턴스를 생성하고 초모수(Hyper-parameter)를 설정한다. |
| 3 | 데이터를 특징 배열과 대상 배열로 준비한다. |
| 4 | 모델 인스턴스의 fit() 메소드를 호출해서 데이터를 학습한다. |
| 5 | 정확도를 확인하고 새로운 데이터에 모델을 적용한다. |

1. Scikit Learn

# 2) Scikit Learn 기본 사용법

"간단한 회귀 분석"

```
import matplotlib.pyplot as plt
import numpy as np

rng = np.random.RandomState(42)
x = 10 * rng.rand(50)
y = 2 * x - 1 + rng.randn(50)
plt.scatter(x, y);
```

1. Scikit Learn

# 2) Scikit Learn 기본 사용법

"간단한 회귀 분석"

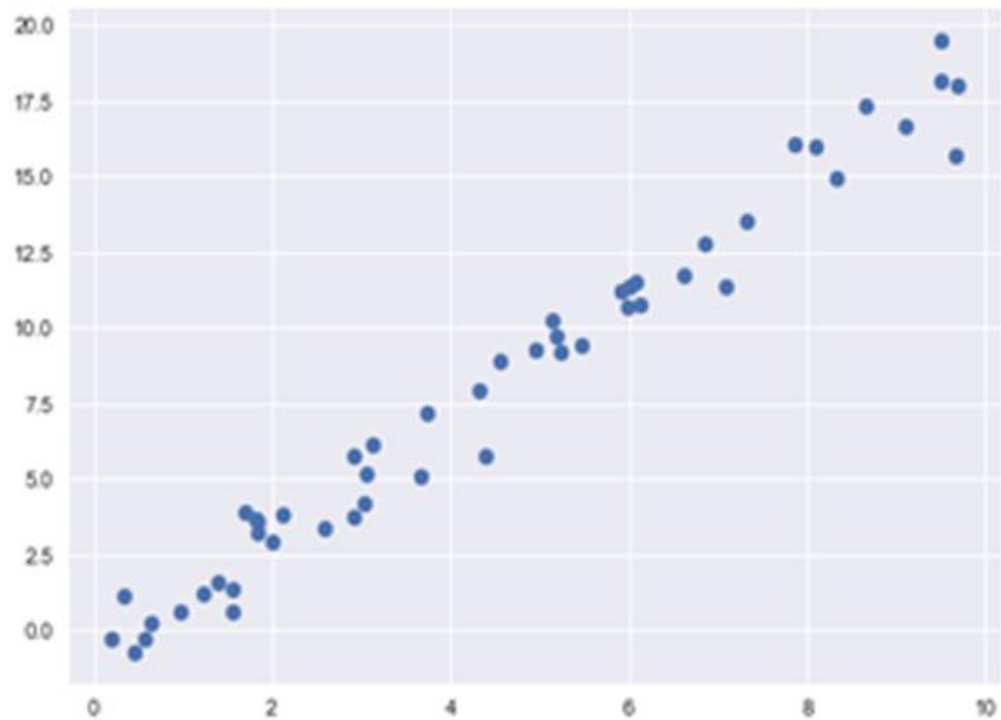
```
import matplotlib.pyplot as plt
import numpy as np

rng = np.random.RandomState(42)
x = 10 * rng.rand(50)
y = 2 * x - 1 + rng.randn(50)
plt.scatter(x, y);
```

1. Scikit Learn

# 2) Scikit Learn 기본 사용법

"간단한 회귀 분석"

```
import matplotlib.pyplot as plt
import numpy as np

rng = np.random.RandomState(42)
x = 10 * rng.rand(50)
y = 2 * x - 1 + rng.randn(50)
plt.scatter(x, y);
```

