Introduction to Statistical Learning

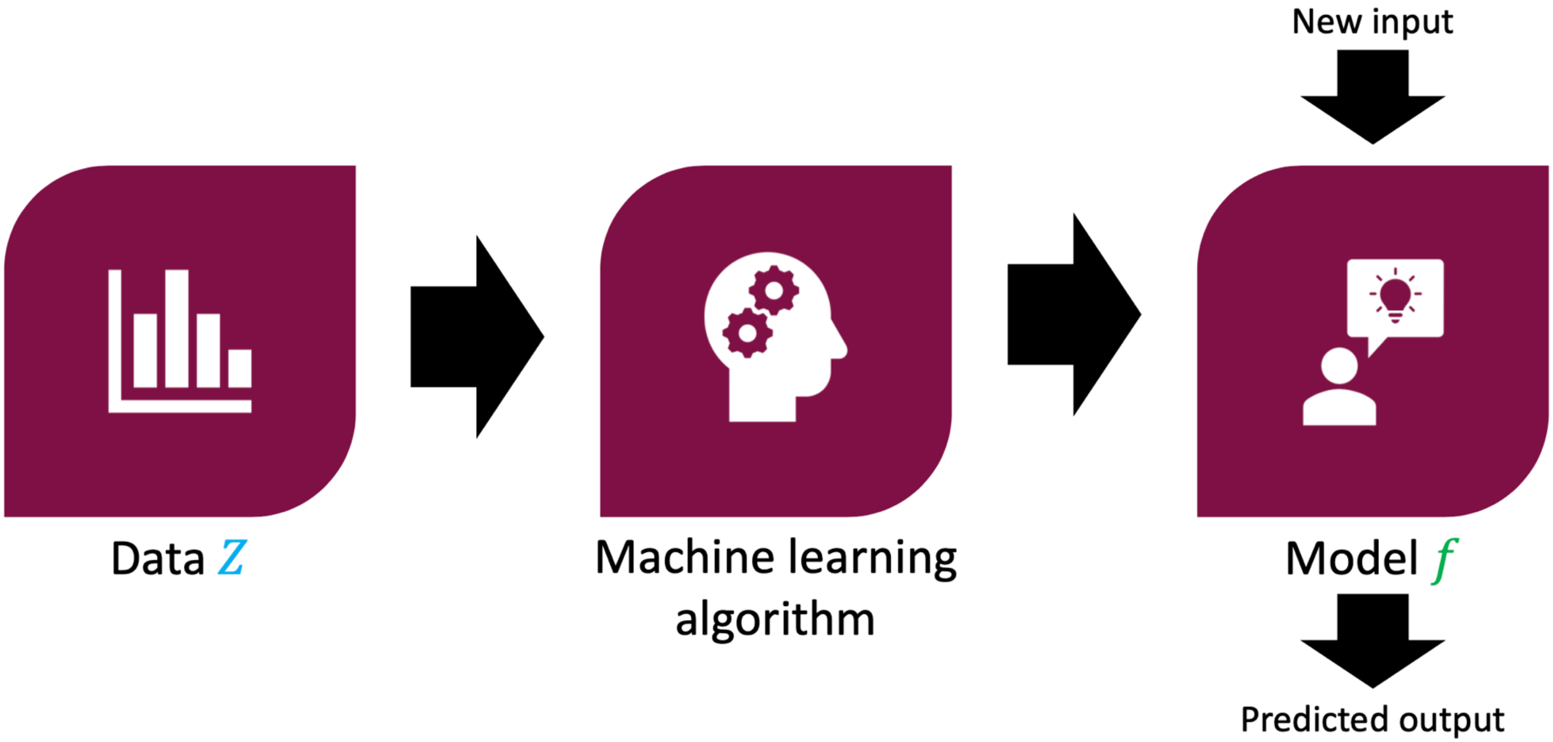
Summary of the First Module

송 준

고려대학교
통계학과 / 융합데이터과학 대학원

What’s Machine Learning?

What is Machine Learning?



