

# Deep Learning with Python

## Chapter 6 DNN 3

## ■ 목차

### 1. 학습 성능을 높이기 위한 데이터 분할

- a. Training set, Validation set, Test set
- b. 2-Way and 3-Way Hold out Method
- c. k-fold cross-validation

### 2. 학습 성능을 높이기 위한 데이터 전처리

- a. Feature engineering
- b. Feature scaling
- c. One-hot encoding
- d. Polynomial Features

## ■ 학습목표

1. 학습 성능을 높이기 위한 데이터 분할 방법을 알아보고  
각 데이터 셋의 목적을 이해한다.
2. Feature Engineering의 개념을 이해하고 Feature Scaling과  
One-hot Encoding, Polynomial Feature의 개념을 배운다.

# 1. 학습 성능을 높이기 위한 데이터 분할

---

# 1. 학습 성능을 높이기 위한 데이터 분할

## Training set, Validation set, Test set의 개념

### \* Training set

- 모델 생성, 학습에 이용하는 데이터 셋으로 학습 데이터 셋이라고 한다.

### \* Validation set

- 모델의 과적합(Overfitting)을 방지하기 위한 데이터 셋으로 검증 데이터 셋이라고 한다.
- 모델의 성능을 높이기 위한 모의 Test set이다.

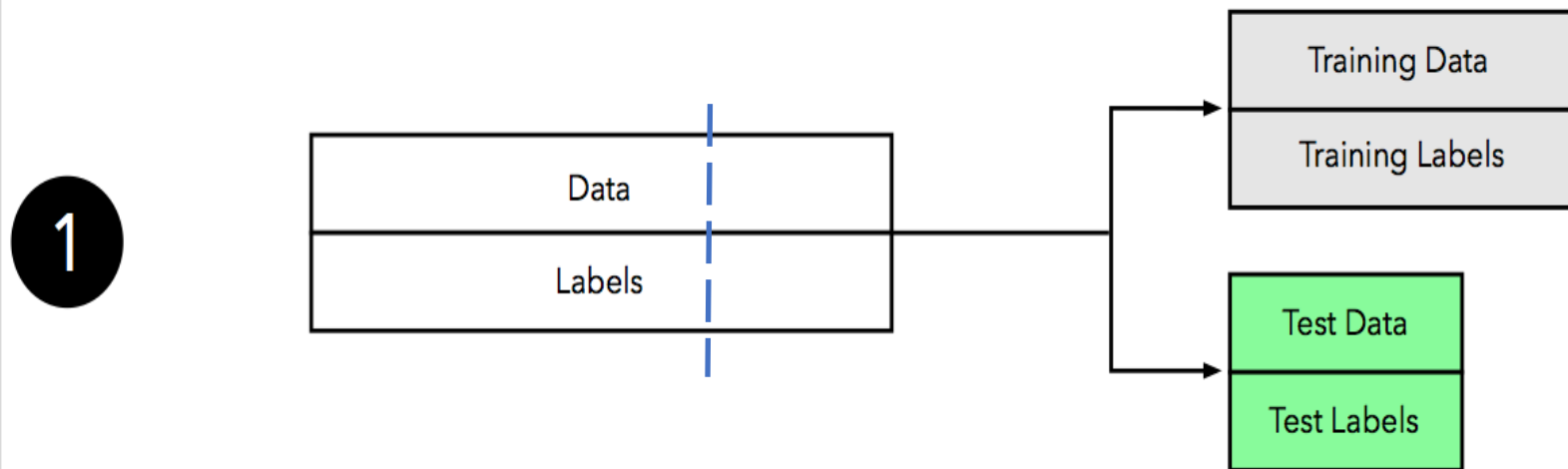
### \* Test set

- 학습용 데이터에 맞춤형으로 만들어진 모델이 다른 상황에도 일반화될 수 있는지 검정하기 위한 데이터 셋이다.
- 모델의 예측 성능을 평가한다.

# 1. 학습 성능을 높이기 위한 데이터 분할

## 🔍 데이터 분할 방법 : 2-Way Holdout Method (1/4)

1. 데이터를 Training set과 Test set 두 개로 분리한다.

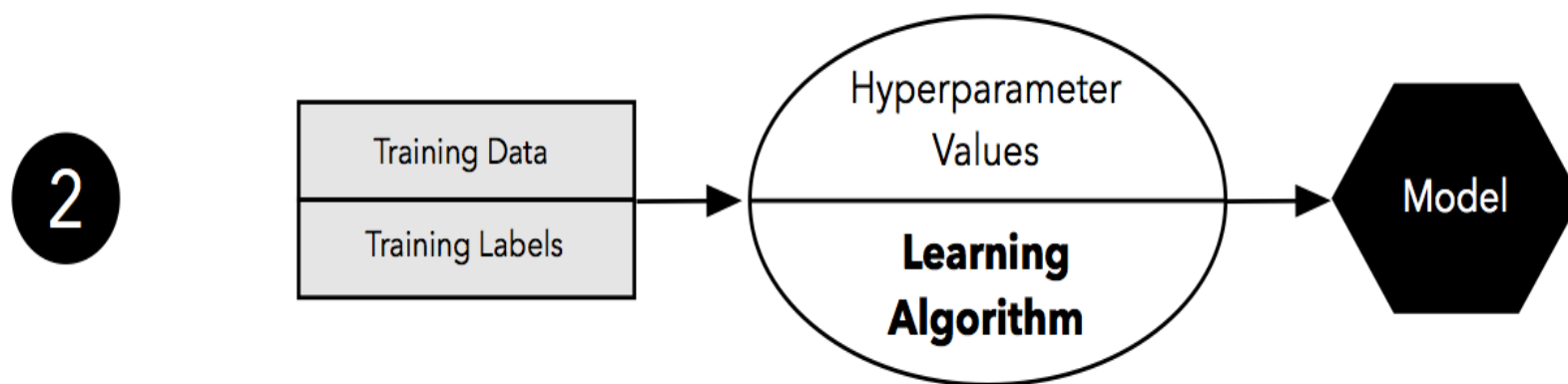


# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : 2-Way Holdout Method (2/4)

2. Training set으로 모델을 학습한다.

Learning Algorithm은 머신러닝, 딥러닝 알고리즘 등 다양하다. 딥러닝에서 Hyperparameter는 학습 과정이 시작되기 전에 미리 세팅하는 값들이다. 예를 들어 Learning rate, Batch size, Epoch 등이 있다. 반대로 Model parameter는 학습 과정을 거쳐서 구하는 값으로 가중치(Weight), 편향(Bias)등이 있다.

