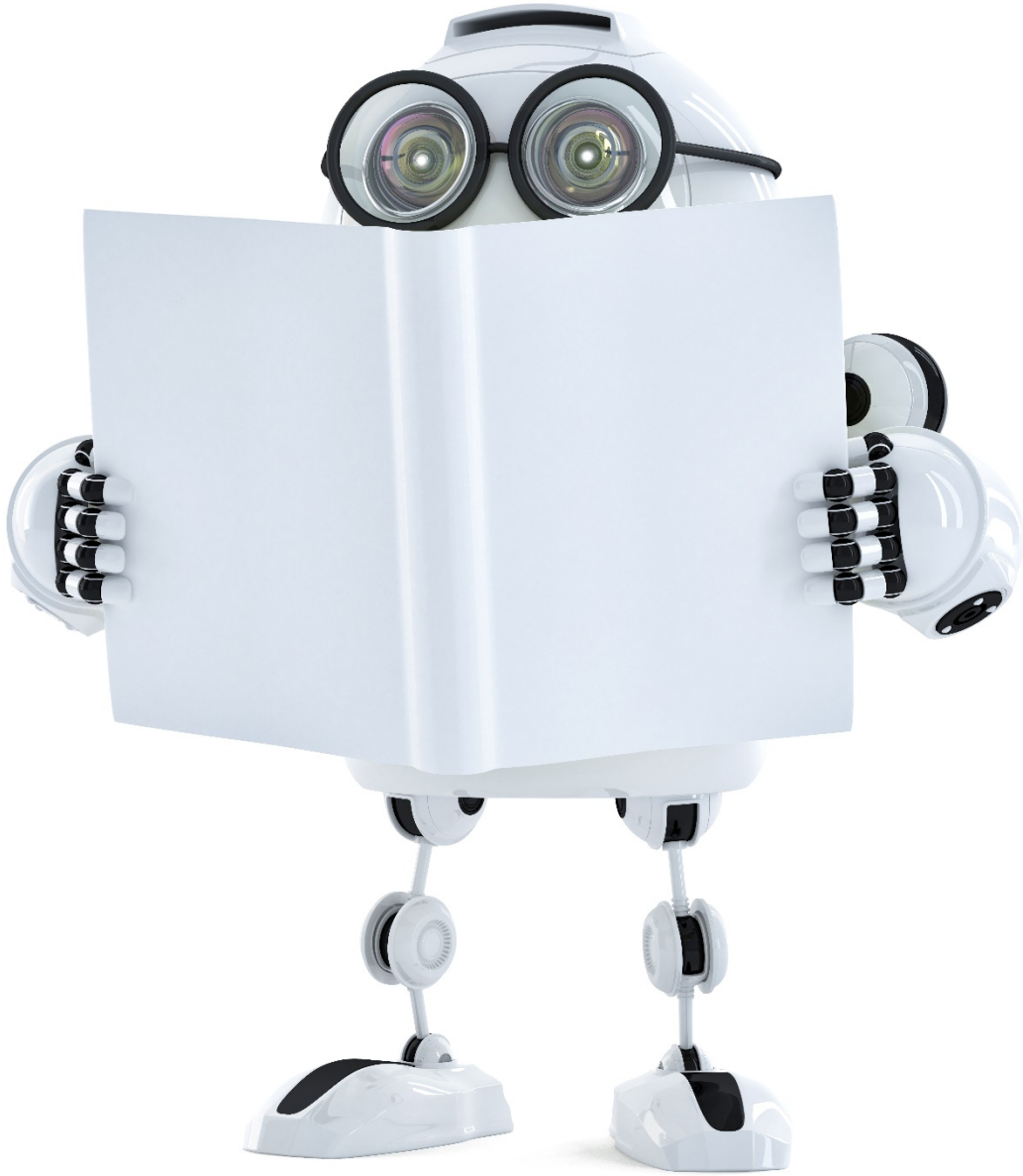
