

03장 머신러닝의 기초를 다집니다

— 수치 예측

Draft

- 딥러닝의 기초가 되는 핵심 알고리즘
 1. 선형 회귀
 2. 경사 하강법
 3. 손실 함수
 4. 선형 회귀를 위한 뉴런을 만들기

Draft

- 03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다
- 03-2 경사 하강법으로 학습하는 방법을 알아봅니다
- 03-3 손실 함수와 경사 하강법의 관계를 알아봅니다
- 03-4 선형 회귀를 위한 뉴런을 만듭니다

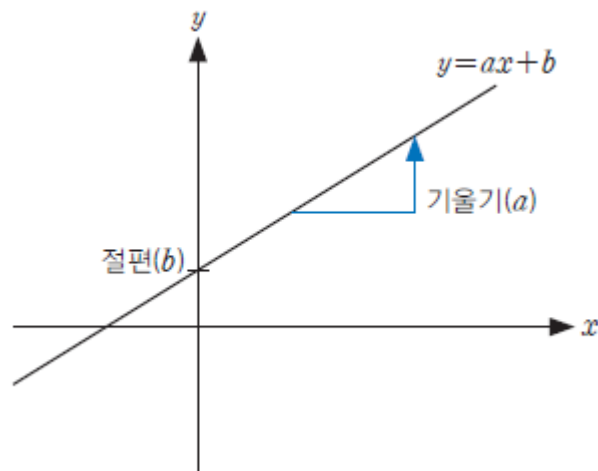
Draft

- 03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다
- 03-2 경사 하강법으로 학습하는 방법을 알아봅니다
- 03-3 손실 함수와 경사 하강법의 관계를 알아봅니다
- 03-4 선형 회귀를 위한 뉴런을 만듭니다

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

1차 함수로 이해하는 선형 회귀

- 선형 회귀는 머신러닝 알고리즘 중 가장 간단하면서도 딥러닝의 기초가 됨
- 선형 회귀는 아주 간단한 1차 함수로 표현할 수 있음
 $y = ax + b$ 기울기 a , 절편 b
- 선형 회귀의 선형이라는 단어의 의미는 다음 수식을 통해 그려지는 직선 그래프를 보면 쉽게 이해 가능



03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

1차 함수로 이해하는 선형 회귀

선형 회귀는 기울기와 절편을 찾아줍니다

- 선형 회귀는 기울기와 절편을 찾아냄
- 학교에서 배운 1차 함수는 기울기와 절편이 주어지면 이를 만족하는 x 와 y 를 찾을 수 있음

$$y = ax + b \text{ 기울기 } a, \text{ 절편 } b$$

1차 함수 문제 기울기가 7이고 절편이 4인 1차 함수 $y=7x+4$ 가 있습니다. x 가 10이면 y 는 얼마인가요?

- ① 74
- ② 72
- ③ 71

- 보통 1차 함수 문제에서는 이런식으로 x 에 따른 y 의 값에 집중
- 선형 회귀에서는 이와 반대로 x, y 가 주어졌을때 기울기와 절편을 찾는 데 집중

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

1차 함수로 이해하는 선형 회귀

선형 회귀는 기울기와 절편을 찾아줍니다

- 잠깐만 생각해 보면 정답을 알 수 있음!!
- 선형 회귀는 위 문제를 어떤 과정을 통해 해결할까??
- 다음을 보면서 조금 더 자세히 알아보자

선형 회귀 문제 x 가 3일 때 y 는 25, x 가 4일 때 y 는 32, x 가 5일 때 y 는 39라면 기울기와 절편의 값으로 적절한 것은 무엇인가요?