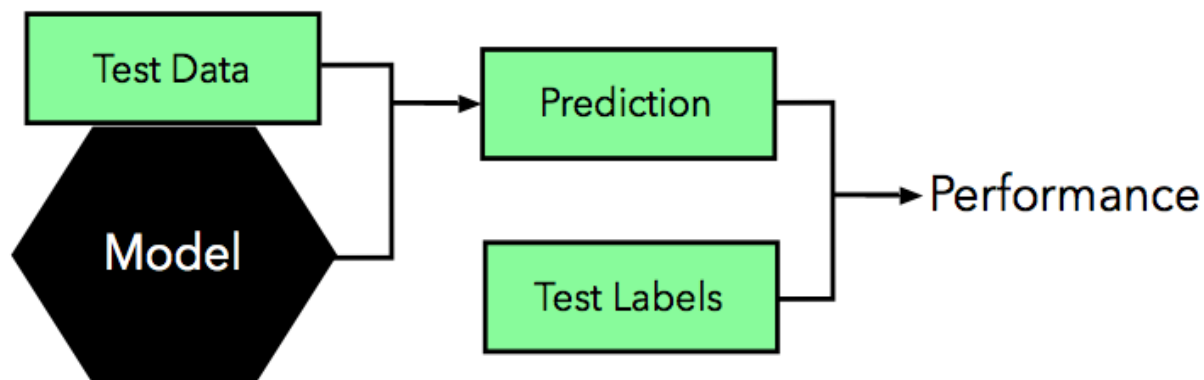


# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : 2-Way Holdout Method (3/4)

3. Training set으로 학습한 모델을 Test set을 대상으로 평가한다.  
평가한 예측값과 Test set의 Label을 비교하여 성능을 측정한다.

3

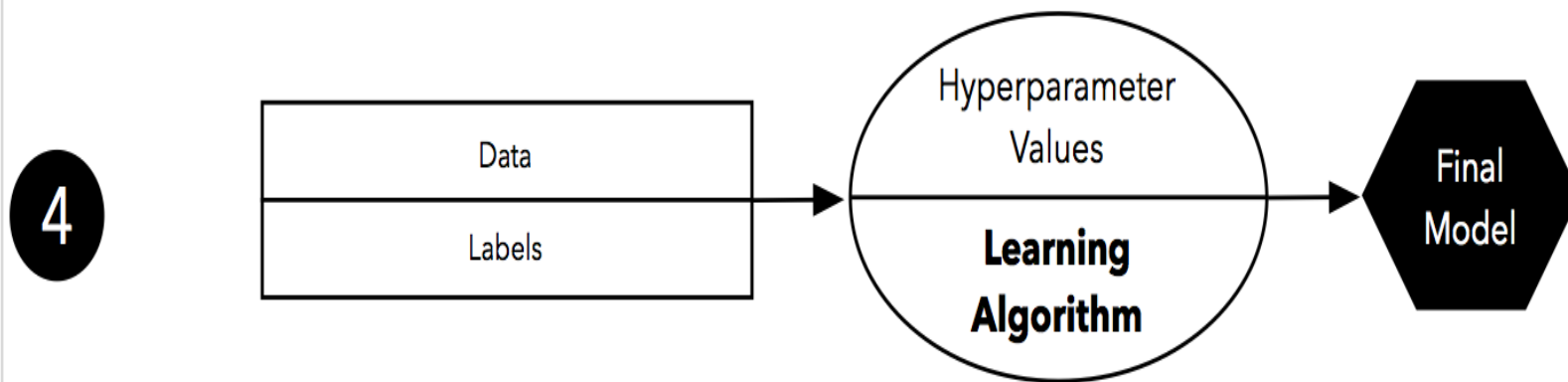




# 1. 학습 성능을 높이기 위한 데이터 분할

## 🔍 데이터 분할 방법 : 2-Way Holdout Method (4/4)

4. 최적의 Hyperparameter Values를 찾기 위해 Data set을 다시 한번 학습하고 최종 모델을 결정한다.



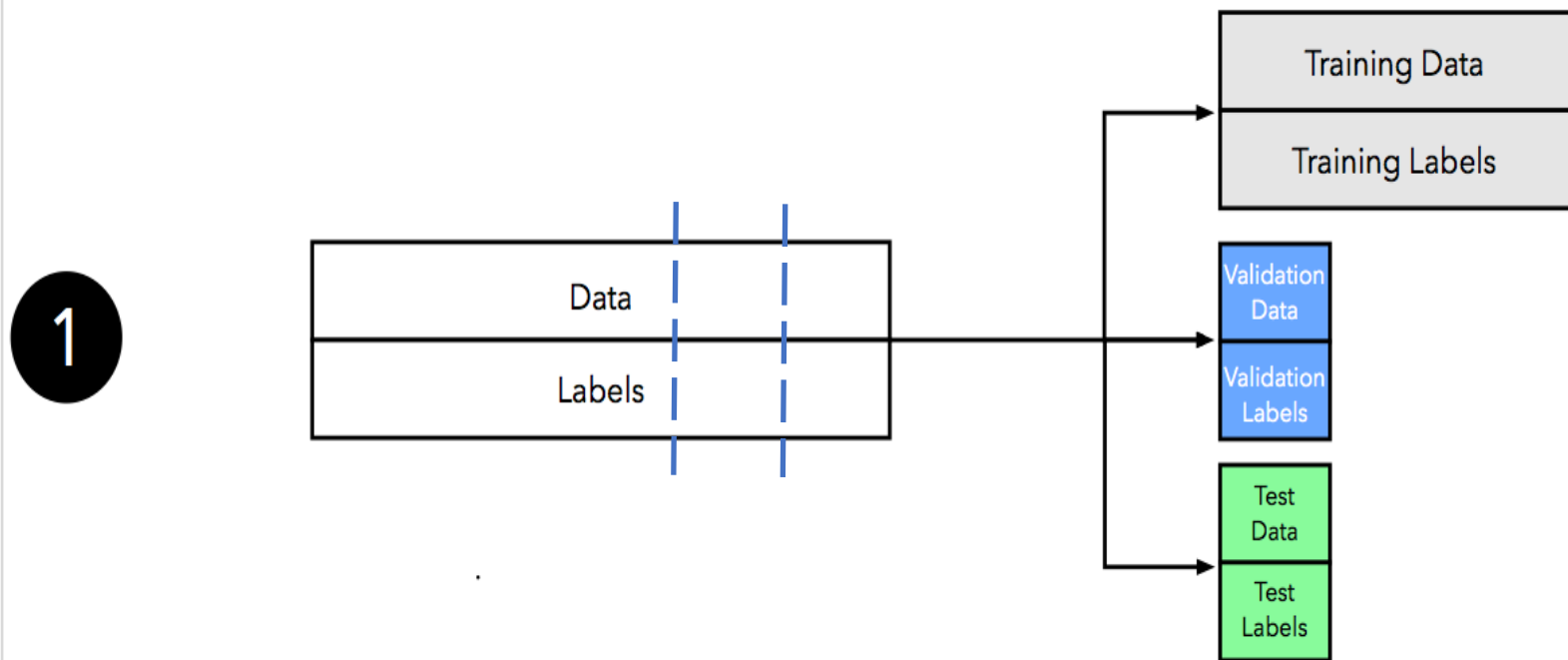
장점 : 데이터 분할이 간단하며 모델 생성에 오랜 시간이 소요되지 않는다.

단점 : Overfitting 방지가 어렵다.

# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : 3-Way Holdout Method (1/6)

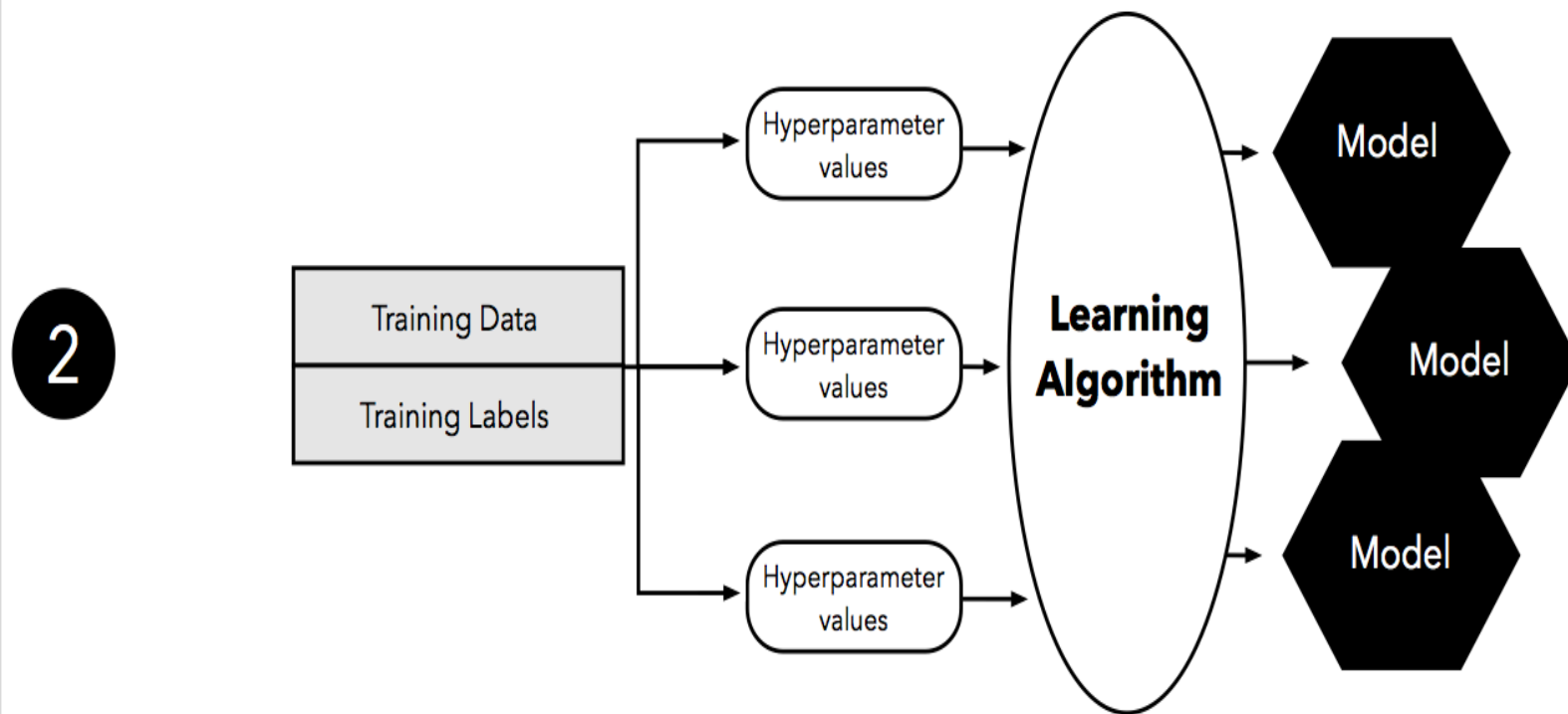
1. 데이터를 Training set, Validation set, Test set으로 분리한다.



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : 3-Way Holdout Method (2/6)

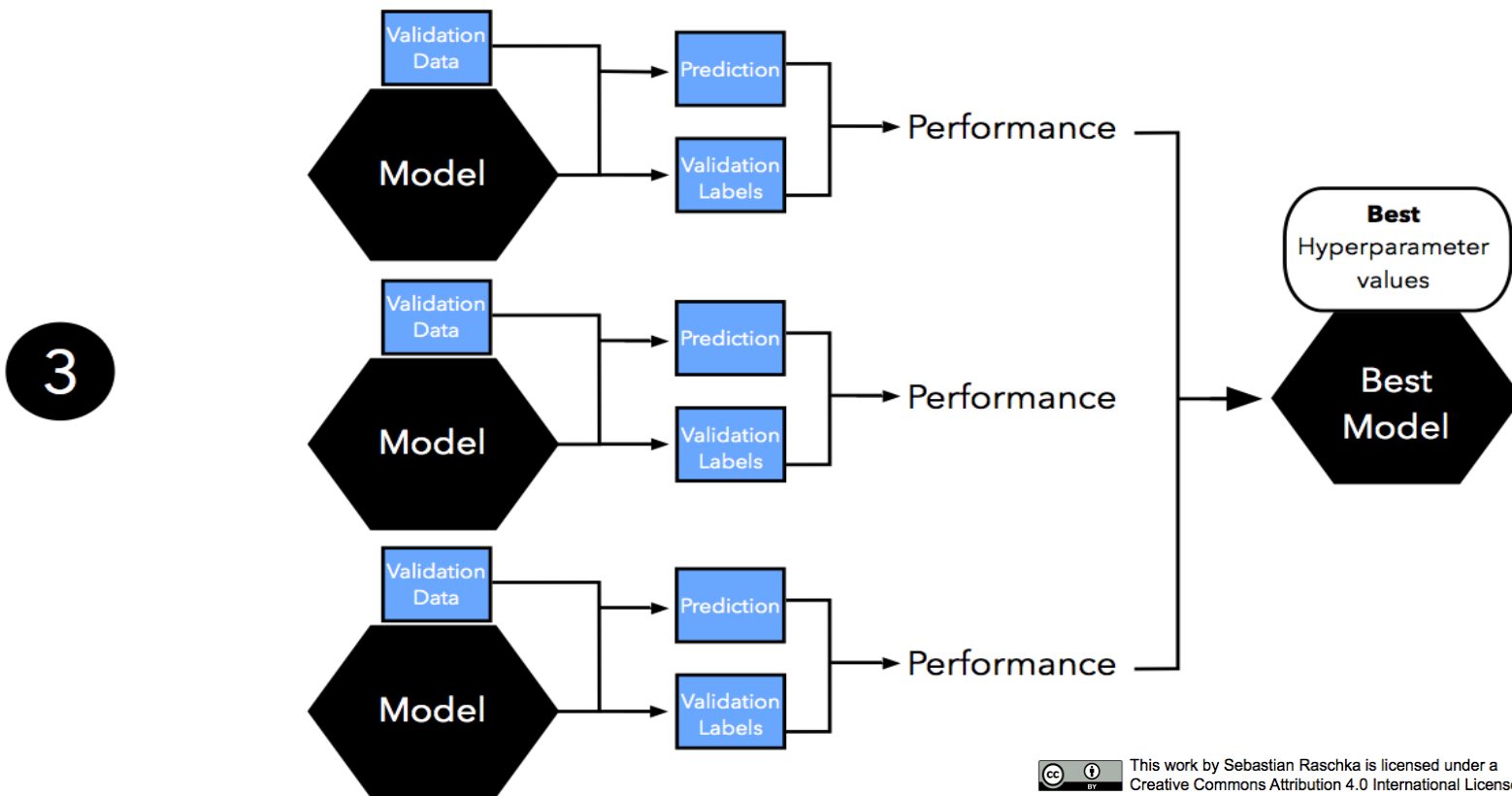
2. 여러 Parameter sets으로 후보 모델들을 학습한다.