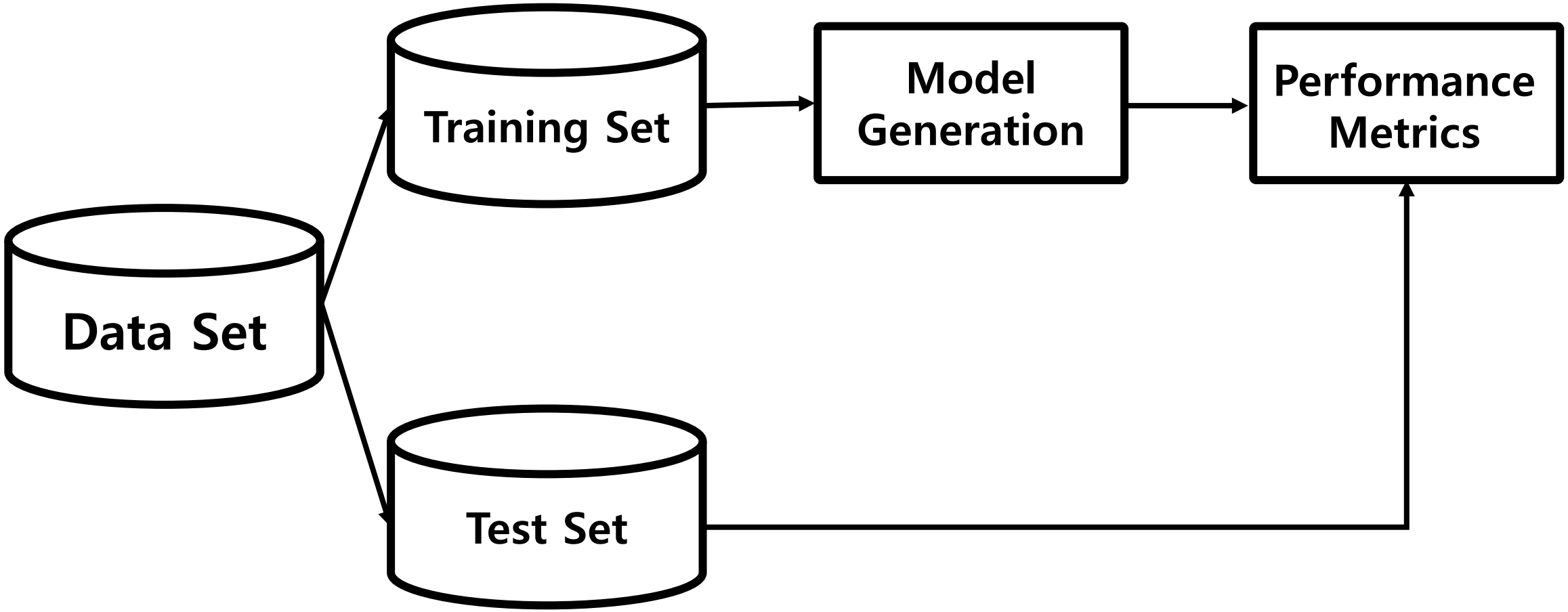
Performance measure techniques