- ① 기울기는 6, 절편은 4
- ② 기울기는 7, 절편은 5
- ③ 기울기는 7, 절편은 4

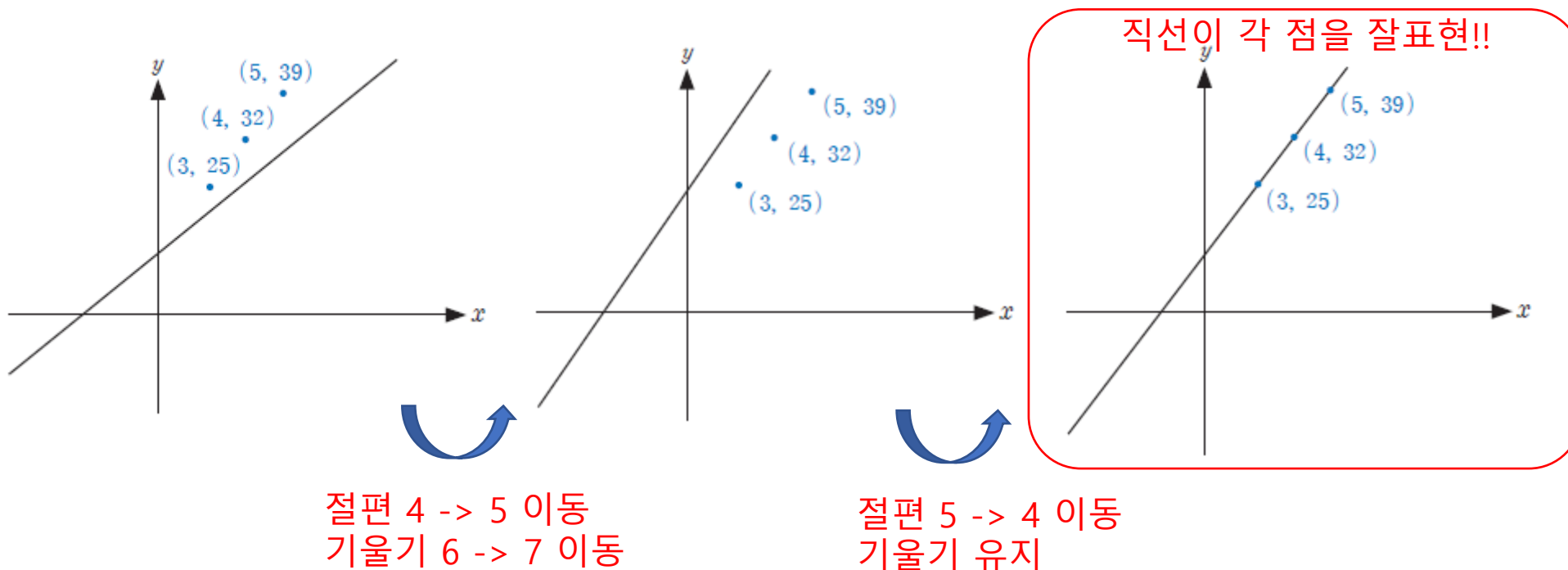
03-1 선형 회귀에 대해 알아보고 데이터 1차 함수로 이해하는 선형 회귀 그래프를 통해 선형 회귀의 문제 해결 과정

선형 회귀 문제 x 가 3일 때 y 는 25, x 가 4일 때 y 는 32, x 가 5일 때 y 는 39라면 기울기와 절편의 값으로 적절한 것은 무엇인가요?

- ① 기울기는 6, 절편은 4
- ② 기울기는 7, 절편은 5
- ③ 기울기는 7, 절편은 4

- 아래 그림은 바로 앞에서 본 선형 회귀 문제를 그래프로 표현한 것
- 점은 x, y 를 표현한 것. 직선은 보기 1의 조건(기울기 6과 절편 4)을 가진 1차 함수를 표현한 것
- 보기 1의 조건을 갖춘 1차 함수는 점을 잘 표현하지 못함

선형 회귀의 문제 해결 과정(기울기와 절편을 조금씩 수정하여 데이터에 맞춤)