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : 3-Way Holdout Method (3/6)

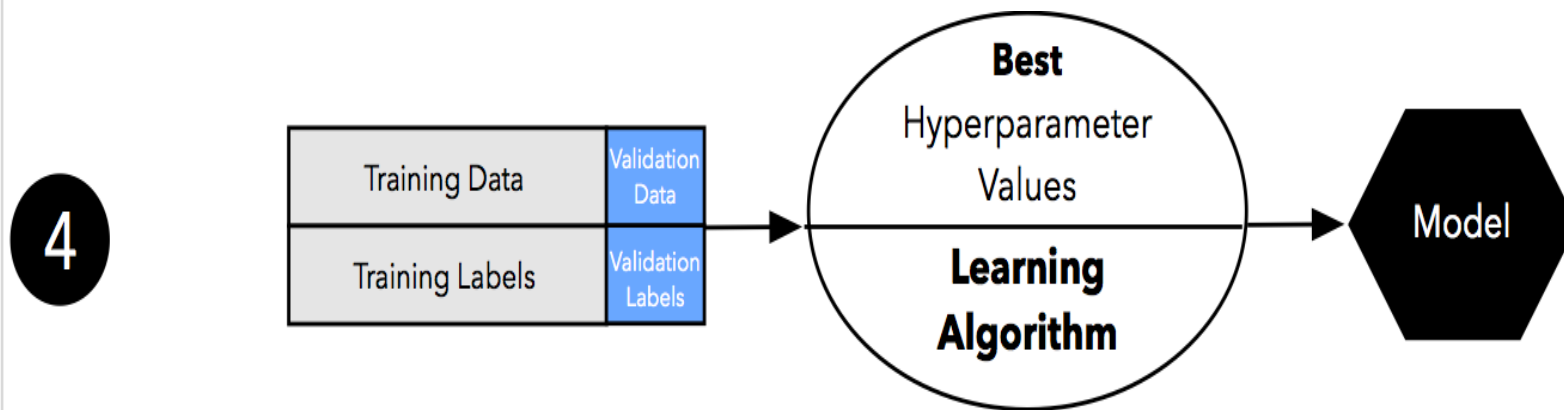
3. Validation set을 이용하여 모델들을 평가하고,  
최적의 parameter set을 결정한다.



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : 3-Way Holdout Method (4/6)

4. Training set과 Validation set을 합쳐,  
Best parameter set으로 모델을 재학습한다.

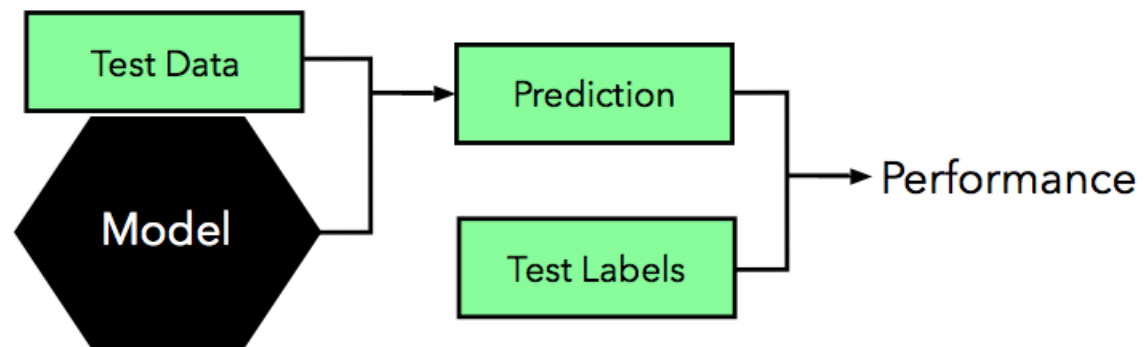


# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : 3-Way Holdout Method (5/6)

5. Training set + Validation set으로 학습한 모델을 Test set을 대상으로 평가한다. 예측값과 Test set의 Label을 비교하여 성능을 측정한다.

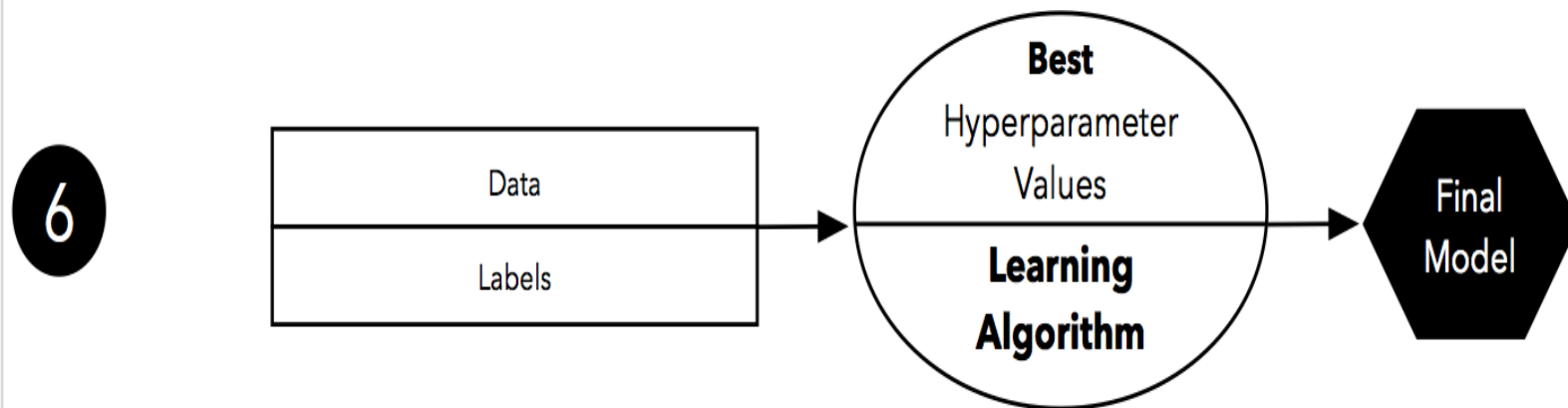
5



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : 3-Way Holdout Method (6/6)

6. 최적의 Hyperparameter Values를 찾기 위해 Data set을 다시 한번 학습하고 최종 모델을 결정한다.



장점 : 2-Way holdout에 비해 학습한 모델의 일반화 성능을 Validation set으로 찾을 수 있다.

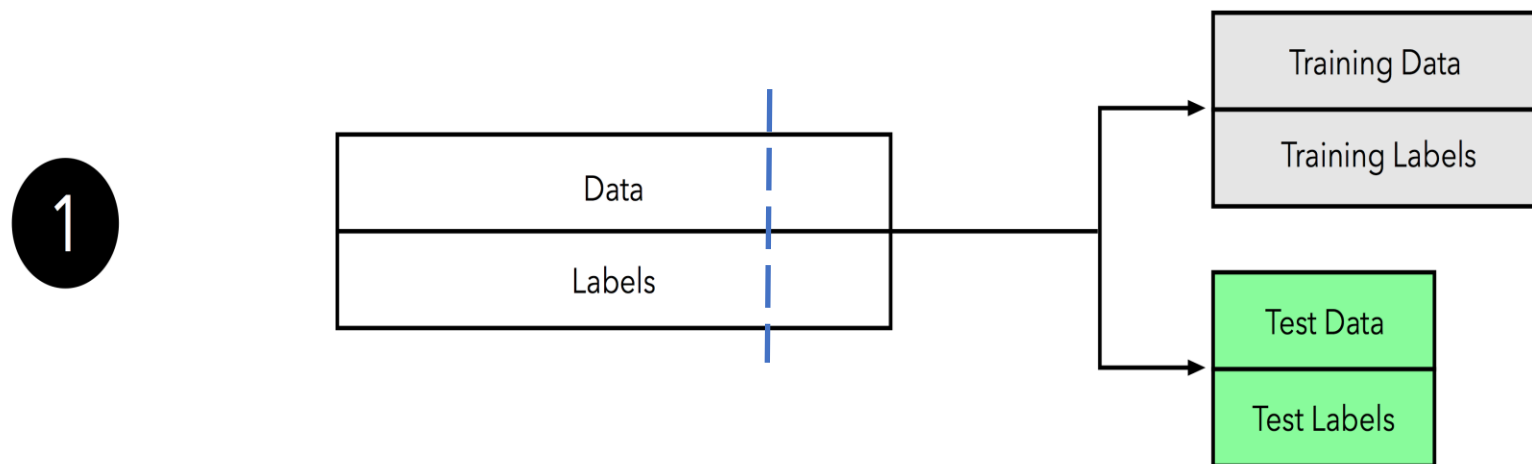
단점 : Training set의 크기가 원 데이터 셋에 비해 작다.

# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : k-fold cross-validation (1/6)

: k-fold CV방법을 단계별로 살펴보면 다음과 같다.

1. 데이터를 Training set, Test set으로 분리한다.

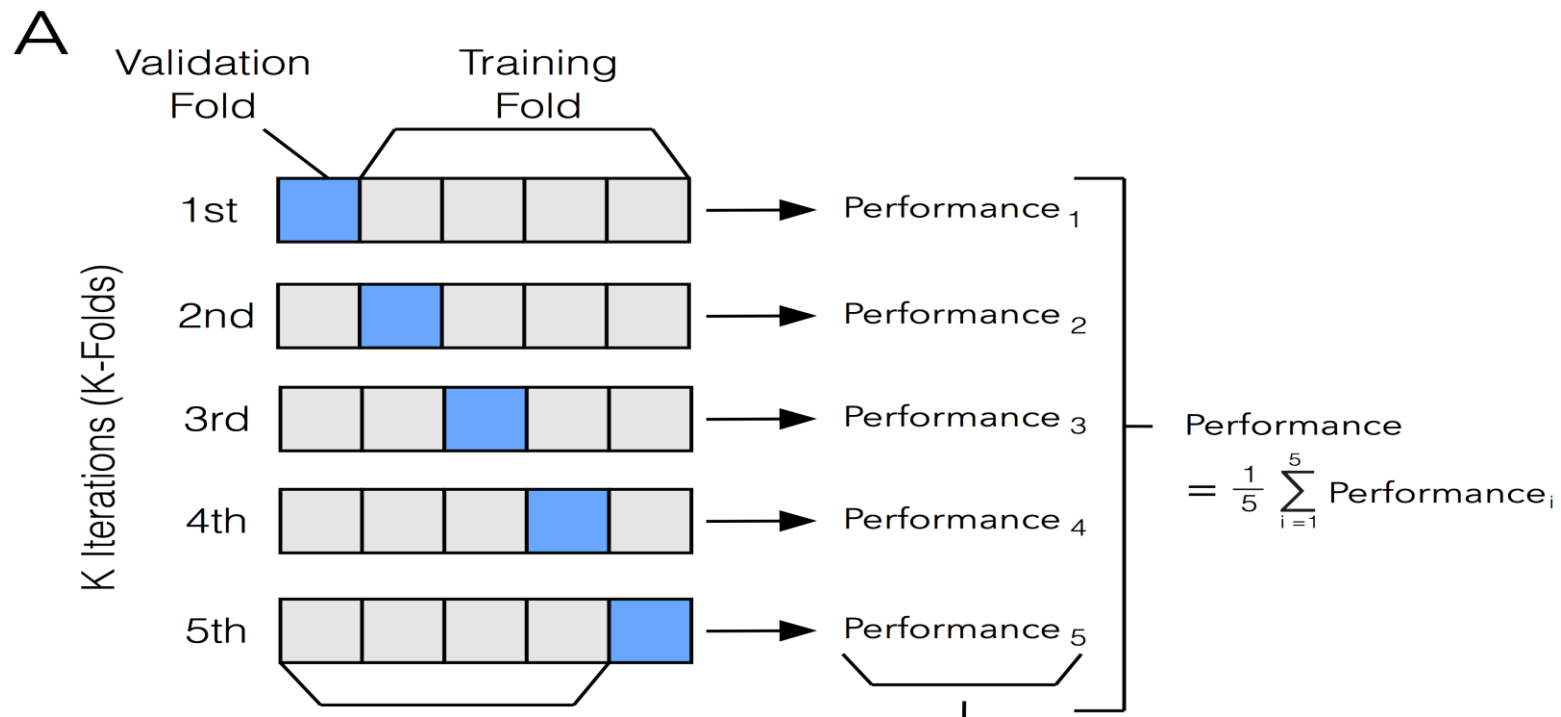




# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : k-fold cross-validation (2/6)

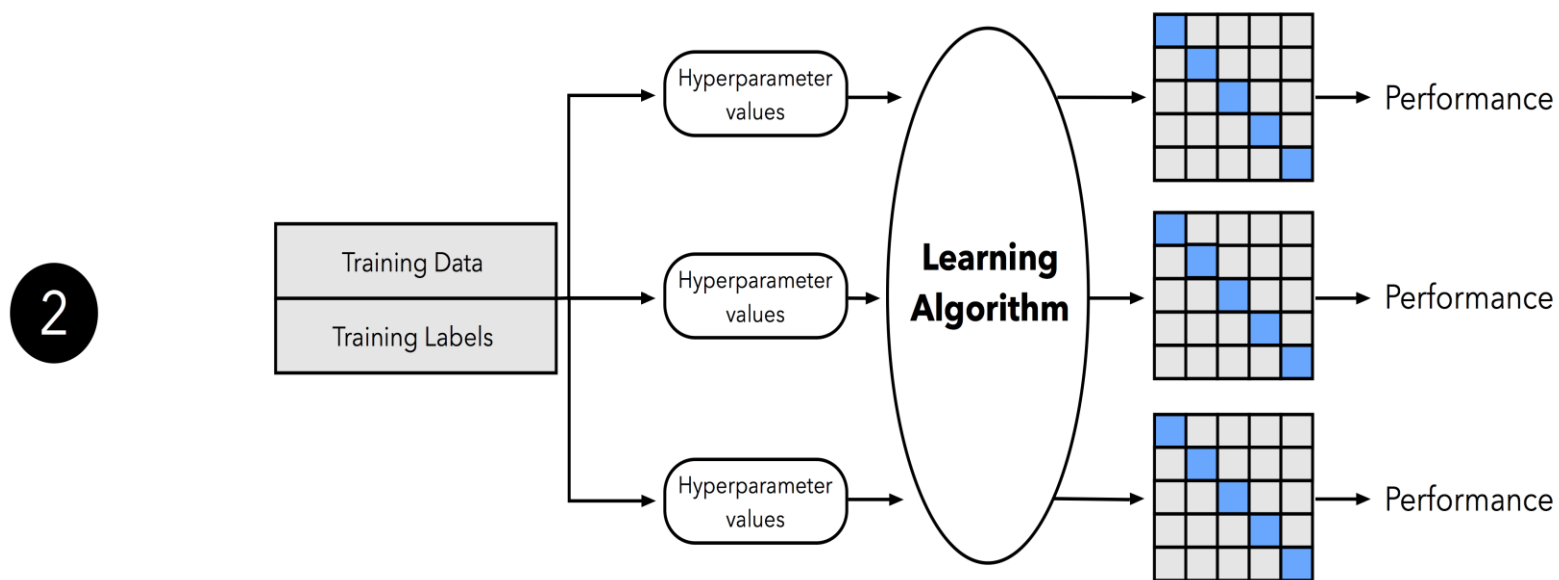
: 2-Way holdout 방법에서 Training set을 k개의 folds로 분리하여 학습하는 방법이다. k개의 folds 중 하나를 Validation fold 나머지를 Training fold로 사용한다.



