

5주차(1/3)

기계학습 작업 흐름 1

파이썬으로 배우는 기계학습

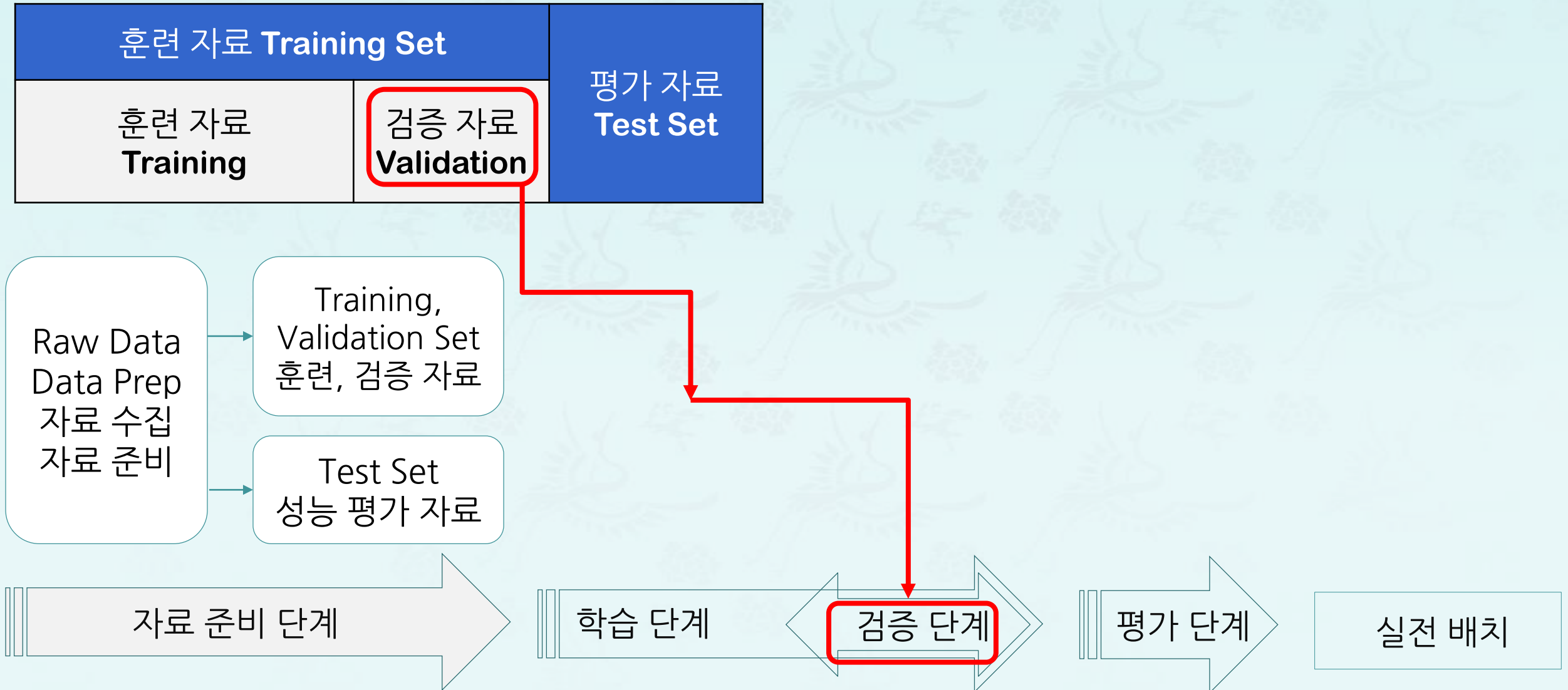
한동대학교
김영섭 교수

기계학습 작업 흐름 1

- 학습 목표
 - 기계학습의 전체 과정을 이해하여 단계별 작업에 필요한 흐름을 이해한다.
- 학습 내용
 - 기계학습 작업 과정에 대한 이해
 - 학습 자료 준비
 - 학습 자료 읽기
 - 학습 자료에서 노이즈