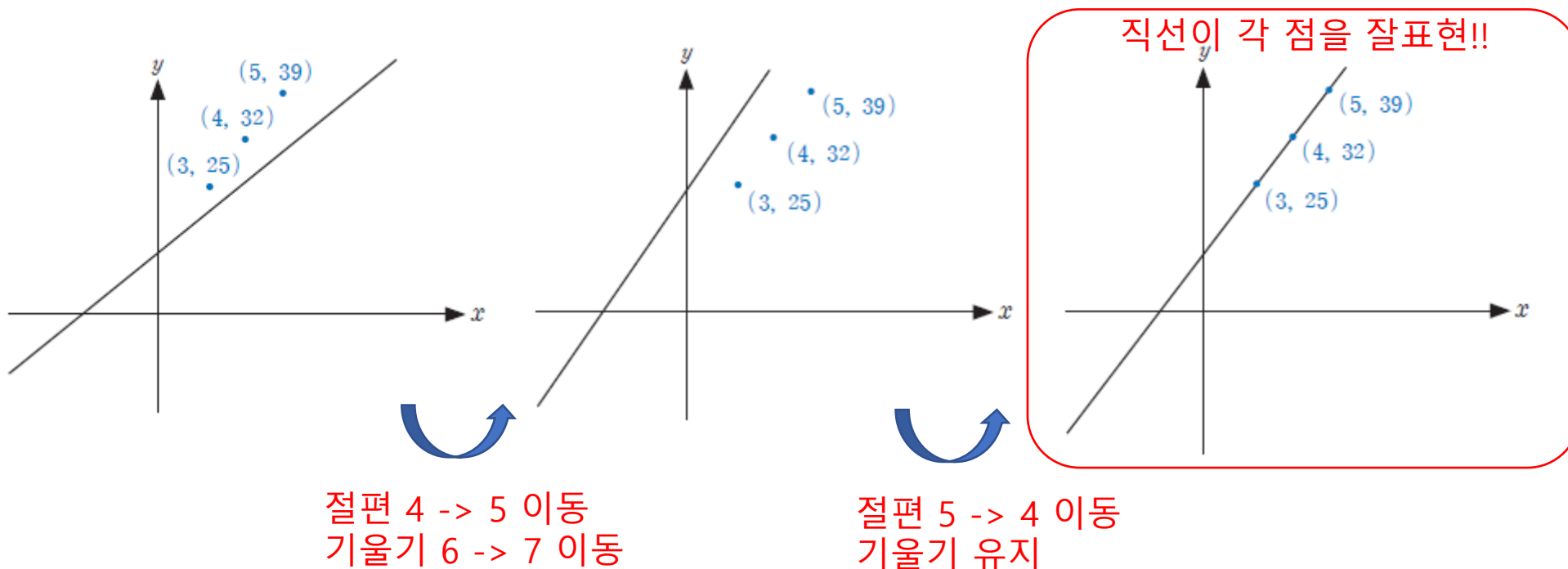
03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

1차 함수로 이해하는 선형 회귀

그래프를 통해 선형 회귀의 문제 해결 과정을 이해

- 지금 만들어진 1차 함수들을 '선형 회귀로 만든 모델'이라고 함
- 마지막에 만들어진 1차 함수가 최적의 선형 회귀 모델임
- 이런 모델을 통해서 새로운 점에 대한 예측을 할 수 있음

선형 회귀의 문제 해결 과정(기울기와 절편을 조금씩 수정하여 데이터에 맞춤)



03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

1차 함수로 이해하는 선형 회귀

그래프를 통해 선형 회귀의 문제 해결 과정을 이해



잠깐! 다음으로 넘어가려면

- ☒ 선형 회귀는 머신러닝 알고리즘 중 하나입니다.
- ☐ 선형 회귀는 2차원 평면에 놓인 점을 표현하는 1차 함수의 기울기와 절편을 찾아줍니다.
- ☐ 선형 회귀로 찾은 이런 1차 함수를 모델이라고 부릅니다.
- ☐ 선형 회귀 모델로 새 값 x 에 대하여 y 를 예측할 수 있습니다.