Data Z

Machine learning algorithm

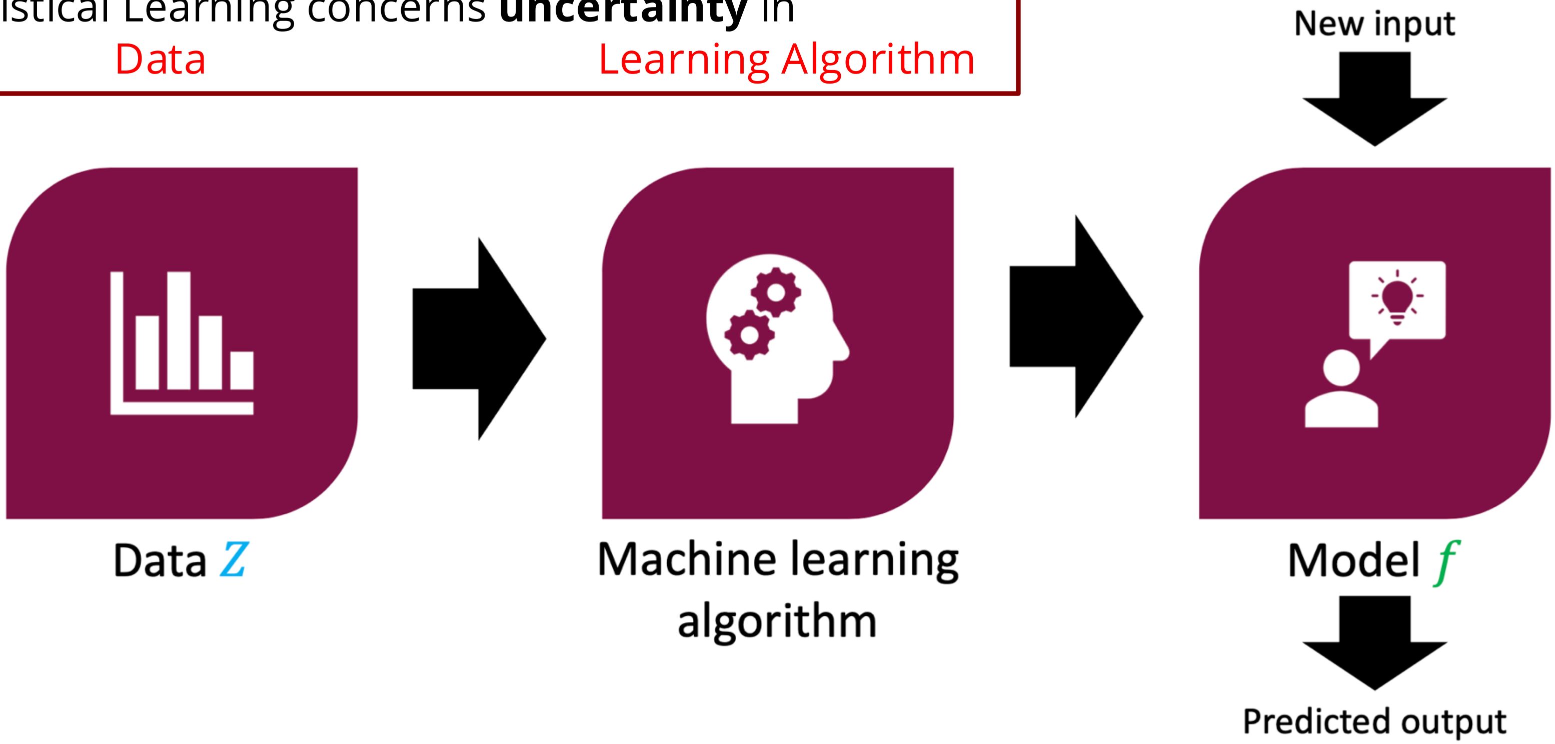
New input

Model f

Predicted output

What is Machine Learning?

Statistical Learning concerns **uncertainty** in
Data Learning Algorithm



New input

Data Z

Machine learning algorithm

Model f

Predicted output

많은 ML/AI 방법론들은 확률 기반의 통계학적 방법론을 근간으로 개발

Types of Learning

Types of Learning

- **Supervised learning**
    - **Input:** Examples of inputs (x) and outputs (y)
    - **Output:** Model that predicts unknown output given a new input

- **Unsupervised learning**
    - **Input:** Examples of some data (x) (output is not specified)
    - **Output:** Representation of structure in the data and further

Types of Learning

- **Supervised learning (with responses or labels (y))**
  - Regression, classification
- **Unsupervised learning (without responses or labels (y))**
  - Density estimation, clustering, dimension reduction

Foundational problem

Types of Learning

- **Supervised learning (with responses or labels (y))**
  - Regression, classification
- **Unsupervised learning (without responses or labels (y))**
  - Density estimation, clustering, dimension reduction