Linear Regression



Director of TEAMLAB
Sungchul Choi

General ML Process
Training / Test Set


Data Set
Training Set
Test Set
Model Generation
Performance Metrics

Holdout Method (Sampling)

- 데이터를 Training과 Test와 나눠서 모델을 생성하고 테스트하는 기법
- 가장 일반적인 모델 생성을 위한 데이터 램덤 샘플링 기법
- Training과 Test를 나누는 비율은 데이터의 크기에 따라 다름

```
import numpy as np
from sklearn.model_selection import train_test_split

X, y = np.arange(10).reshape((5, 2)), range(5)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42)
```

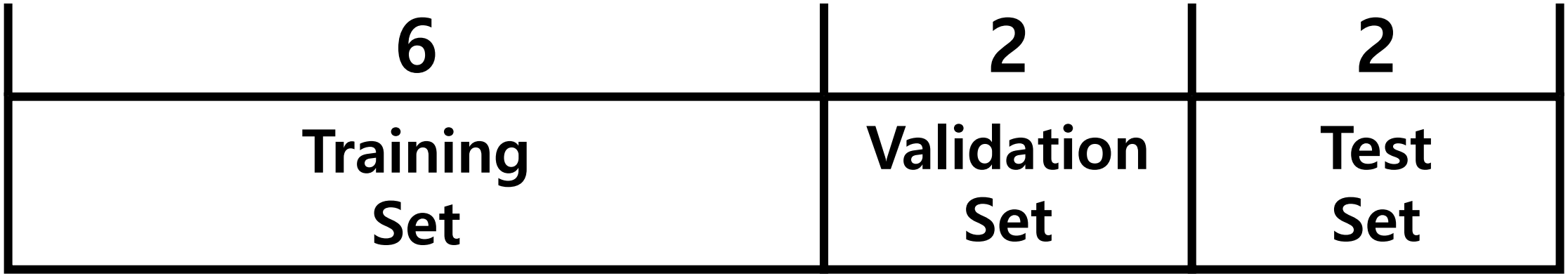
성능 측정을 위해 데이터를 나누는 방법

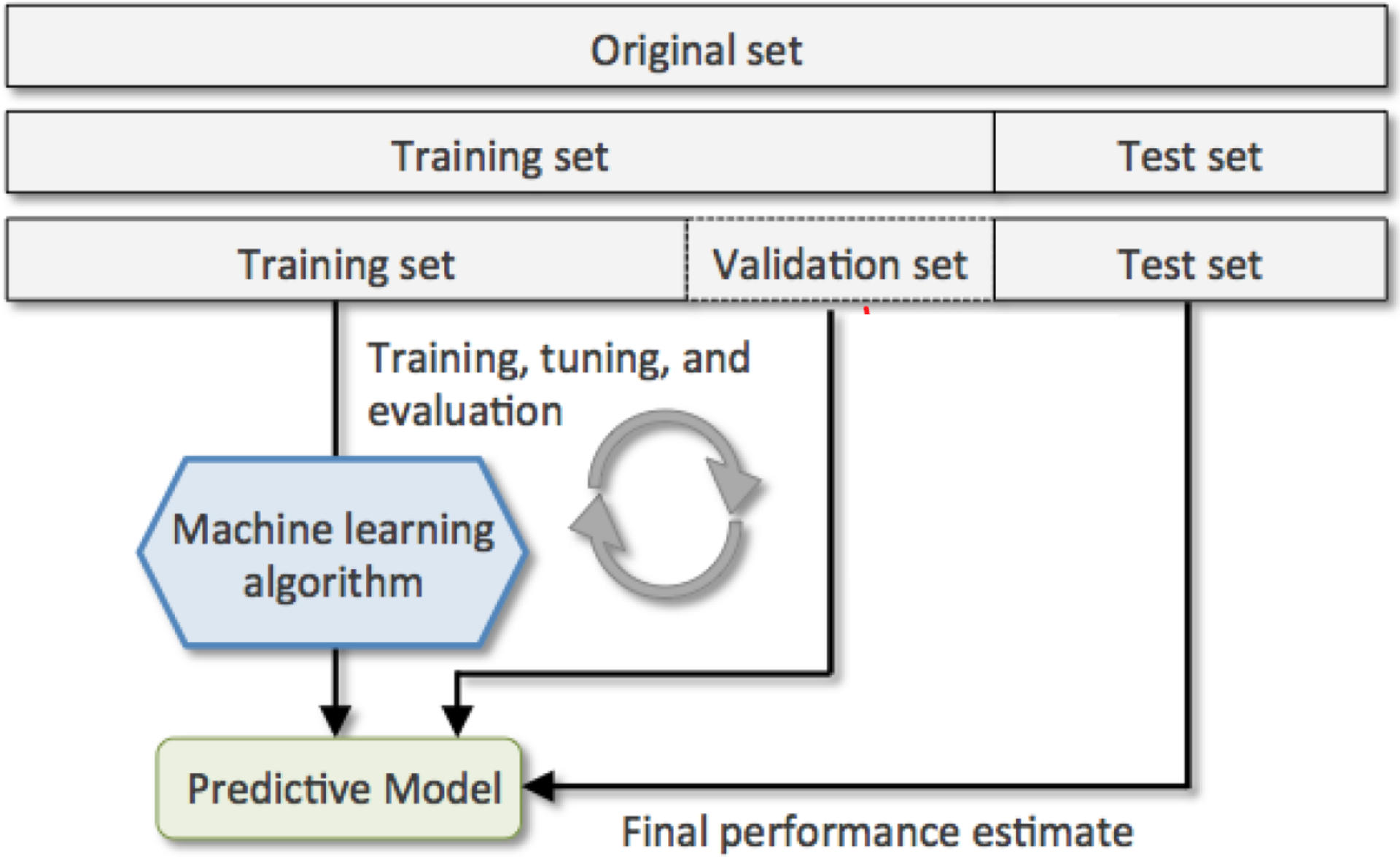
Training - Validation - Test

Training	Validation	Test
Model Building	Model Check	Model Evaluation

Validation Set

- Test Set은 Model이 생성시 절대 Training Set에 포함되지 않아야 함
- Test Set과 달리 Model 생성시 Model에 성능을 평가하기 위해 사용
- Hyper Parameter Turning 시 성능 평가를 통해 Overfitting 방지
- Training 중간에 Model의 성능을 점검