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

문제 해결을 위해 당뇨병 환자의 데이터 준비하기

사이킷런에서 당뇨병 환자 데이터 가져오기

- 목표 : 당뇨병 환자의 1년 후 병의 진전된 정도를 예측하는 모델을 만드는 것
- 문제를 해결하기 위해 가장 먼저 해야 할 일은 충분한 양의 입력 데이터와 타깃 데이터를 준비하는 것
- 머신러닝, 딥러닝 패키지에는 인공지능 학습을 위한 데이터 세트가 준비되어 있음
- 사이킷런 패키지의 당뇨병 환자 데이터 셋을 사용
- 사이킷런 패키지 다운로드

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

Installing the latest release

Operating System

Windows

macOS

Linux

Packager

pip

conda

Use pip virtualenv

Install the 64bit version of Python 3, for instance from <https://www.python.org>.
Then run:

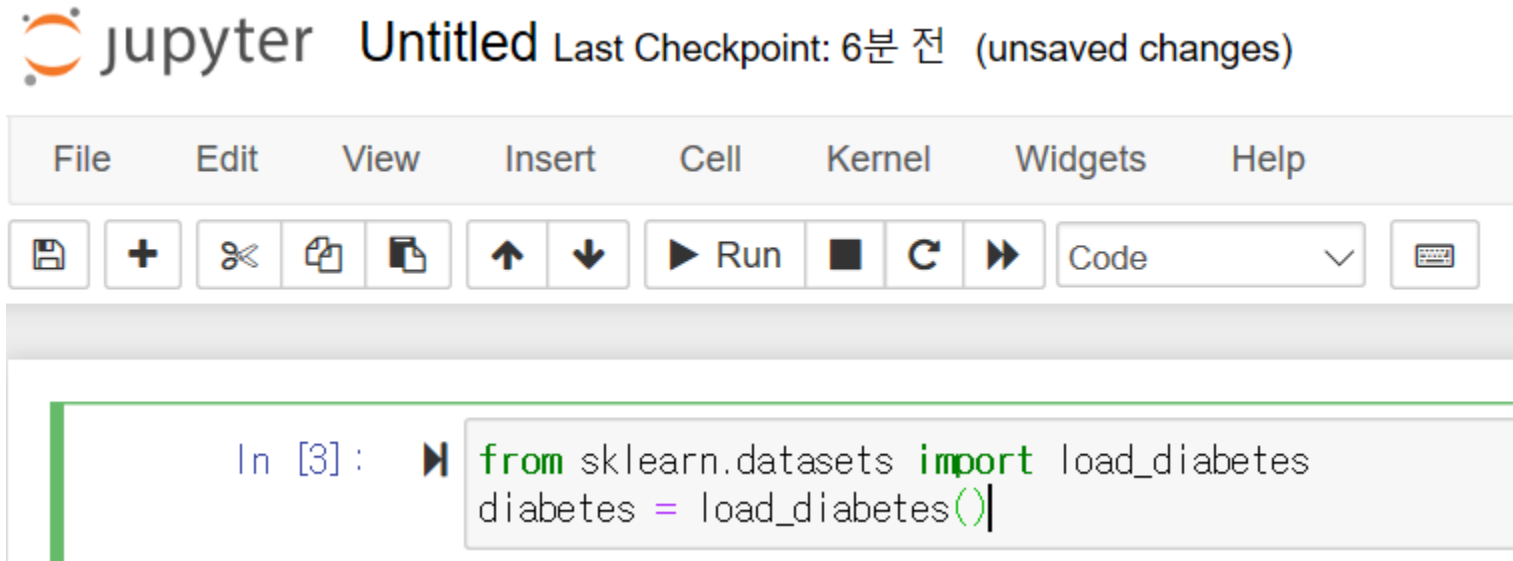
```
$ pip install -U scikit-learn
```

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

문제 해결을 위해 당뇨병 환자의 데이터 준비하기

사이킷런에서 당뇨병 환자 데이터 가져오기

1. 함수로 당뇨병 데이터 준비하기 : `load_diabetes()` 함수를 임포트한 후 매개변수 값을 넣지 않은 채로 함수를 호출하면 `diabetes`에 당뇨병 데이터가 저장됨
2. `Diabetes` 변수에 저장된 값의 자료형은 파이썬 딕셔너리와 유사한 `bunch` 클래스 (`bunch` 클래스는 예제 데이터 세트를 위해 준비된 것일 뿐 특별한 기능이 있는 건 아님. 파이썬 딕셔너리처럼 생각해도 무방!!!)