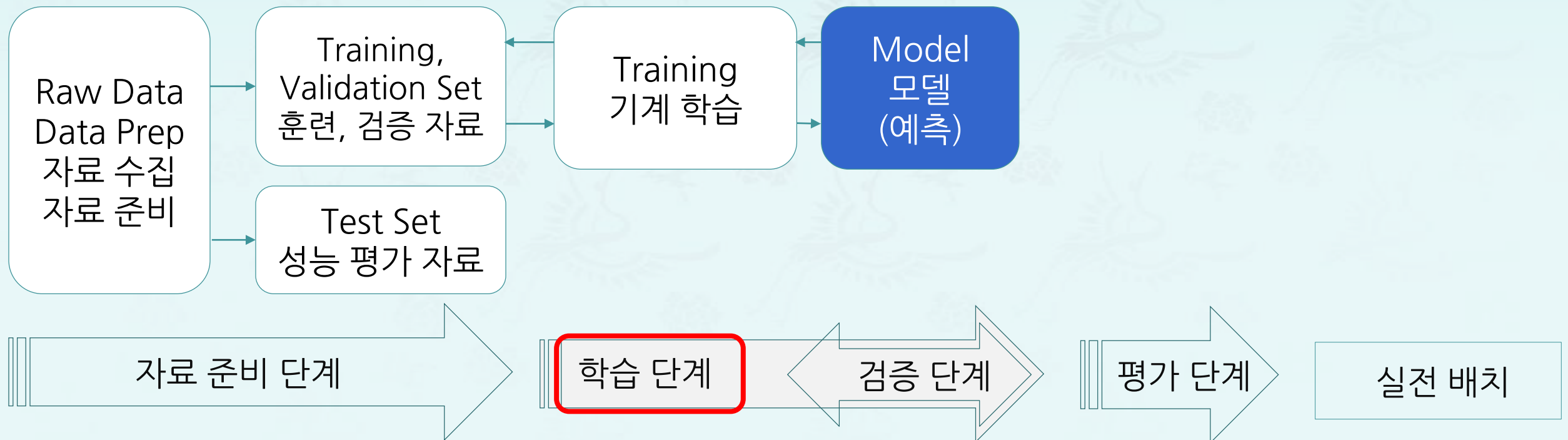
1. 기계학습 작업 흐름: 자료준비단계

1) 자료 준비 단계: 자료 수집과 전처리

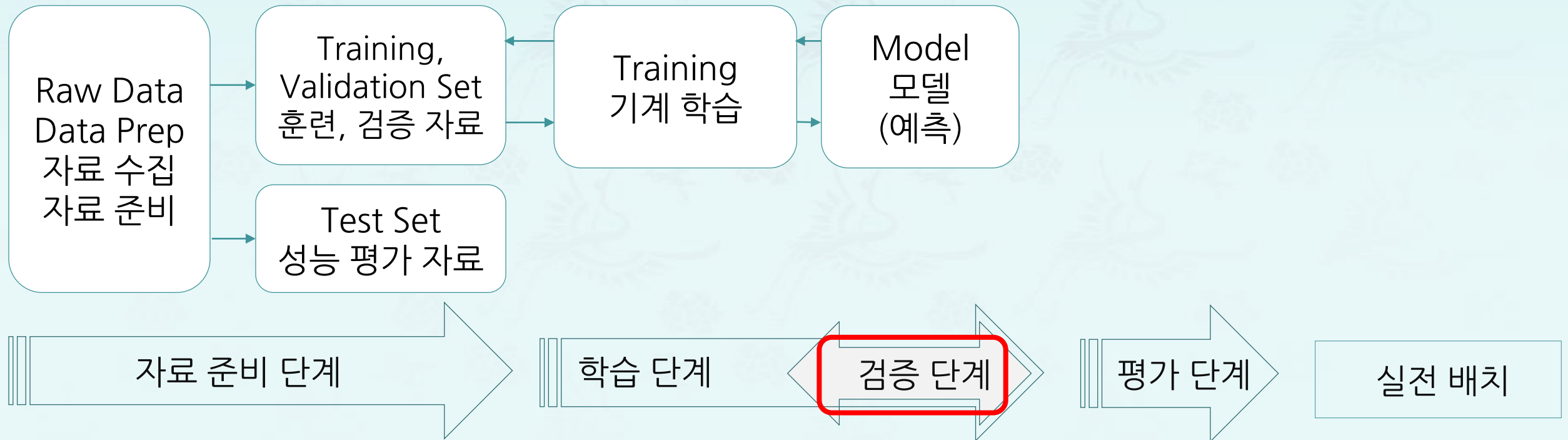


1. 기계학습 작업 흐름 : 학습단계

- 1) 자료 준비 단계: 자료 수집과 전처리
- 2) 학습 단계: 모델 훈련과 완성

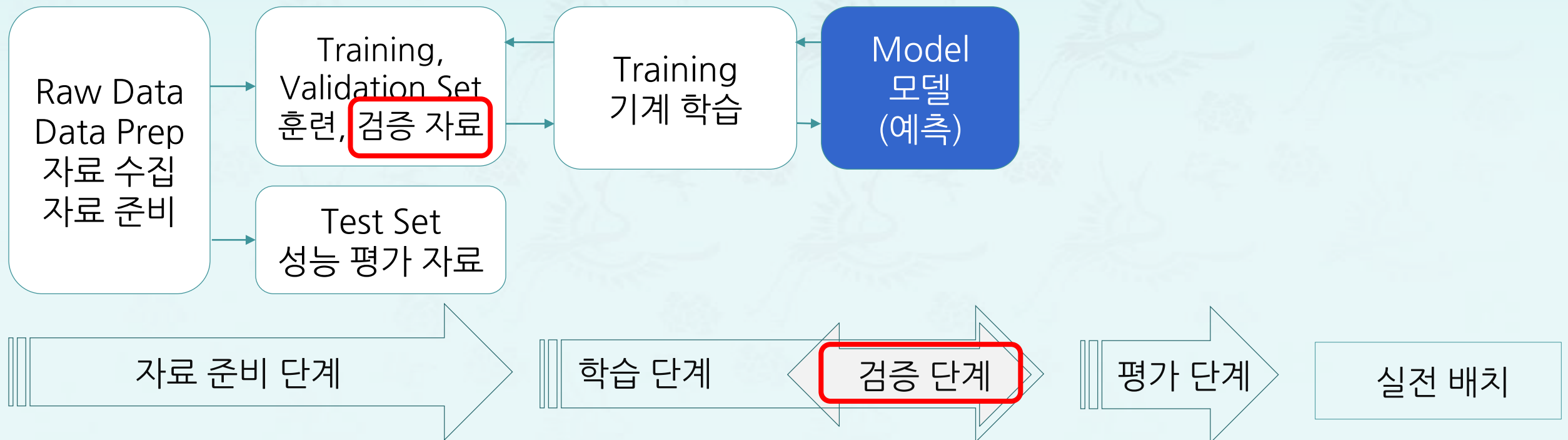


1. 기계학습 작업 흐름 : 검증단계



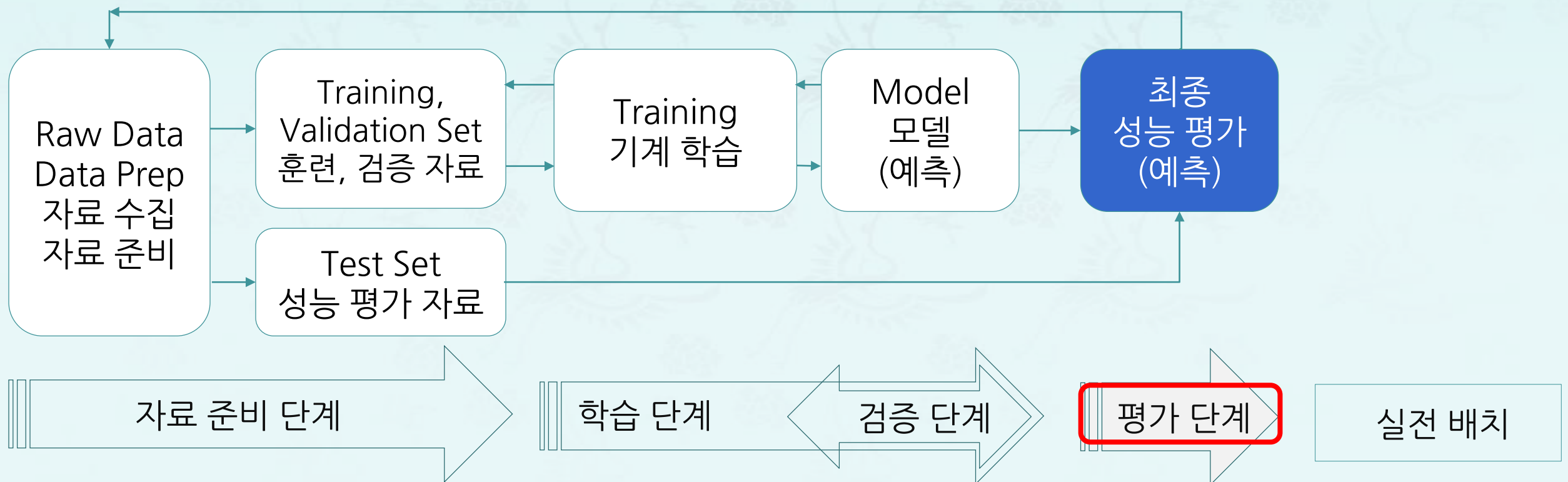
1. 기계학습 작업 흐름 : 검증단계

- 1) 자료 준비 단계: 자료 수집과 전처리
- 2) 학습 단계: 모델 훈련과 완성
- 3) 검증 단계: 하이퍼 파라미터 조정



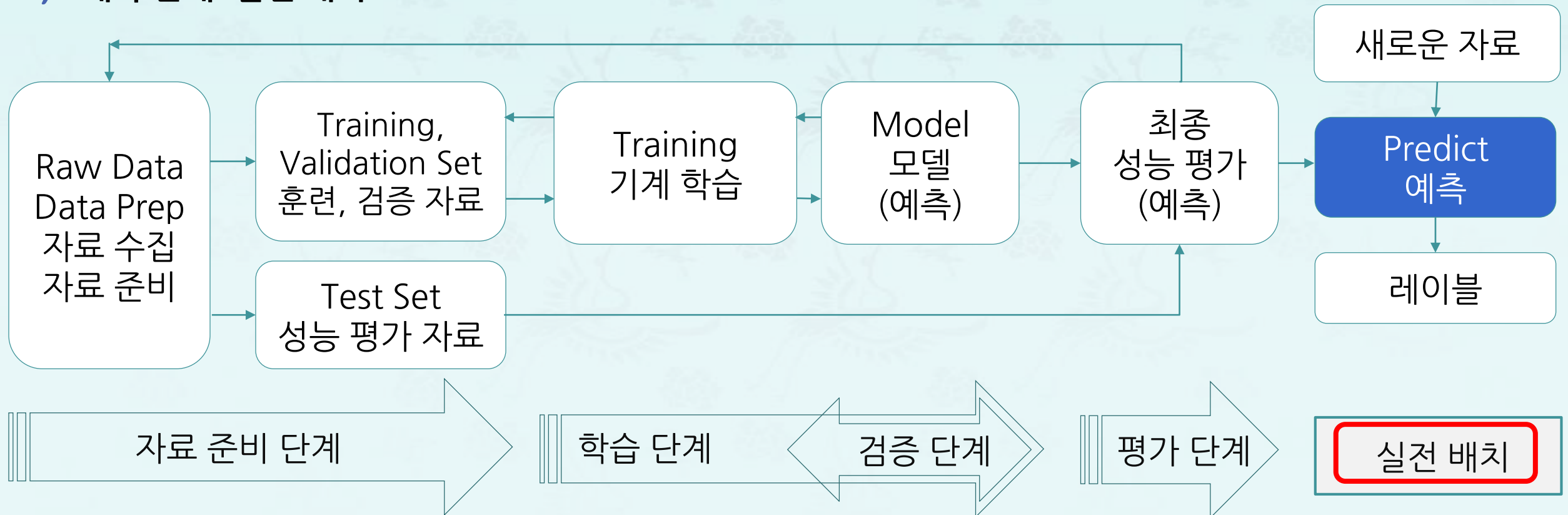
1. 기계학습 작업 흐름 : 평가단계

- 1) 자료 준비 단계: 자료 수집과 전처리
- 2) 학습 단계: 모델 훈련과 완성
- 3) 검증 단계: 하이퍼 파라미터 조정