Foundational problem

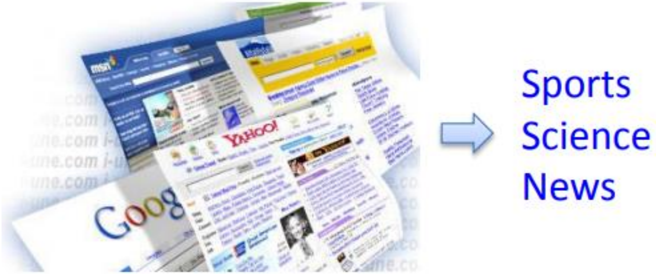
As SL/ML have become highly developed and more sophisticated, more problems have arisen in a variety of scenarios

- Reinforcement learning (interactive, maximizing reward)
- Semi-supervised learning (y's are partially observed)
- Self-supervised learning (no y, but give y manually)
- Active learning (interactive, machine-human)
- Online learning (incremental, update pre-fitted model (large) with a new data (small))
- Transfer learning (using pre-trained model in a new problem)
- Multitask learning (multi-task from one model)
- Federated learning (multi-source, privacy consideration)
- etc

Supervised Learning

Supervised Learning

Goal: Construct a predictor $f: X \mapsto Y$ that minimizes a risk $\boldsymbol{R(f)}$, **performance measure**



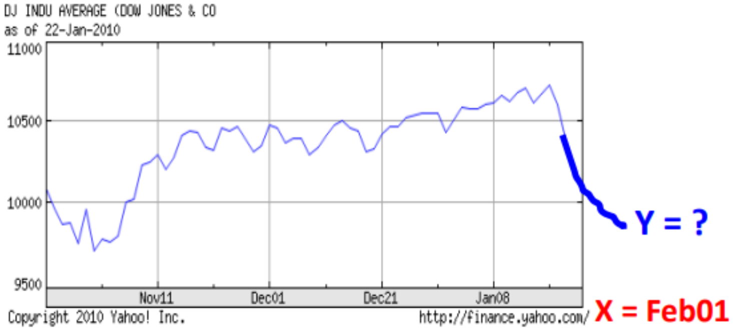
Sports

Science

News

✓ **Classification** output: a class

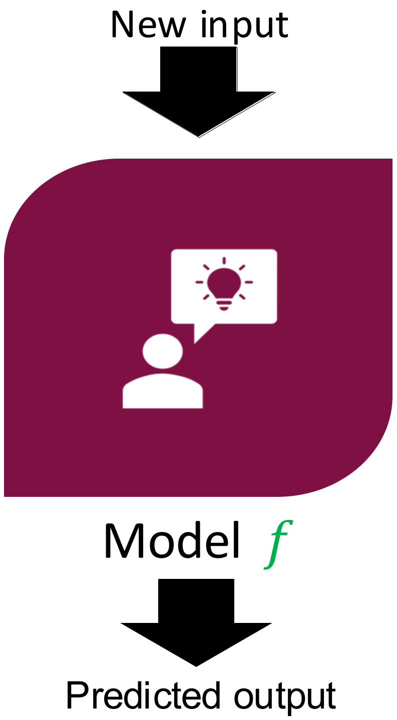
$$R(Y, f) = P\big(Y \neq f(X)\big)$$



✓ **Regression** output: a number

$$R(Y, f) = E\left[\left(Y - f(X)\right)^2\right]$$

Performance Measures : Loss



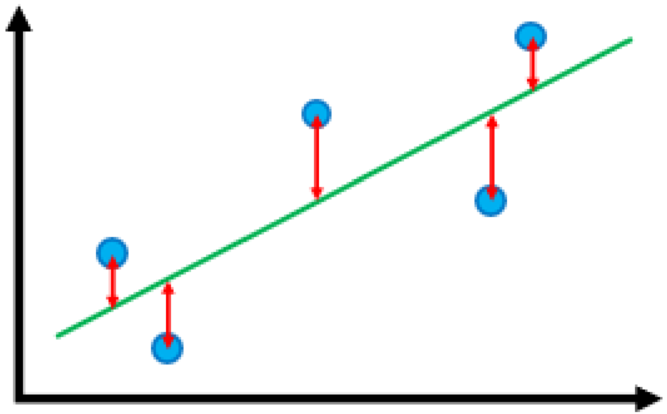
New input

Model f

Predicted output

Loss = loss(true value, predicted value)

e.g., loss$(y_i, f(x_i))$: i번째 관측값 pair 의 loss



Performance Measures : Risk

Performance:

- $loss(Y, f(X))$: Measure of closeness between true label Y and prediction f(X)
- We want to perform well on any test data : $(X, Y) \sim P_{XY}$
- Given an X drawn randomly from a distribution, how well does the predictor perform on average?

$$\boxed{\text{Risk } R(f) \equiv \mathbb{E}_{XY}[loss(Y, f(X))]}$$

Performance Measures

Performance of supervised learning:

$$\text{Risk } R(f) \equiv \mathbb{E}_{XY}[loss(Y, f(X))]$$

	Classification	Regression
$loss(Y, f(X))$	$\mathbb{1}_{\{f(X) \neq Y\}}$	$(f(X) - Y)^2$
Risk $R(f)$	$P(f(X) \neq Y)$	$\mathbb{E}[(f(X) - Y)^2]$

Performance : Are we done?