The image shows a Jupyter Notebook window titled "Untitled" with a subtitle "Last Checkpoint: 6분 전 (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar is a toolbar with icons for saving, adding cells, deleting, copying, pasting, undo, redo, and running code. The code cell contains the following Python code:

```
In [3]: from sklearn.datasets import load_diabetes
diabetes = load_diabetes()
```

번치와 딕셔너리의 차이점

<https://stackoverflow.com/questions/56286221/what-is-the-difference-between-bunch-and-dictionary-type-in-python>

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

문제 해결을 위해 당뇨병 환자의 데이터 준비하기

사이킷런에서 당뇨병 환자 데이터 가져오기

2. 입력과 타겟 데이터의 크기 확인하기
 - Diabetes 속성 중 data 속성과 target 속성에는 입력과 타겟 데이터가 넘파이 배열로 저장되어 있음
 - Shape 속성을 이용하여 넘파이 배열의 크기를 알 수 있음

```
In [4]: ▶ print(diabetes.data.shape diabetes.target.shape)
(442, 10) (442,)
```

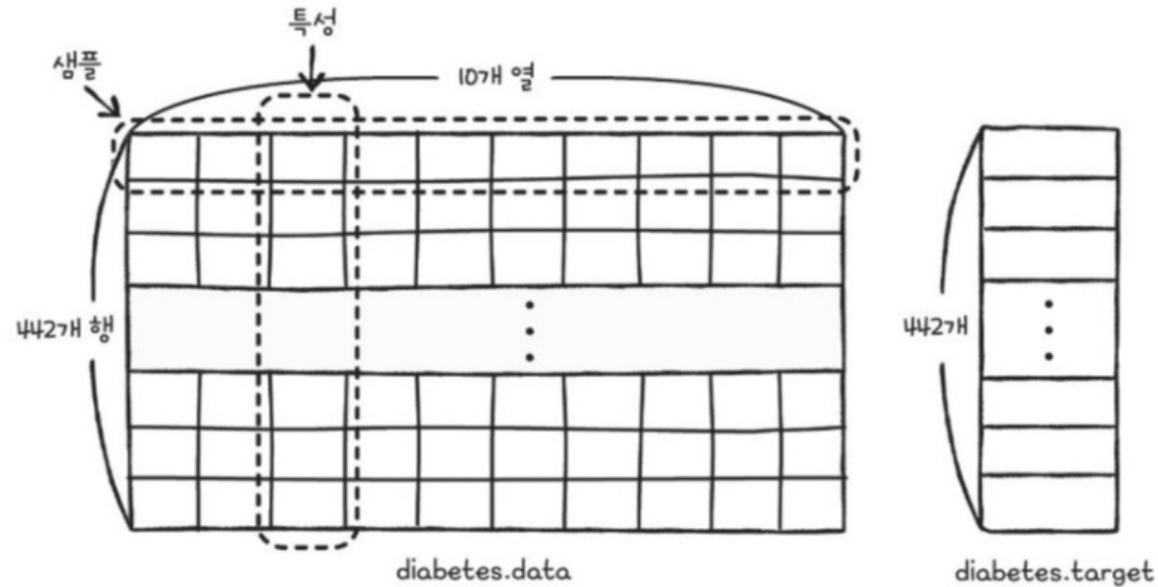
속성 행

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

문제 해결을 위해 당뇨병 환자의 데이터 준비하기

사이킷런에서 당뇨병 환자 데이터 가져오기

data는 442×10 크기의 2차원 배열이고 target은 442개의 요소를 가진 1차원 배열입니다.
다음 그림은 data와 target을 그림으로 나타낸 것입니다.