- 4) 평가 단계: 최종 실전 배치 여부 결정



1. 기계학습 작업 흐름 : 예측단계

- 1) 자료 준비 단계: 자료 수집과 전처리
- 2) 학습 단계: 모델 훈련과 완성
- 3) 검증 단계: 하이퍼 파라미터 조정
- 4) 평가 단계: 최종 실전 배치 여부 결정
- 5) 예측 단계: 실전 배치



2. 학습 자료 준비 단계: 자료 내용

- joydata.txt

```
!cat data/joydata.txt
```

-1.72	-3.12	1
0.31	1.85	1
1.56	2.85	1
2.64	2.41	1
1.23	2.54	1
1.33	2.03	1
1.26	2.68	1
2.58	1.79	1
2.40	0.91	1
0.51	2.44	1

2. 학습 자료 준비 단계: 표기법

특성 feature	클래스 레이블
↓ x1	↓ x2
↓ x3	↓ y
-1.72 0.31 1.56 2.64 1.23 1.33 1.26 2.58 2.40 0.51	-3.12 1.85 2.85 2.41 2.54 2.03 2.68 1.79 0.91 2.44
1 1 1 1 1 1 1 1 1 1	

2. 학습 자료 준비 단계: 표기법

- 입력: \mathbf{x} 혹은 \mathbf{X}
 $x^1, x^2, \dots, x^{(i)}, \dots, x^m$
- 입력의 각 특성: $x_1^i, x_2^i, \dots, x_j^i, \dots, x_n^i$
- 입력 특성 개수: \mathbf{n} (혹은 \mathbf{m})
- 클래스 레이블: \mathbf{y}
 $y^{(1)}, y^{(2)}, \dots, y^{(i)}, \dots, y^{(m)}$
- $x_j^{(i)}$: i 번째 샘플의 j 번째 특성 자료

			x_1	x_2	y
			↓	↓	↓
			data_joydata.txt		
샘플 sample example			-1.72	-3.12	1
			0.31	1.85	1
			1.56	2.85	1
			2.64	2.41	1
			1.23	2.54	1
			1.33	2.03	1
			1.26	2.68	1
			2.58	1.79	1
			2.40	0.91	1
			0.51	2.44	1