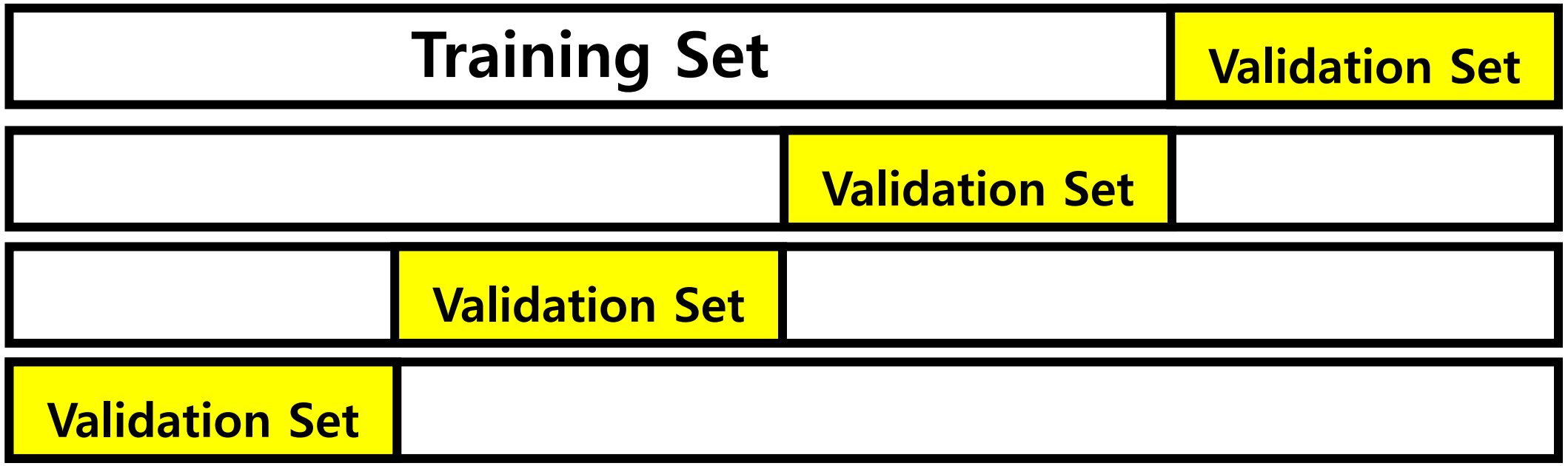

6
2
2
Training Set
Validation Set
Test Set



From: Python Machine Learning, https://goo.gl/JR9vxM

K-fold cross validation

- 학습 데이터를 K번 나눠서 Test와 Train을 실시 → Test의 평균값을 사용
- 모델의 Parameter 튜닝, 간단한 모델의 최종 성능 측정 등 사용



K-fold Cross Validation

```python
from sklearn.model_selection import KFold

kf = KFold(n_splits=10, shuffle=True)

for train_index, test_index in kf.split(X):
    print("TRAIN - ", train_index[:10])
    print("TEST - ", test_index[:10])
```

```
TRAIN -  [0 1 2 3 4 5 6 7 8 9]
TEST -  [ 16  22  24  25  28  58  60  79  92 110]
TRAIN -  [0 1 2 3 4 5 6 7 8 9]
TEST -  [ 23  30  33  56  66  69  72  73  74 107]
TRAIN -  [ 0  1  2  3  4  5  6  7  9 10]
TEST -  [  8  12  39  41  61  78  96  97 100 112]
TRAIN -  [ 0  1  2  3  4  6  7  8  9 10]
TEST -  [  5  15  31  38  46  85  91  95 116 124]
TRAIN -  [0 1 2 3 4 5 6 7 8 9]
TEST -  [ 18  37  40  43  55  57  75  77  90 104]
```