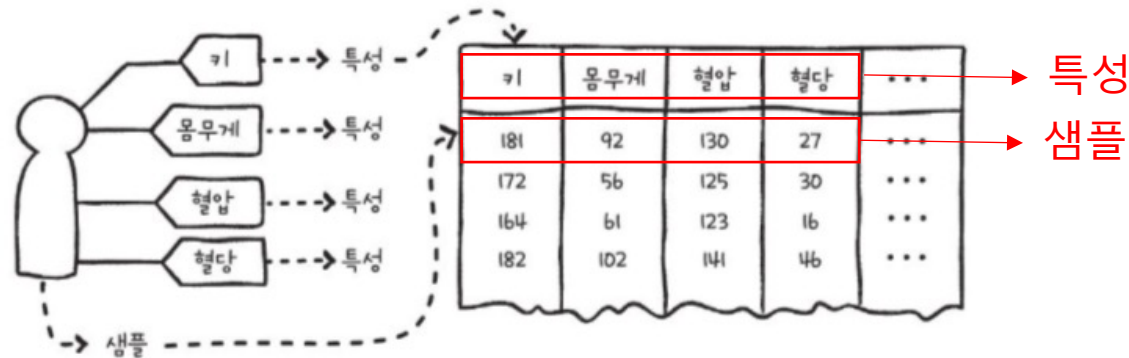


03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

문제 해결을 위해 당뇨병 환자의 데이터 준비하기

사이킷런에서 당뇨병 환자 데이터 가져오기

diabetes.data를 보면 442개의 행과 10개의 열로 구성되어 있음을 알 수 있습니다. 여기서 행은 샘플(sample)이고, 열은 샘플의 특성(feature)입니다. **샘플**이란 당뇨병 환자에 대한 특성으로 이루어진 데이터 1세트를 의미하고, **특성**은 당뇨병 데이터의 여러 특징들을 의미합니다. 쉽게 말해 당뇨병 데이터에는 환자의 혈압, 혈당, 몸무게, 키 등의 특징(특성)이 있는데, 그 특징들의 수치를 모아 1세트로 만들면 1개의 샘플이 나온다고 생각하면 됩니다. 다음은 샘플과 특성의 이해를 돕기 위해 가상의 환자와 당뇨병 데이터를 그림으로 나타낸 것입니다. 실제 데이터와는 차이가 있으니 주의하세요.



03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

문제 해결을 위해 당뇨병 환자의 데이터 준비하기

사이킷런에서 당뇨병 환자 데이터 가져오기

3. 입력데이터 자세히 보기

- diabetes.data에 저장된 입력 데이터 일부만 출력해보기
- 슬라이싱을 사용해 입력 데이터 앞부분의 샘플 3개만 출력
- 안쪽 대괄호에는 특성의 값 10개가 나열되어 있는데 3개의 샘플을 추출했으므로 3x10 크기의 배열이 나타남

```
In [5]: ▶ diabetes.data[0:3]
```

```
Out [5]: array([[ 0.03807591,  0.05068012,  0.06169621,  0.02187235, -0.0442235 ,
                 -0.03482076, -0.04340085, -0.00259226,  0.01990842, -0.01764613],
                [-0.00188202, -0.04464164, -0.05147406, -0.02632783, -0.00844872,
                 -0.01916334,  0.07441156, -0.03949338, -0.06832974, -0.09220405],
                [ 0.08529891,  0.05068012,  0.04445121, -0.00567061, -0.04559945,
                 -0.03419447, -0.03235593, -0.00259226,  0.00286377, -0.02593034]])
```