2. 학습 자료 준비 단계: 표기법

- 입력: \mathbf{x} 혹은 \mathbf{X}
 $x^1, x^2, \dots, x^{(i)}, \dots, x^m$
- 입력의 각 특성: $x_1^i, x_2^i, \dots, x_j^i, \dots, x_n^i$
- 입력 특성 개수: \mathbf{n} (혹은 \mathbf{m})
- 클래스 레이블: \mathbf{y}
 $y^{(1)}, y^{(2)}, \dots, y^{(i)}, \dots, y^{(m)}$
- $x_j^{(i)}$: i 번째 샘플의 j 번째 특성 자료

	x_1	x_2	y
	data_joydata.txt		
$x_1^{(1)}, x_2^{(1)}, y^{(1)}$	-1.72	-3.12	1
	0.31	1.85	1
	1.56	2.85	1
	2.64	2.41	1
	1.23	2.54	1
	1.33	2.03	1
	1.26	2.68	1
$x_1^{(8)}, x_2^{(8)}, y^{(8)}$	2.58	1.79	1
	2.40	0.91	1
	0.51	2.44	1

2. 학습 자료 준비 단계: 표기법

- 입력: \mathbf{x} 혹은 \mathbf{X}
 $x^1, x^2, \dots, x^{(i)}, \dots, x^m$
- 입력의 각 특성: $x_1^i, x_2^i, \dots, x_j^i, \dots, x_n^i$
- 입력 특성 개수: n (혹은 m)
- 클래스 레이블: \mathbf{y}
 $y^{(1)}, y^{(2)}, \dots, y^{(i)}, \dots, y^{(m)}$
- $x_j^{(i)}$: i 번째 샘플의 j 번째 특성 자료

■ 퀴즈: $x_1^{(3)} = ?$

```
!cat data/joydata.txt
```


-1.72	-3.12	1
0.31	1.85	1
1.56	2.85	1
2.64	2.41	1
1.23	2.54	1
1.33	2.03	1
1.26	2.68	1
2.58	1.79	1
2.40	0.91	1
0.51	2.44	1

2. 학습 자료 준비 단계: 표기법

- 입력: \mathbf{x} 혹은 \mathbf{X}
 $x^1, x^2, \dots, x^{(i)}, \dots, x^m$
- 입력의 각 특성: $x_1^i, x_2^i, \dots, x_j^i, \dots, x_n^i$
- 입력 특성 개수: n (혹은 m)
- 클래스 레이블: \mathbf{y}
 $y^{(1)}, y^{(2)}, \dots, y^{(i)}, \dots, y^{(m)}$
- $x_j^{(i)}$: i 번째 샘플의 j 번째 특성 자료

■ 퀴즈: $x_1^{(3)} = 1.56$

```
!cat data/joydata.txt
```



-1.72	-3.12	1
0.31	1.85	1
1.56	2.85	1
2.64	2.41	1
1.23	2.54	1
1.33	2.03	1
1.26	2.68	1
2.58	1.79	1
2.40	0.91	1
0.51	2.44	1

3. 학습 자료 읽기: 파일 데이터 셋

```
!cat data/joydata.txt
```

-1.72	-3.12	1
0.31	1.85	1
1.56	2.85	1
2.64	2.41	1
1.23	2.54	1
1.33	2.03	1
1.26	2.68	1
2.58	1.79	1
2.40	0.91	1
0.51	2.44	1

3. 학습 자료 읽기: 어떻게 읽을 것인가?

```
!cat data/joydata.txt
```

-1.72	-3.12	1
0.31	1.85	1
1.56	2.85	1
2.64	2.41	1
1.23	2.54	1
1.33	2.03	1
1.26	2.68	1
2.58	1.79	1
2.40	0.91	1
0.51	2.44	1

4. 학습 자료 다루기: 퀴즈

```
1 data = np.genfromtxt('data/joydata.txt')
2 print(data)
```

```
[[ -1.72  -3.12   1.   ]
 [  0.31   1.85   1.   ]
 [  1.56   2.85   1.   ]
 [  2.64   2.41   1.   ]
 [  1.23   2.54   1.   ]
 [  1.33   2.03   1.   ]
 [  1.26   2.68   1.   ]
 [  2.58   1.79   1.   ]
 [  2.4    0.91   1.   ]
 [  0.51   2.44   1.   ]
```

data.shape: (100, 3)

4. 학습 자료 다루기: Slicing 준비

```
1 data = np.genfromtxt('data/joydata.txt')
2 print(data)
```

```
[ [-1.72 -3.12  1.  ]
 [  0.31  1.85  1.  ]
 [  1.56  2.85  1.  ]
 [  2.64  2.41  1.  ]
 [  1.23  2.54  1.  ]
 [  1.33  2.03  1.  ]
 [  1.26  2.68  1.  ]
 [  2.58  1.79  1.  ]
 [  2.4   0.91  1.  ]
 [  0.51  2.44  1.  ]
```

data.shape: (100, 3)

x.shape: (100,2)

y.shape: (100,1)
(100,)