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : k-fold cross-validation (3/6)

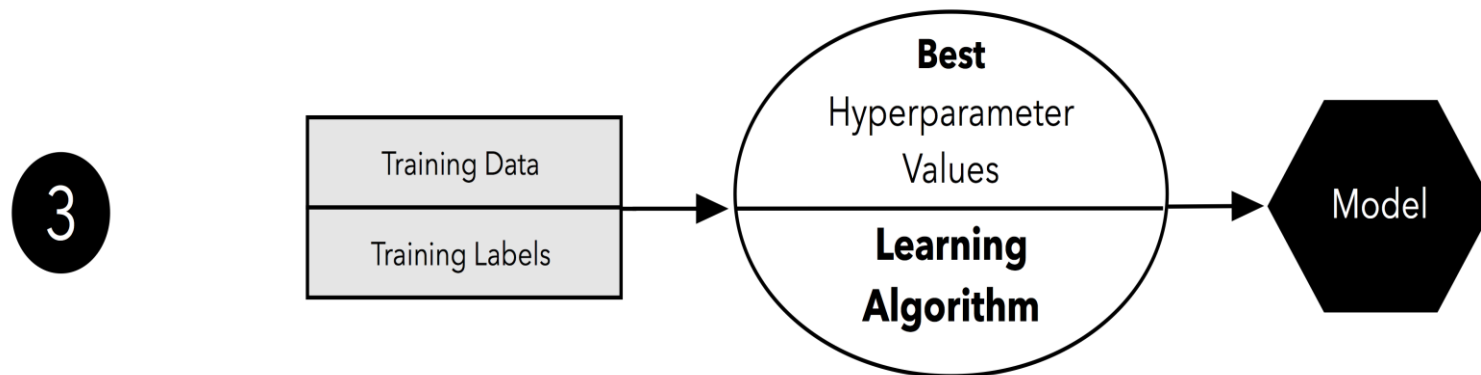
2. Training set을 이용하여 여러 개의 parameter set에 대해 k-fold cross-validation을 수행한다.



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : k-fold cross-validation (4/6)

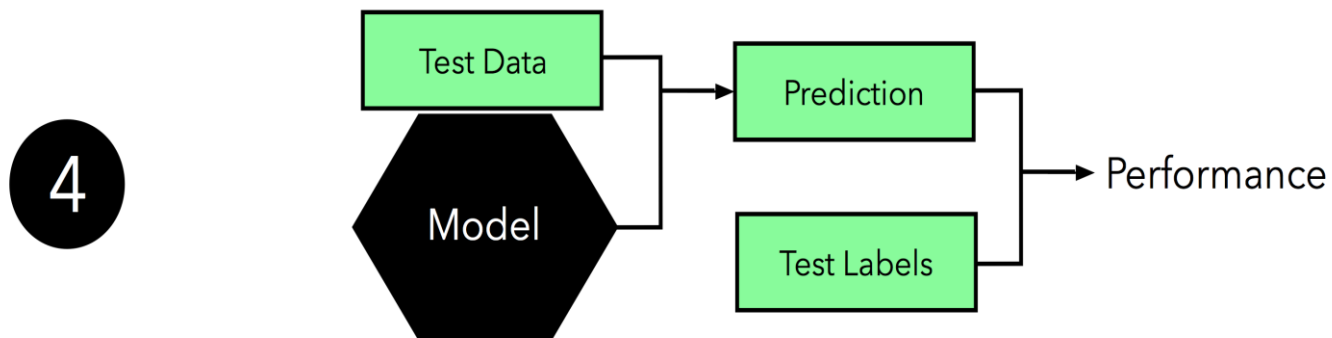
3. k-fold corss-validation 학습 결과가 가장 좋은 parameter set을 선택하고, 이를 이용해 training set에 모델을 학습한다.



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : k-fold cross-validation (5/6)

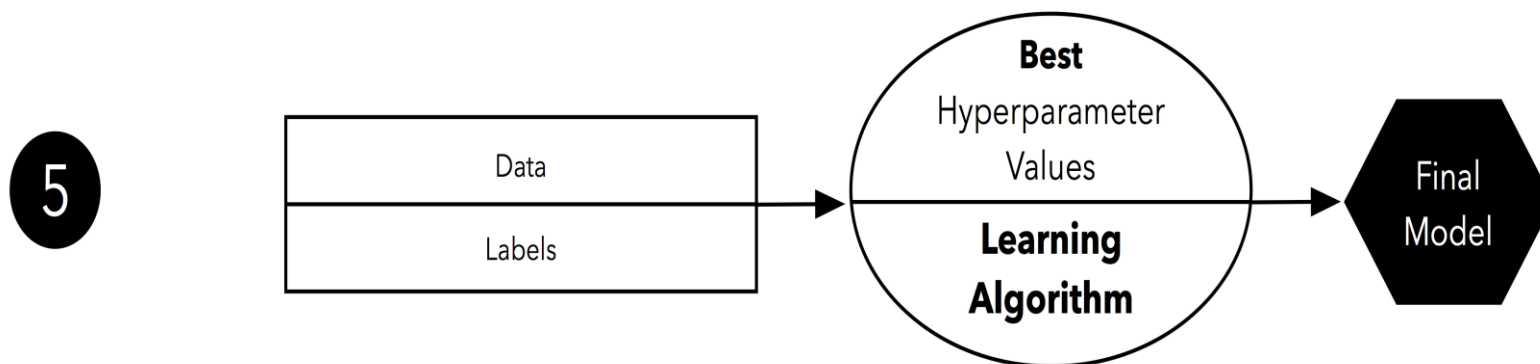
4. 학습한 모델을 Test set을 대상으로 평가한다.  
예측값과 Test set의 Label을 비교하여 성능을 측정한다.



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 방법 : k-fold cross-validation (6/6)

5. 전체 데이터를 대상으로 최종 모델을 학습한다.