03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

문제 해결을 위해 당뇨병 환자의 데이터 준비하기

사이킷런에서 당뇨병 환자 데이터 가져오기

3. 타깃데이터 자세히 보기

- [:3]로 표현하면 배열의 첫 번째 요소부터 추출한다는 의미
- 수치가 무엇을 의미하는지 반드시 알아야 필요는 없음. 수치에 대한 해석은 전문가의 영역이고, 우리는 입력 데이터와 타깃 데이터의 수치만 보고 둘 사이의 규칙을 찾으려 함

```
In [6]: ▶ diabetes.target[:3]
```

```
Out [6]: array([151., 75., 141.])
```

```
In [5]: ▶ diabetes.data[0:3]
```

```
Out [5]: array([[ 0.03807591,  0.05068012,  0.06169621,  0.02187235, -0.0442235 ,  
                -0.03482076, -0.04340085, -0.00259226,  0.01990842, -0.01764613],  
               [-0.00188202, -0.04464164, -0.05147406, -0.02632783, -0.00844872,  
                -0.01916334,  0.07441156, -0.03949338, -0.06832974, -0.09220405],  
               [ 0.08529891,  0.05068012,  0.04445121, -0.00567061, -0.04559945,  
                -0.03419447, -0.03235593, -0.00259226,  0.00286377, -0.02593034]])
```

대응

대응

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

문제 해결을 위해 당뇨병 환자의 데이터 준비하기

사이킷런에서 당뇨병 환자 데이터 가져오기



리키의 팁 메모 | 실무에서는 데이터 준비에 많은 공을 들입니다

여러분은 지금까지 미리 준비된 데이터를 사용해 손쉽게 실습했습니다. 하지만 실무에서는 데이터를 준비하는 과정에 많은 시간과 비용이 필요합니다. 충분한 데이터가 없으면 제대로 된 모델을 만들 수 없기 때문이죠. 어떤 경우에는 데이터를 구매하기도 합니다. 그러면 앞으로 사용하게 될 데이터들은 모두 구매해야 할까요? 아닙니다. 실습에서는 학습을 위한 데이터를 제공합니다. 즉, 데이터 이용료는 공짜이며 데이터를 준비하는 시간도 매우 짧습니다. 그래도 실전에서는 데이터를 준비할 때 아주 많은 공을 들여야 한다는 것을 잊지 마세요.

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

당뇨병 환자 데이터 시각화하기

1. 맷플롯립의 함수로 산점도 그리기
 - 당뇨병 데이터 세트에는 10개의 특성이 있으므로 이 특성을 모두 그래프로 표현하려면 3차원 이상의 그래프를 그려야함
 - 3차원 이상의 그래프는 그릴 수 없으므로 1개의 특성만 사용한 예제
 - 세 번째 특성과 타깃 데이터로 산점도를 그림