K-fold Cross Validation

```
from sklearn.model_selection import cross_validate
from sklearn.linear_model import Lasso, Ridge
from sklearn.metrics import mean_squared_error
kf = KFold(n_splits=10, shuffle=True)
lasso_regressor = Lasso(warm_start=False)
ridge_regressor = Ridge()

lasso_mse = []
ridge_mse = []

for train_index, test_index in kf.split(X):
    lasso_regressor.fit(X[train_index], y[train_index])
    ridge_regressor.fit(X[train_index], y[train_index])

    lasso_mse.append(mean_squared_error(y[test_index], lasso_regressor.predict(X[test_index])))
    ridge_mse.append(mean_squared_error(y[test_index], ridge_regressor.predict(X[test_index])))
```

```
sum(lasso_mse) / 10, sum(ridge_mse) / 10
```

```
(28.232782613547545, 23.611435666742835)
```

K-fold Cross Validation

- cross_val_score 함수로, 한번에 해결 가능
- sklearn은 pipeline 등을 위해 “High is better”로 처리
  → MSE를 Negative로 변환
- 이로인해 RMSE를 지원하지 않음

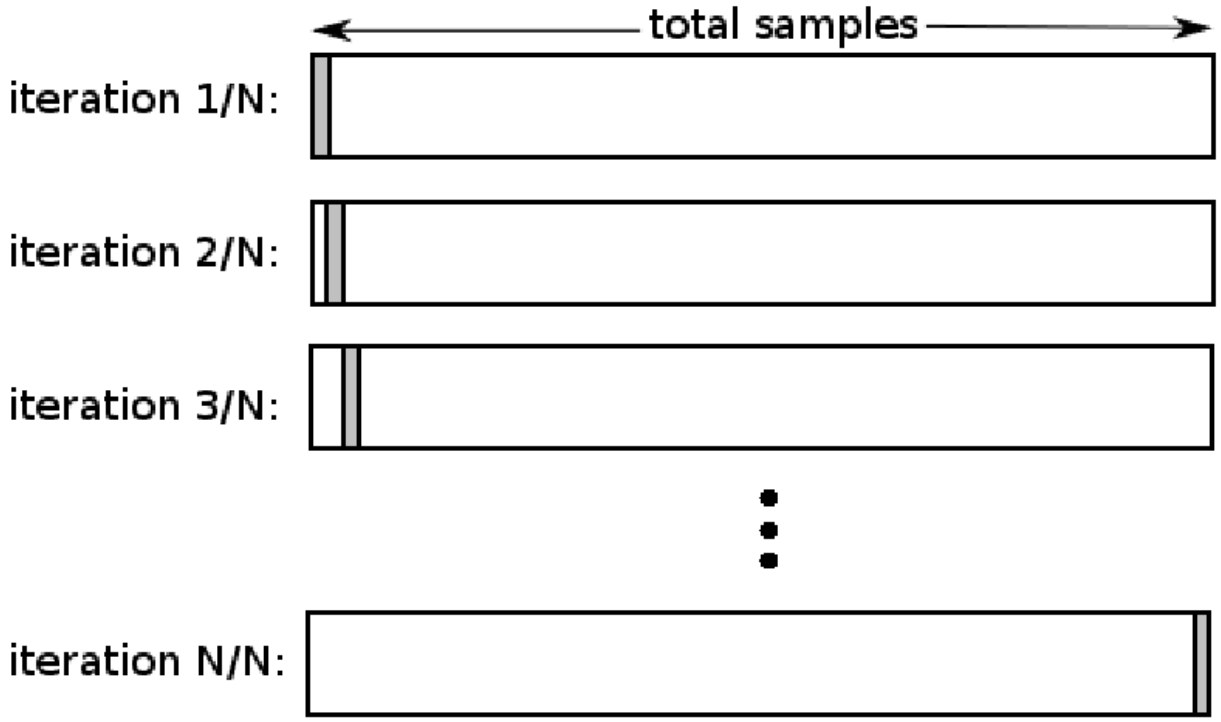
```
from sklearn.model_selection import cross_val_score
import numpy as np

lasso_regressor = Lasso(warm_start=False)
ridge_regressor = Ridge()

lasso_scores = cross_val_score(lasso_regressor, X, y, cv=10, scoring='neg_mean_squared_error')
ridge_scores= cross_val_score(ridge_regressor, X, y, cv=10, scoring='neg_mean_squared_error')
np.mean(lasso_scores), np.mean(ridge_scores)
```

```
(-34.468098837801122, -34.135235282917357)
```

https://github.com/scikit-learn/scikit-learn/issues/2439

Leave One Out (LOO)