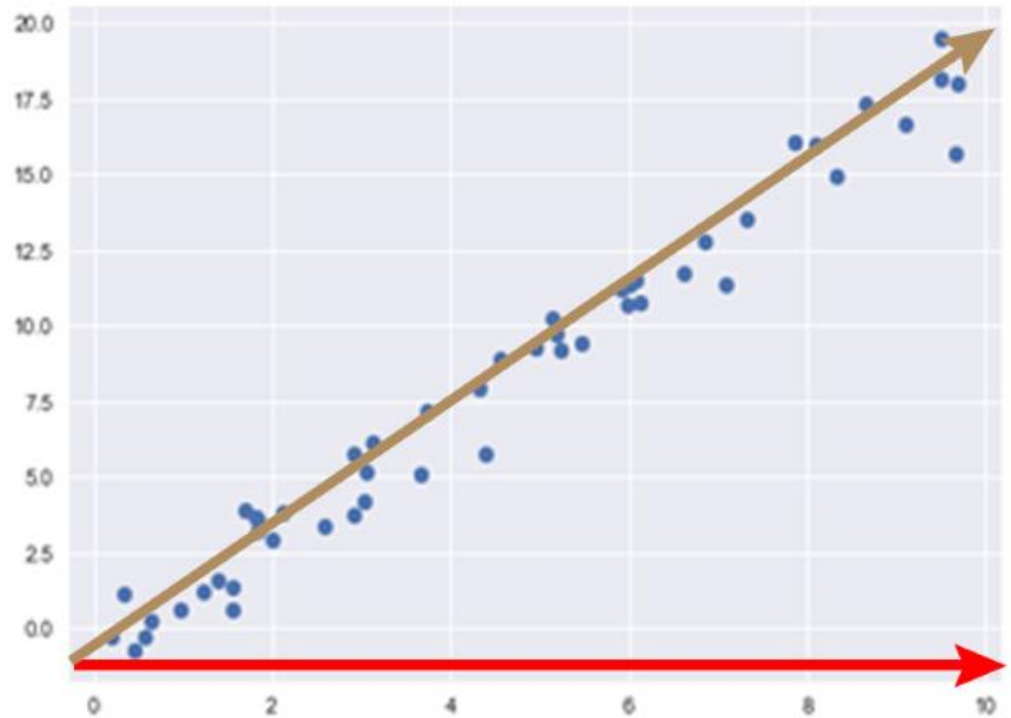


1. Scikit Learn

# 2) Scikit Learn 기본 사용법

"간단한 회귀 분석"

```
import matplotlib.pyplot as plt
import numpy as np

rng = np.random.RandomState(42)
x = 10 * rng.rand(50)
y = 2 * x - 1 + rng.randn(50)
plt.scatter(x, y);
```

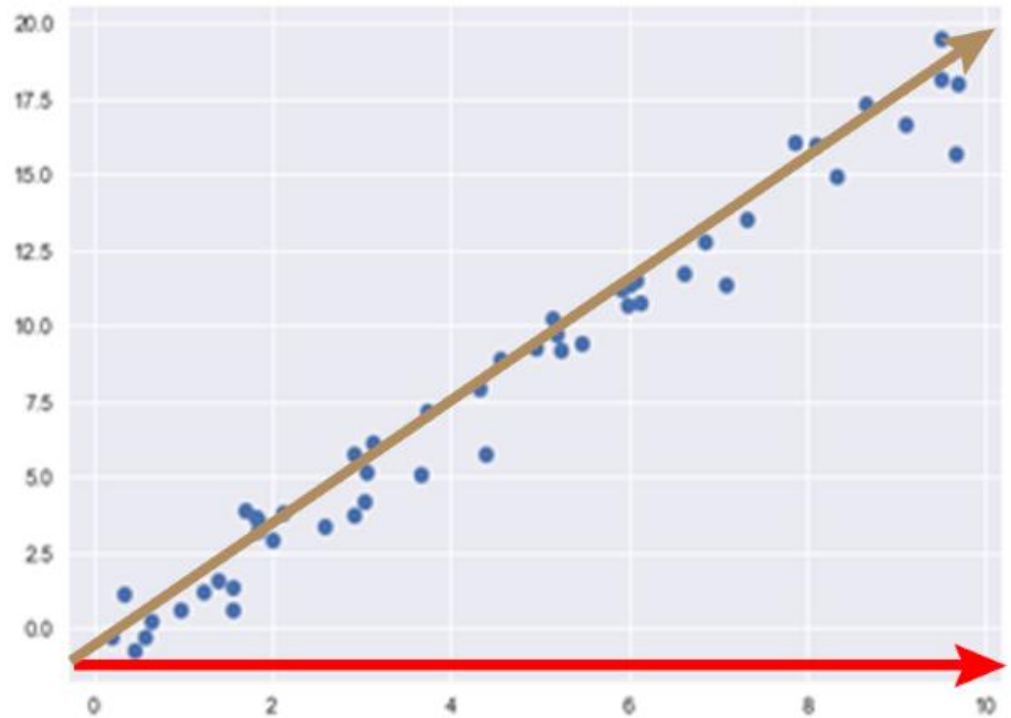


1. Scikit Learn

# 2) Scikit Learn 기본 사용법

"간단한 회귀 분석"