In [8]: `import matplotlib.pyplot as plt` 세번째 속성에 대한 데이터 (총 442라인)

```
plt.scatter(diabetes.data[:,2], diabetes.target)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

In [12]:

1	<code>x = diabetes.data[2:3,1]</code>
2	<code>print(x)</code>

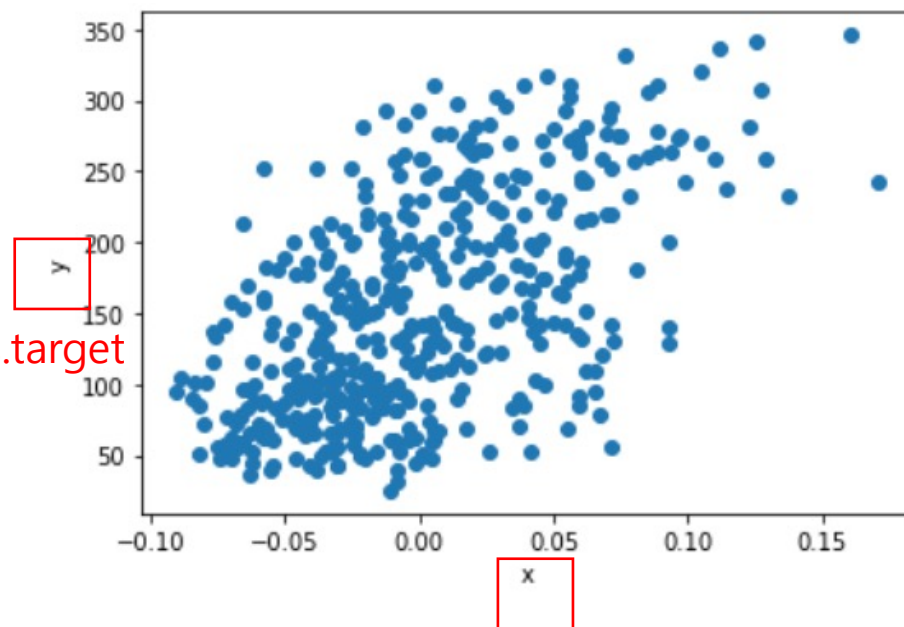
[0.05068012]

In [5]: `diabetes.data[0:3]`

Out [5]:

```
array([[ 0.03807591,  0.05068012,  0.06169621,  0.02187235, -0.0442235 ,
        -0.03482076, -0.04340085, -0.00259226,  0.01990842, -0.01764613],
       [-0.00188202, -0.04464164, -0.05147406, -0.02632783, -0.00844872,
        -0.01916334,  0.07441156, -0.03949338, -0.06832974, -0.09220405],
       [ 0.08529891,  0.05068012,  0.04445121, -0.00567061, -0.04559945,
        -0.03419447, -0.03235593, -0.00259226,  0.00286377, -0.02593034]])
```

우는 딥러닝 입문



입력 데이터와
타깃 데이터가
정비례 관계!!!

Diabetes.data의 세 번째 특성

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

당뇨병 환자 데이터 시각화하기

2. 훈련 데이터 준비하기

매번 `diabetes.data`를 입력하여 입력 데이터의 속성을 참고하는 방법은 번거로우니 입력 데이터의 세 번째 특성(입력 데이터)을 미리 분리하여 변수 x에 저장하고 타겟 데이터는 변수 y에 저장합니다. 이후 실습에서는 x에 있는 데이터와 y에 있는 데이터를 이용해 모델을 훈련할 것입니다.

```
x = diabetes.data[:, 2]
y = diabetes.target
```

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

당뇨병 환자 데이터 시각화하기

- 2. 훈련 데이터 준비하기
 - 선형 회귀 알고리즘 개념을 알아봄
 - 또한, 실제 알고리즘을 만들어보기 위한 당뇨병 데이터 세트를 준비해 봄!
 - 다음 단계에서는 이 데이터를 활용하여 모델을 훈련하기 위한 핵심 *최적화 알고리즘*인 **경사 하강법**에 대해 배우고자 함

03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다

당뇨병 환자 데이터 시각화하기

실습 1) iris 데이터 응용

-keys(), feature_names 기능 사용해보기

```
boston.keys()
dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename'])

boston.feature_names
array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
       'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7')
```

-데이터 import하고 pyplot으로 그래프로 표현하기 (x축은 2번째 속성)

```
import matplotlib.pyplot as plt
from sklearn import datasets
iris = datasets.load_iris()
print(iris.data.shape, iris.target.shape)

(150, 4) (150,)
```

Diabete 데이터와
동일

실습 2) boston 데이터 import 하고 pyplot으로 그래프로 표현하기 (x축은 2번째 속성)