- Simple cross validation → k = data size
- 한번에 한 개의 데이터만 Test set으로 사용함 → 총 k번 iteration



https://www.researchgate.net/profile/Nikolaos_Chlis/publication/266617511/figure/fig11/AS:295705362092036@1447513060277/Leave-One-Out-Cross-Validation.png

Leave One Out (LOO)

```
from sklearn.model_selection import LeaveOneOut

test = [1, 2, 3, 4]
loo = LeaveOneOut()
for train, test in loo.split(test):
    print("%s %s" % (train, test))
```

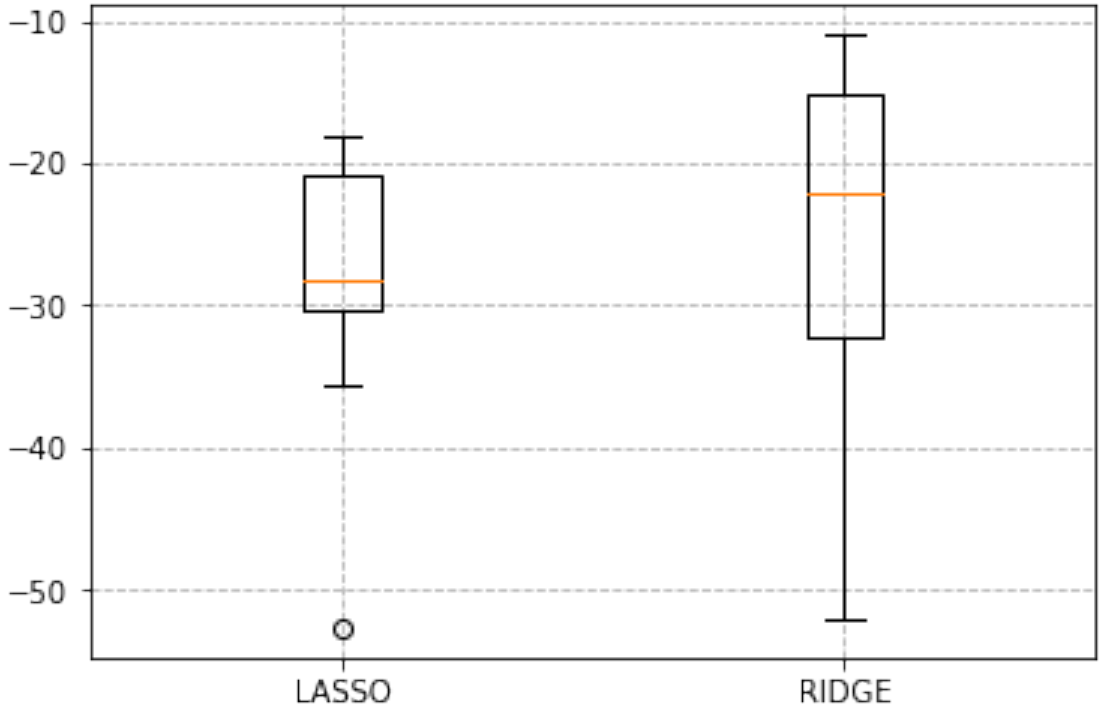
```
[1 2 3] [0]
[0 2 3] [1]
[0 1 3] [2]
[0 1 2] [3]
```

Leave One Out (LOO)

```
loo = LeaveOneOut()

lasso_scores = cross_val_score(lasso_regressor, X, y, cv=loo, scoring='neg_mean_squared_error')
ridge_scores= cross_val_score(ridge_regressor, X, y, cv=loo, scoring='neg_mean_squared_error')
np.mean(lasso_scores), np.mean(ridge_scores)
```

```
(-28.411385916387573, -23.867078861847261)
```

```
import matplotlib.pyplot as plt
labels=["LASSO", "RIDGE"]
plt.boxplot((lasso_scores, ridge_scores), labels=labels)
plt.grid(linestyle="--")
plt.show()
```