4. 학습 자료 다루기: Slicing 방법

```
1 data = np.genfromtxt('data/joydata.txt')  
2 print(data)
```

```
[ [-1.72 -3.12 1. ]  
  [ 0.31  1.85 1. ]  
  [ 1.56  2.85 1. ]  
  [ 2.64  2.41 1. ]  
  [ 1.23  2.54 1. ]  
  [ 1.33  2.03 1. ]  
  [ 1.26  2.68 1. ]  
  [ 2.58  1.79 1. ]  
  [ 2.4   0.91 1. ]  
  [ 0.51  2.44 1. ]
```

data.shape = (100, 3)

x = data[A, B]

y = data[C, D]


4. 학습 자료 다루기: Slicing 방법

```
1 data = np.genfromtxt('data/joydata.txt')
2 print(data)
```

```
[ [-1.72 -3.12 1. ]
 [ 0.31  1.85 1. ]
 [ 1.56  2.85 1. ]
 [ 2.64  2.41 1. ]
 [ 1.23  2.54 1. ]
 [ 1.33  2.03 1. ]
 [ 1.26  2.68 1. ]
 [ 2.58  1.79 1. ]
 [ 2.4   0.91 1. ]
 [ 0.51  2.44 1. ]
```


data.shape = (100, 3)
x = data[:, :2]
y = data[:, 2]

4. 학습 자료 다루기: Slicing 코드




```
1 data = np.genfromtxt('data/joydata.txt')
2 x, y = data[:, :2], data[:, 2]
3 y = y.astype(np.int)
4 print(x[:5])
5 print(y[:5])
```

4. 학습 자료 다루기: Slicing 코드



```
1 data = np.genfromtxt('data/joydata.txt')
2 x, y = data[:, :2], data[:, 2]
3 y = y.astype(np.int)
4 print(x[:5])
5 print(y[:5])
```

4. 학습 자료 다루기: Slicing 코드



```
1 data = np.genfromtxt('data/joydata.txt')
2 x, y = data[:, :2], data[:, 2]
3 y = y.astype(np.int)
4 print(x[:5])
5 print(y[:5])
```

4. 학습 자료 다루기: 시각화 코드

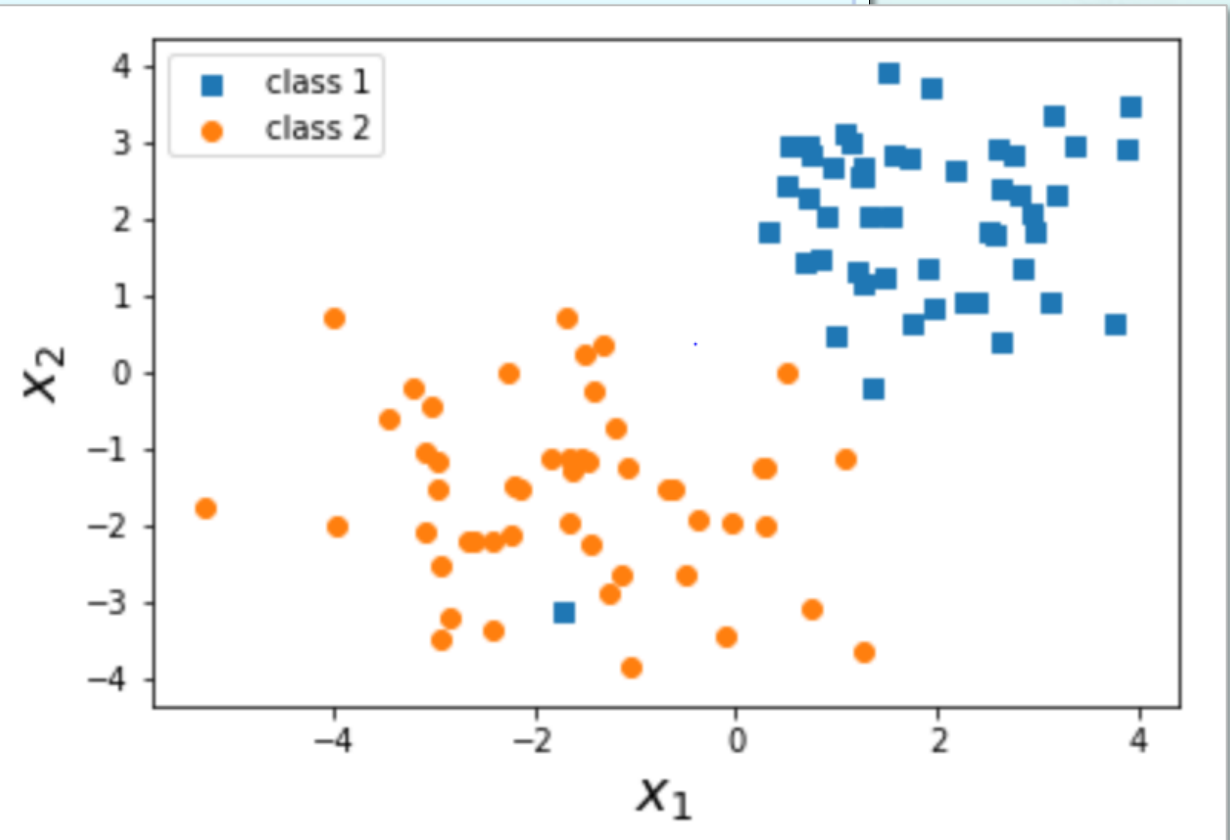
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
data = np.genfromtxt('data/joydata.txt')
x, y = data[:, :2], data[:, 2]
y = y.astype(np.int)