Ideal goal: Construct prediction rule $f^* : \mathcal{X} \rightarrow \mathcal{Y}$

$$\boldsymbol{f^* = argmin_f \mathbb{E}_{XY}[loss(Y, f(X))]}$$

Bayes optimal rule

Practical goal:

Given $\{(X_i, Y_i)\}_{i=1}^n$, **learn** prediction rule $\widehat{f_n} : \mathcal{X} \rightarrow \mathcal{Y}$

Often: $\widehat{f_n} = argmin_{f \in F} \frac{1}{n} \sum_{i=1}^n [loss\big(Y_i, f(X_i)\big)]$

Empirical Risk minimizer

$$\boxed{\frac{1}{n} \sum_{i=1}^n [loss\big(Y_i, f(X_i)\big)] \xrightarrow{L.L.N} \mathbb{E}_{XY}[loss\big(Y, f(X)\big)]}$$

Performance of Estimated Function

Optimal predictor:

$$f^* = argmin_f \mathbb{E}[(f(X) - Y)^2]$$

Empirical Risk Minimizer:

$$\widehat{f}_n = argmin_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} ((f(X_i) - Y_i))^2$$

Class of predictors Empirical mean

$\boldsymbol{\hat{f}_n}$: A function of observed data $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$

Training Error(Risk):

$$\mathbb{E}_n \left[\text{loss}\left(Y, \boldsymbol{\hat{f}_n}(X)\right) \right] = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \boldsymbol{\hat{f}_n}(X_i)\right)^2$$

Performance of Estimated Function

Optimal predictor:

$$f^* = argmin_f \mathbb{E}[(f(X) - Y)^2]$$

Empirical Risk Minimizer:

$$\widehat{f_n} = argmin_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} ((f(X_i) - Y_i\ ))^2$$

Class of predictors Empirical mean

Expected Risk(Generalization Error)

$$\mathbb{E}_{D_n}[R(\widehat{f_n})] \doteq \ \mathbb{E}_{D_n}[\mathbb{E}_{XY}[loss\left(Y, \widehat{f_n}(X)\right)]]$$

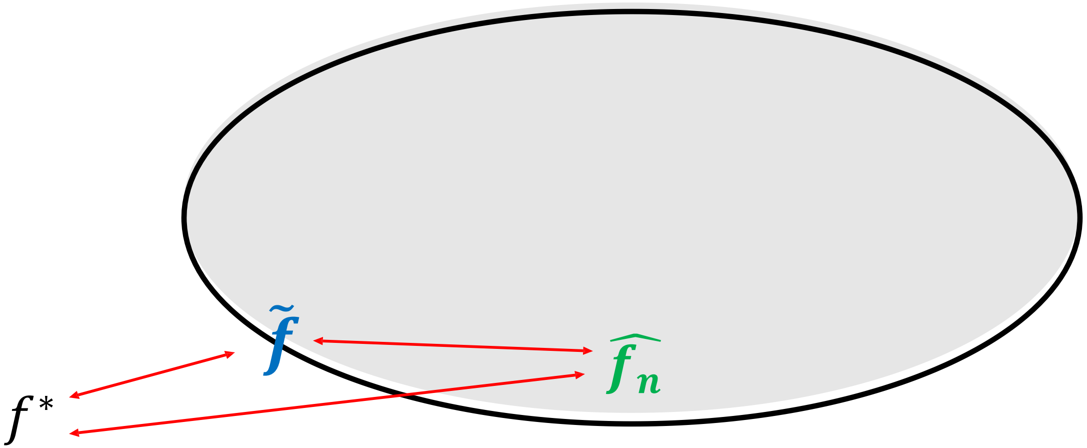
Function Estimation : limited in practice