Check variation of cross validation

Validation set for parameter turning

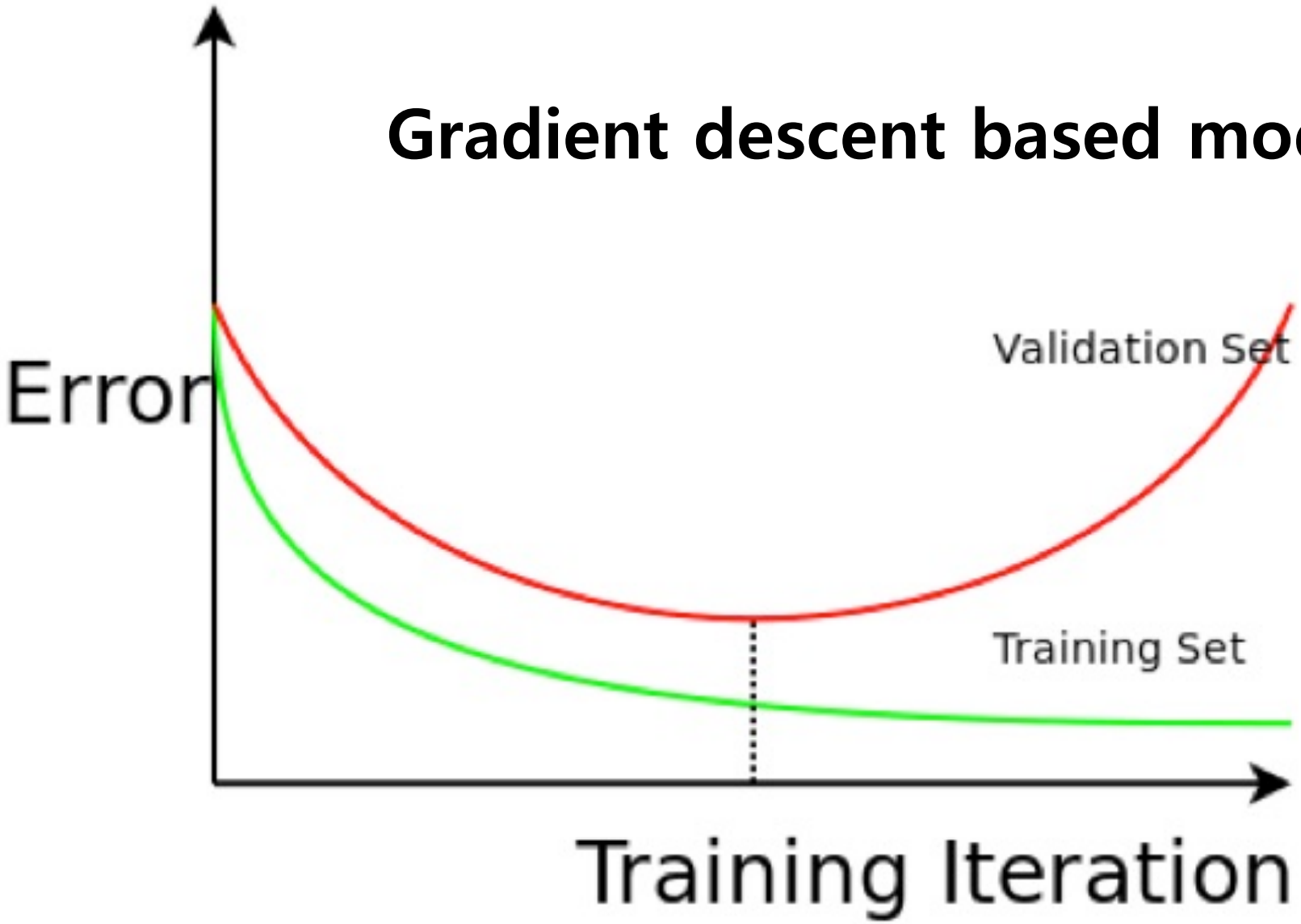
- Validation set의 많은 이유중 하나가 Hyper parameter turning

- Number of iterations (SGD), Number of branch (Tree-based) etc.