plt.scatter(x[y==1, 0], x[y==1, 1], label='class 1', marker='s')
plt.scatter(x[y==0, 0], x[y==0, 1], label='class 2', marker='o')
plt.xlabel('$x_1$', fontsize=18)
plt.ylabel('$x_2$', fontsize=18)
plt.legend()
plt.show()
```


4. 학습 자료 다루기: 시각화 결과

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
data = np.genfromtxt('data/joydata.txt')
x, y = data[:, :2], data[:, 2]
y = y.astype(np.int)

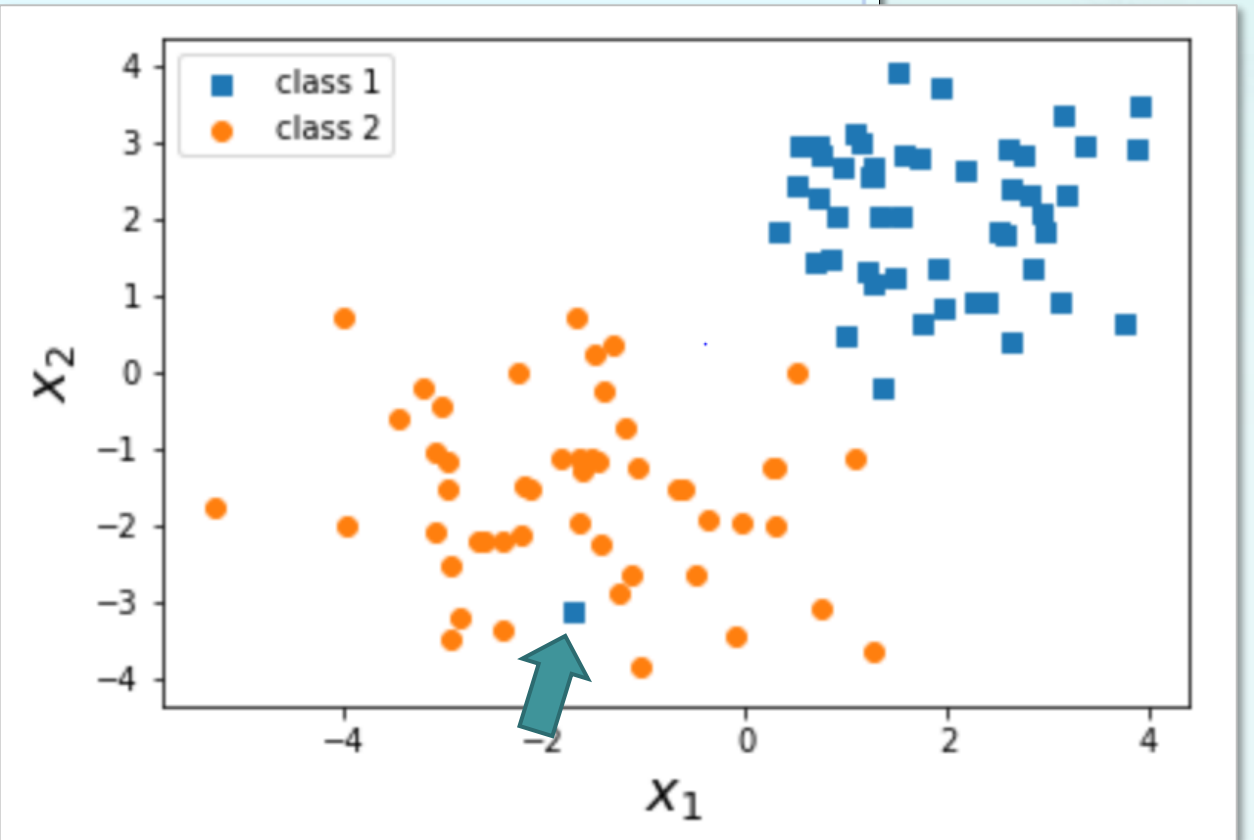
plt.scatter(x[y==1, 0], x[y==1, 1], label='class 1')
plt.scatter(x[y==0, 0], x[y==0, 1], label='class 2')
plt.xlabel('$x_1$', fontsize=18)
plt.ylabel('$x_2$', fontsize=18)
plt.legend()
plt.show()
```