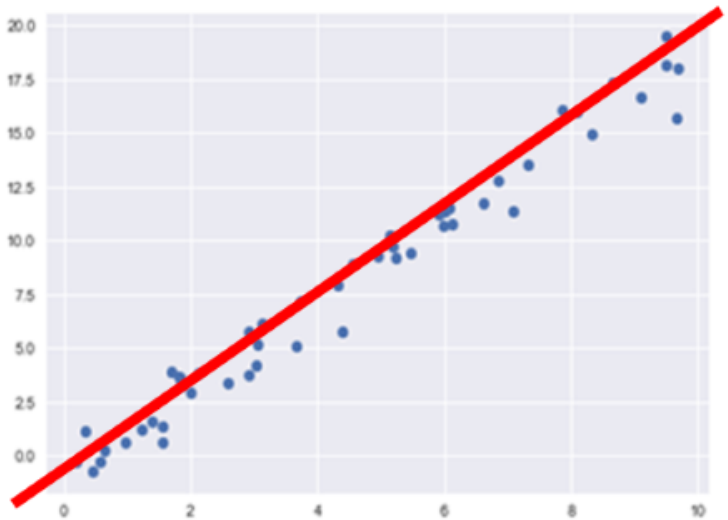
```
import matplotlib.pyplot as plt
import numpy as np

rng = np.random.RandomState(42)
x = 10 * rng.rand(50)
y = 2 * x - 1 + rng.randn(50)
plt.scatter(x, y);
```



1. Scikit Learn

# 2) Scikit Learn 기본 사용법

## " 간단한 회귀 분석 "




$y = ax + b$

선형 회귀 분석의 결과가 예상대로 나오는지 확인 필수

1. Scikit Learn

# 3) 회귀 분석 과정

## " 모델 클래스 선택 "

- from sklearn.linear_model import LinearRegression

1. Scikit Learn

# 3) 회귀 분석 과정

## " 모델 초모수 선택 "

- model = LinearRegression(fit_intercept=True)

1. Scikit Learn

# 3) 회귀 분석 과정

## " 모델 초모수 선택 "

- model = LinearRegression(fit_intercept=True)

  - 절편 사용 여부 결정
  - 절편 사용 OK

- LinearRegression 클래스에 절편 옵션을 사용하는 model 인스턴스를 생성하는 과정

1. Scikit Learn

# 3) 회귀 분석 과정

## "데이터 차원 변경"

- X = x[:, np.newaxis]
  X.shape

- X = x.reshape(50, 1)
  X.shape

1. Scikit Learn

# 3) 회귀 분석 과정

## “ 모델에 데이터 적용 ”

- model.fit(X, y)

1. Scikit Learn

# 3) 회귀 분석 과정

## "모델에 데이터 적용"

- model.fit(X, y)

- 모델에 따라 결정된 여러 가지 내부 계산
- 계산 결과는 모델 인스턴스 속성에 저장

1. Scikit Learn

# 3) 회귀 분석 과정

## "모델 확인"

- print(model.coef_) → [1.9776566]
- print(model.intercept_) → -0.90331072553111146

1. Scikit Learn

# 3) 회귀 분석 과정

## " 모델 정확도 확인 "

- model.score(X, y)

0.97 정도의 $R^2$(결정 계수)값 산출

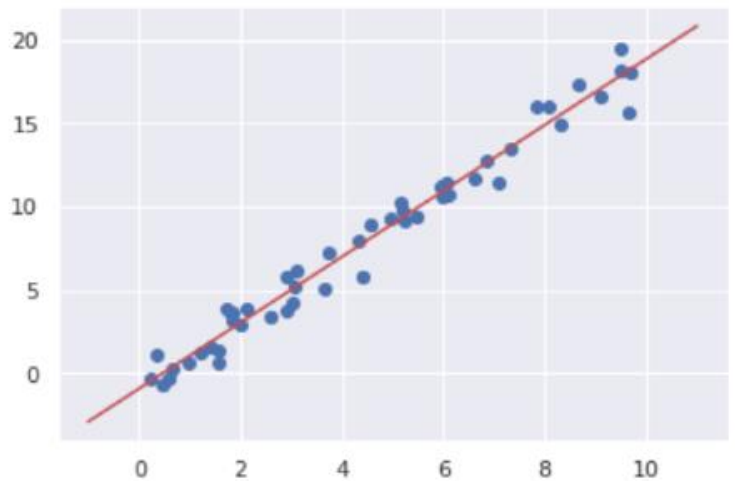
1에 가까운 값을 나타낼 수록 정확도가 높다고 판단

1. Scikit Learn

# 3) 회귀 분석 과정

## " 모델 시각화 "

- xfit = np.linspace(-1, 11)
  plt.scatter(x, y)
  plt.plot(xfit, model.coef_ * xfit + model.intercept_, '-r');



# 이번 시간에는

2 **Scikit Learn API 사용법**

- Scikit Learn API 장점
- Scikit Learn 기본 사용법
- 간단한 회귀 분석

# 이번 시간에는

**실습 참고 자료**

- Colab 노트북 파일
- Matplotlib 공식 사이트
  → https://matplotlib.org/tutorials/index.html

# 다음 시간에는

3 빅데이터 회귀 분석 심화

- 최적의 모델 선택
- Scikit Learn 다항식 회귀 모델
- 학습 곡선