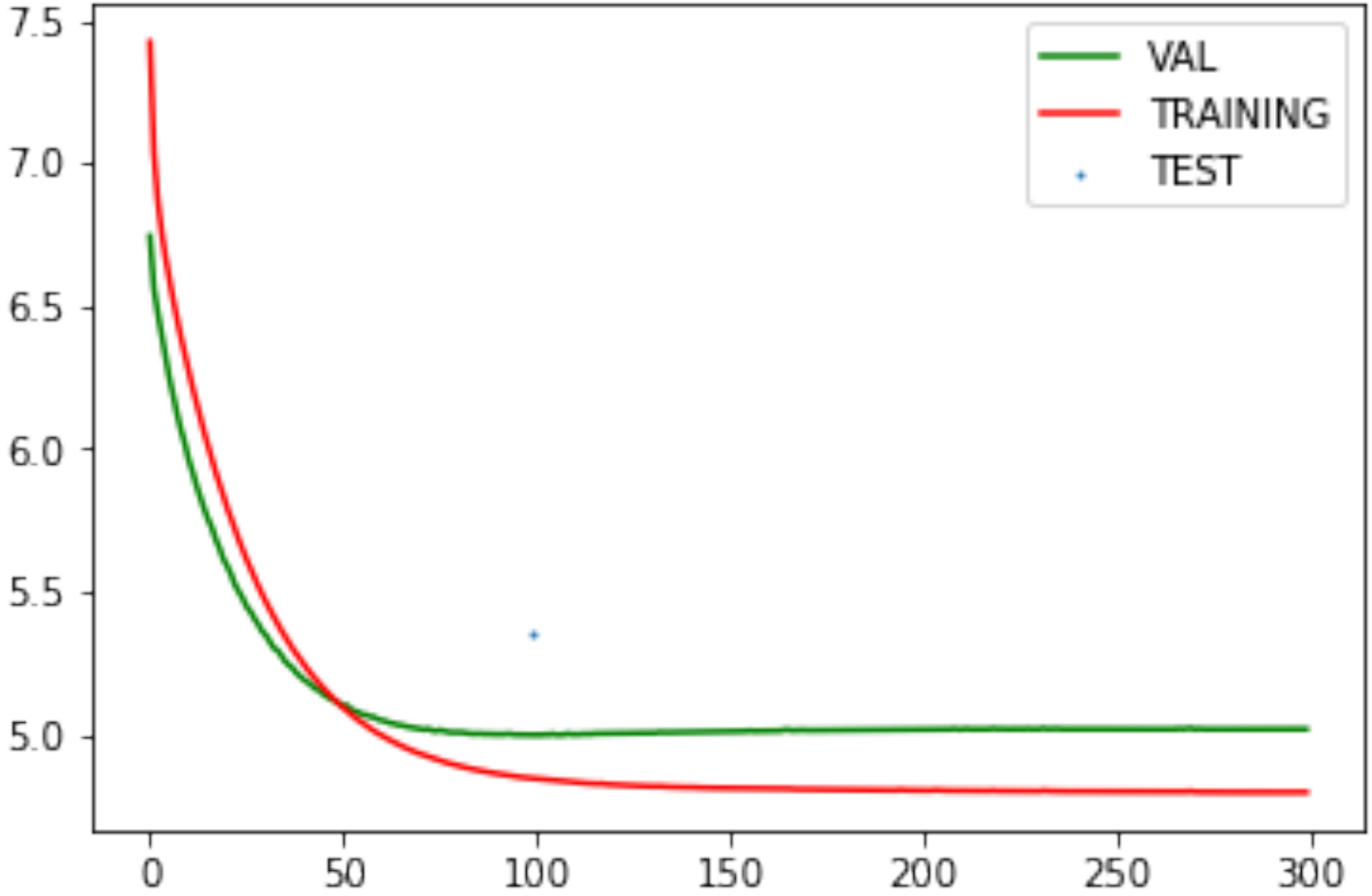
- Validation set의 성능으로 최적의 parameter를 찾음

- Validation set 결과와 Training set 결과의 차이가 벌어지면 overfitting

Gradient descent based model



Validation set for parameter turning



Etc…

- RepatedKFold – 중복이 포함된 K-Fold 생성

- LeavePOut – 한번에 P개를 뽑음 (Not LOO for one data)

- ShuffleSplit – 독립적인(중복되는) 데이터 Sampling

- StratifiedKFold – Y 값 비율에 따라 뽑음

- GroupKFold – 그룹별로 데이터를 Sampling

Cross validation
Train-Validation-Test

Human knowledge belongs to the world.