4. 학습 자료 다루기: 노이즈

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
data = np.genfromtxt('data/joydata.txt')
x, y = data[:, :2], data[:, 2]
y = y.astype(np.int)

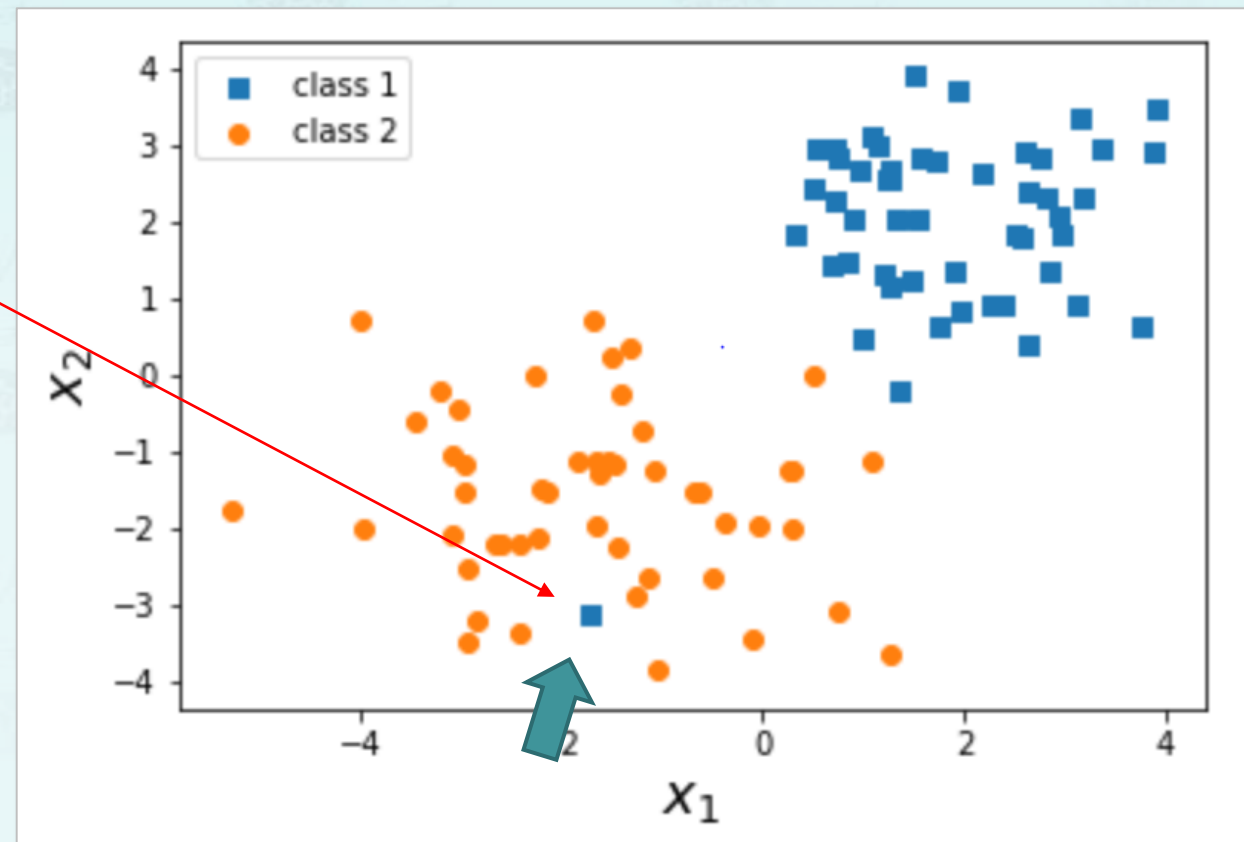
plt.scatter(x[y==1, 0], x[y==1, 1], label='class 1')
plt.scatter(x[y==0, 0], x[y==0, 1], label='class 2')
plt.xlabel('$x_1$', fontsize=18)
plt.ylabel('$x_2$', fontsize=18)
plt.legend()
plt.show()
```



4. 학습 자료 다루기: 노이즈

```
!cat data/joydata.txt
```

-1.72	-3.12	1
0.31	1.85	1
1.56	2.85	1
2.64	2.41	1
1.23	2.54	1
1.33	2.03	1
1.26	2.68	1
2.58	1.79	1
2.40	0.91	1
0.51	2.44	1



기계학습 작업 흐름 1

- 학습 정리
 - 기계학습 작업 과정에 대한 이해
 - 학습 자료 준비
 - 학습 자료 읽기/다루기
 - 학습 자료에서 노이즈