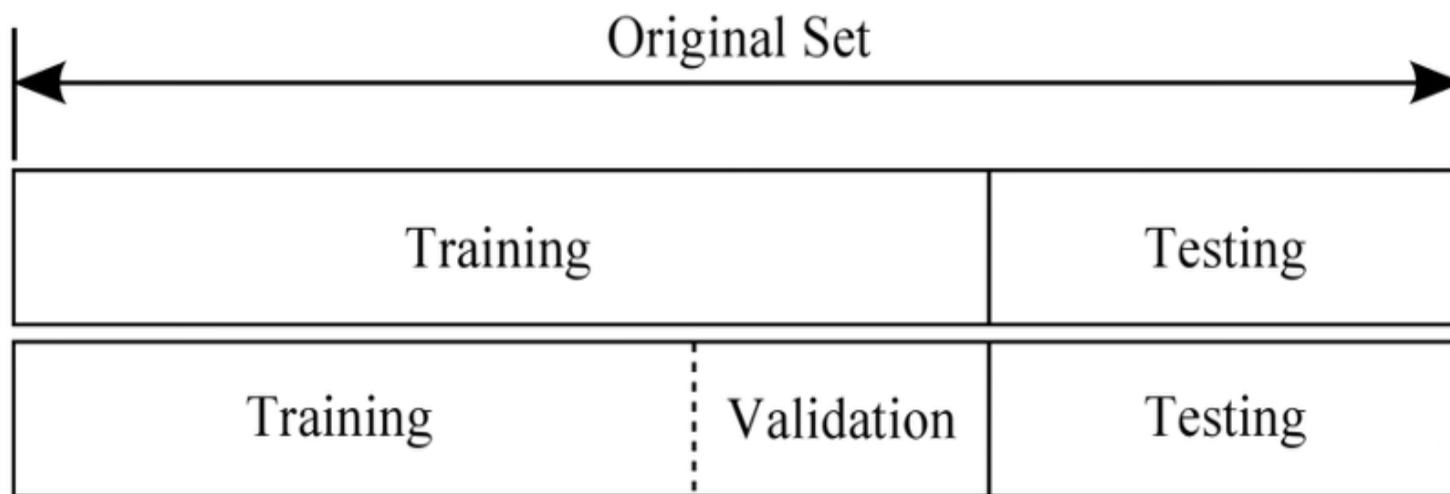
장점 : 겹치지 않는 Validation set을 이용하므로 모델의 편향(bias)과 분산(variance)을 측정하기에 용이하다.

단점 : 여러 번 모델 학습이 이뤄지므로 학습 시간이 오래 걸린다.

# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 시 고려해야 할 사항 (1/2)

- 전체 데이터를 분할할 때 비율은 보통 다음과 같다.
- Training 70% / Test 30%
- Training 50% / Validation 25% / Test 25%
- Training 60% / Validation 20% / Test 20%



# 1. 학습 성능을 높이기 위한 데이터 분할

## 데이터 분할 시 고려해야 할 사항 (2/2)

- 데이터 분할 전에 전체 데이터 셋을 random하게 섞는 것이 좋다
- `sklearn.util.shuffle` 함수 이용
- `sklearn.model_selection.train_test_split` 함수로 분할을 함께 처리
- 다만 시계열 데이터처럼 앞뒤 순서가 있는 경우 섞지 않은 것이 좋다.

## 2. 학습 성능을 높이기 위한 데이터 전처리




## 2. 학습 성능을 높이기 위한 데이터 전처리

### Feature란?

- 현상들을 설명하고 표현하는 요소로 변수 또는 특징이라고 한다.
- 다른 말로 Variable, Attribute, Factor, Field, Column이라고도 한다.
- 머신러닝/딥러닝에서는 입력 변수를 Feature라고 한다.

Predictor variables(예측 변수)  
Input variables(입력 변수)  
Independent variables(독립 변수)  
Feature(특징)

Target variables(타겟 변수)  
Output variables(출력 변수)  
Dependent variables(종속 변수)



ID	$X_1$	$X_2$	...	$X_p$	Y
1	$X_{11}$	$X_{12}$	...	$X_{1p}$	$Y_1$
2	$X_{21}$	$X_{22}$	...	$X_{2p}$	$Y_2$
...	...	...	...	...	...
n	$X_{n1}$	$X_{n2}$	...	$X_{np}$	$Y_n$

## 2. 학습 성능을 높이기 위한 데이터 전처리

### Feature Engineering 이란?

- 모델의 성능을 높이기 위해 주어진 초기 데이터로부터 특징(Feature)을 가공하고 생성하여 모델에 입력할 데이터를 만드는 전체 과정을 의미한다.
- Feature Engineering을 데이터 사이언스 단계로 살펴보면 다음과 같다.
  1. Project Scoping (문제 정의)
  2. Data Collection (데이터 수집)
  3. EDA (탐색적 자료분석)
  4. Data Preprocessing (데이터 전처리)
  - 5. Feature Engineering (데이터 가공)**
  6. Modeling (모델링)
  7. Evaluation (모델 평가)
  8. Project Delivery / Insights (통찰력 제고)

## 2. 학습 성능을 높이기 위한 데이터 전처리

### Feature Engineering 의 구성

- Feature Engineering은 Feature Selection(특징 선택), Feature Extraction(특징 추출), Feature Transformation and Generation(특징 변형과 생성)으로 구성된다.

#### 1. Feature Selection(특징 선택)

- 특징 선택의 목적은 원본 데이터에서 불필요한 특징 집합을 제거하여 간결한 특징 집합을 만드는 것이다.

#### 2. Feature Extraction(특징 추출)

- 원본 데이터들의 특징 조합으로 새로운 특징을 만들어 내는 것이다.

#### 3. Feature Transformation and Generation(특징 변형과 생성)

- 특징 변형과 생성에는 Feature Scaling, One-hot Encoding 등의 방법이 있다.

## 2. 학습 성능을 높이기 위한 데이터 전처리



### Feature Scaling

- Feature Scaling이란 Feature(입력값)의 범위를 조정하는 작업으로 Feature간의 범위 차이가 크다면 경사하강법을 적용하기가 어려워진다.

X1	X2
1	2000
3	-6000
7	1000
4	-5000
⋮	⋮

## 2. 학습 성능을 높이기 위한 데이터 전처리

### Feature Scaling 방법

#### Centering

- 각 Feature(입력값)의 평균을 추출하는 방법으로 Feature를 0을 중심으로 만든다. Centering에서 Scaling한 입력값은 다음과 같다.

$$X' = X - \mu_x, \quad \mu_x : X \text{의 평균}$$

#### Standardization

- Centering 방법에서 0을 중심으로 조정된 입력값에 표준편차로 나눈다.

$$X' = \frac{X - \mu_x}{\sigma_x}, \quad \mu_x : X \text{의 평균}, \sigma_x : X \text{의 표준편차}$$

#### min-max scaling

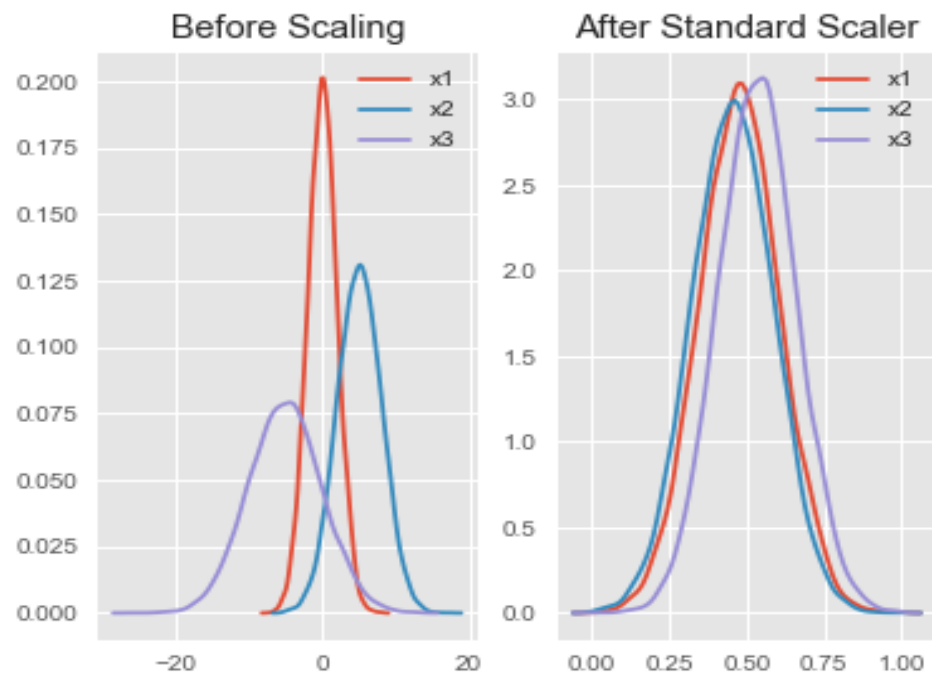
- [최솟값, 최댓값]의 고정된 범위에 입력값을 Scaling 한다.

$$[0,1] \text{ scaling: } X' = \frac{1}{\max(x) - \min(x)}(X - \min(x))$$

## 2. 학습 성능을 높이기 위한 데이터 전처리

### Feature Scaling 예시

- 아래 예시처럼 Feature Scaling으로 입력값의 범위를 조정하면 경사하강법을 적용하기가 용이하다.



Standardization 예시

## 2. 학습 성능을 높이기 위한 데이터 전처리

### One-hot Encoding

- One-hot Encoding이란 단 하나의 값만 True이고 나머지는 모두 False인 Encoding을 의미한다.
- 계절이라는 범주형 변수로 One-hot Encoding하면 다음과 같다.

ID	계절
1	봄
2	여름
3	가을
4	겨울
⋮	⋮



ID	봄	여름	가을	겨울
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
⋮	⋮	⋮	⋮	⋮

## 2. 학습 성능을 높이기 위한 데이터 전처리

### One-hot Encoding

- ‘계절’이라는 하나의 feature에 1~4의 값을 담는 것보다 [봄, 여름, 가을, 겨울] 4개의 feature에 one-hot encoding하는 것이 학습에 유리하다
- ‘계절’이라는 하나의 feature에 값을 담으면 해당 feature에 대응하는 가중치가 점진적으로 학습되며 2.5와 같은 연속적이고 불명확한 중간 값을 가질 수 있기 때문이다.
- [봄, 여름, 가을, 겨울] 4개 feature로 명확히 나누게 되면 불명확한 중간 값이 없어지고 해당 feature를 정확하게 학습시킬 수 있다.



## 2. 학습 성능을 높이기 위한 데이터 전처리



### Polynomial Features

- 주어진 feature를 조합하여 새로운 feature를 만들어낼 수 있다.
- 주택 가격을 예측하는 모델을 만든다고 가정했을 때 feature로 가로 세로 길이가 주어진다면 이 둘을 곱한 '면적'을 추가하는 것이 학습에 유리하다.
- 비선형의 특징을 가지는 데이터라면 해당 비선형을 표현할 수 있는 함수로 feature를 변형하여 추가하면 학습이 더 잘된다.

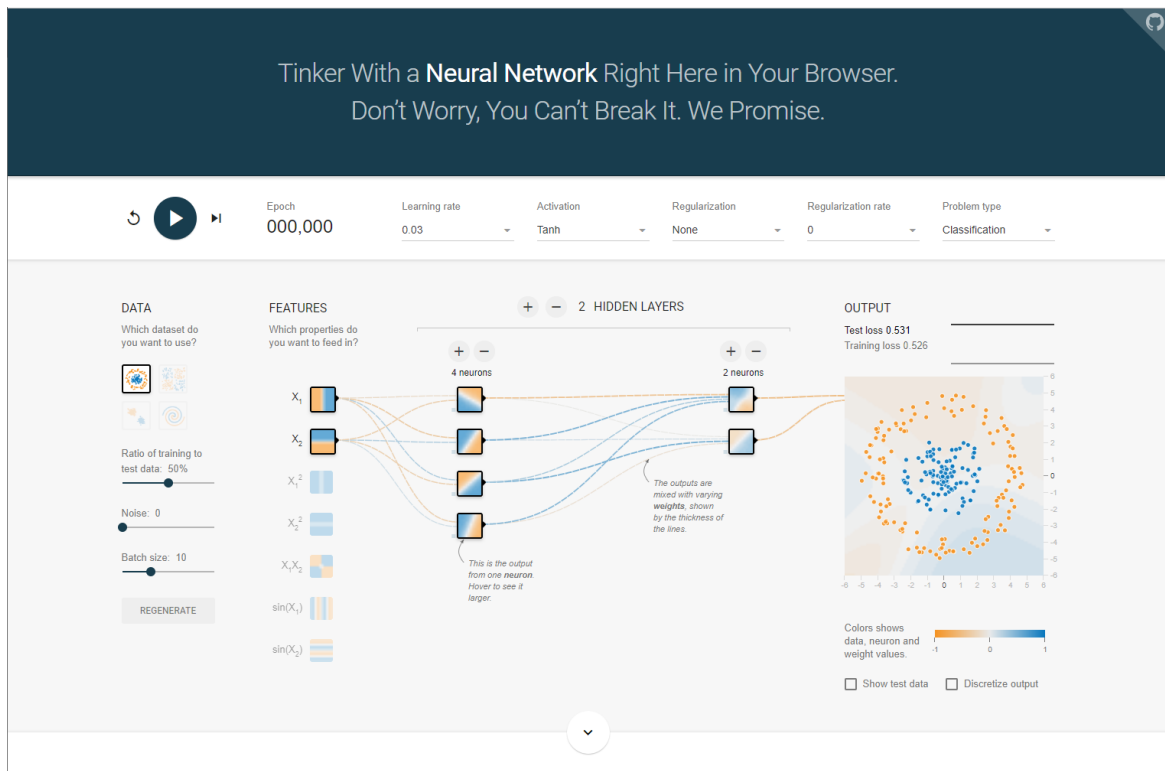
## 2. 학습 성능을 높이기 위한 데이터 전처리



### Polynomial Features

- Neural Network Playground 예제

<https://playground.tensorflow.org>



감사합니다.

## 참고자료(Reference)

“모두를 위한 딥러닝 강좌 시즌 1”, 김성훈

1. MNIST 문자인식

<http://solarisailab.com/archives/303>

<https://localab.jp/blog/simple-neural-network-using-tensorflow/>

2. Keras 소개 및 특징

<https://blog.naver.com/sundooedu/221315683165>

3. Keras 이미지

<https://mparsec.com/wp-content/uploads/2017/07/anacondafeaturedimg.png>

4. Holdout method 이미지

<https://sebastianraschka.com/images/blog/2016/model-evaluation-selection-part1/testing.png>

<https://sebastianraschka.com/images/blog/2016/model-evaluation-selection-part2/holdout-validation.png>

5. 데이터 분할 이미지

<http://cfile22.uf.tistory.com/image/9951E5445AAE1BE0258820>

6. Feature Engineering 개념

[http://hero4earth.com/blog/learning/2018/01/29/Feature\\_Engineering\\_Basic/](http://hero4earth.com/blog/learning/2018/01/29/Feature_Engineering_Basic/)

7. Feature Scaling 이미지

[https://cdn-images-1.medium.com/max/1600/1\\*-9SPkqHA12dkiDCAHnXLuw.png](https://cdn-images-1.medium.com/max/1600/1*-9SPkqHA12dkiDCAHnXLuw.png)