Ideal goal: Construct prediction rule $f^*:\ \mathcal{X} \rightarrow \mathcal{Y}$

$$\boldsymbol{f^* = argmin_f \mathbb{E}_{XY}[loss(Y, f(X))]}$$

$$\tilde{\boldsymbol{f}} = \text{arg}\min_{f \in F} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]$$

$$\widehat{\boldsymbol{f_n}} = \text{arg}\min_{f \in F} \sum_{i=1}^{n} \text{loss}\big(Y_i, f(X_i)\big)$$



Linear Regression

Simplest Regression Method

Introduction

- **회귀 모형 (Regression Model)**:

$$\boxed{Y} = f(\boxed{X}) + \boxed{\epsilon}$$

오차 (error)
*주의: 뒤에 나오는 잔차(residual)와 다른 개념
- 불확정성(uncertainty), noise, etc

종속변수 (Dependent Variable) 독립변수 (Independent
Variable)
반응변수 (Response Variable) 설명변수 (Explanatory Variable)
반응변수 (Response Variable) 예측변수 (Predictor Variable)
Output Input

독립변수, 반응변수는 각각 확률변수로 여러 개의 확률변수가 있을 수 있음

Goal of Regression Models

- **회귀 모형 (Regression Model)**:

$$Y = f(X) + \epsilon$$

- **Goal of Regression Models:**
  - **추정 (Estimation):** 관계를 나타내는 함수 f에 대한 추정
  - **예측 (Prediction)**: X 값이 주어졌을 때 대응되는 Y 값의 예측
  - **추론 (Inference)**: Further investigation
    - 예측이 "얼마나" 정확한가?
    - 함수 f() 가 얼마나 정확한가?
    - 예측변수가 여러 개 있을 때 모든 변수가 Y의 값에 영향을 주나?
    - 모형이 충분히 적합 됐나?

Goal of Regression Models

- **회귀 모형 (Regression Model)**:

$$Y = f(X) + \epsilon$$

- **Goal of Regression Models:**
  - **추정 (Estimation):** 관계를 나타내는 함수 f에 대한 추정
  - **예측 (Prediction)**: X 값이 x로 주어졌을 때 Y 값의 예측
  - **추론 (Inference)**: Further investigation of the data
    - 예측이 “얼마나” 정확한가?
    - 함수 f() 가 얼마나 정확한가?
    - 예측변수가 여러 개 있을 때 모든 변수가 Y의 값에 영향을 주나?
    - 모형이 충분히 적합 됐나?
  - **예측**만 목표로 할 시: 다양한 방법론 적용 가능
  - **추론**을 목표로 할 시: 관계를 나타내는 f()에 제약이 필요함
  - 단순한 모형부터 시작! **f 는 선형함수.**

Linear Regression Models

- **회귀 모형 (Regression Model)**:

$$Y = f(X) + \epsilon$$

  - $X = (X_1, \cdots, X_p)$: p차원 확률변수
  - Y: 1차원 확률변수

- **선형 회귀 모형 (Linear Regression Model)**:

$$f\colon \mathbb{R}^p \to \mathbb{R}$$

  - f : X와 Y가 선형관계를 가진다

$$\Updownarrow$$

  - $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$ for some $\beta_j, j = 0, \ldots, p$

Estimation: Simplification

- **Goal:** Using the data (observations)
  - Estimate $f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
  - Estimate $\beta = \left(\beta_0, \beta_1, \ldots, \beta_p\right)^T$
- Sometimes, ignore β_0: X 값에 영향을 받지 않는 Y만의 평균값.
  - 편의상 $\beta_0 = 0$ 이라 가정하기도 함 (Y_i 대신 $Y_i - \beta_0$가 output이라고 생각), 혹은
  - input x 에 1이 고정적으로 있다고 가정. $\beta = \left(\beta_0, \beta_1, \ldots, \beta_p\right)^T$

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1p} & \cdots & x_{np} \end{pmatrix} \qquad y_i \approx \beta_0 \cdot 1 + \beta_1 \cdot x_{i1} + \cdots + \beta_p \cdot x_{ip}$$

Linear Functions

Linear Functions

- Consider the space of linear functions $f_\beta(x)$ defined by

$$f_\beta(x) = \beta^T x = \begin{bmatrix} \beta_1 & \cdots & \beta_p \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix} = \beta_1 x_1 + \cdots + \beta_p x_p$$

- $x \in \mathbb{R}^p$ is called an **input** (a.k.a. **features** or **covariates**)
- $\beta \in \mathbb{R}^p$ is called the **parameters** (a.k.a. **parameter vector**)
- $y = f_\beta(x)$ is called the **label** (a.k.a. **output** or **response**)

Choice of Loss Function

f_β 가 주어졌을 때 i번째 관측치에 대한 loss

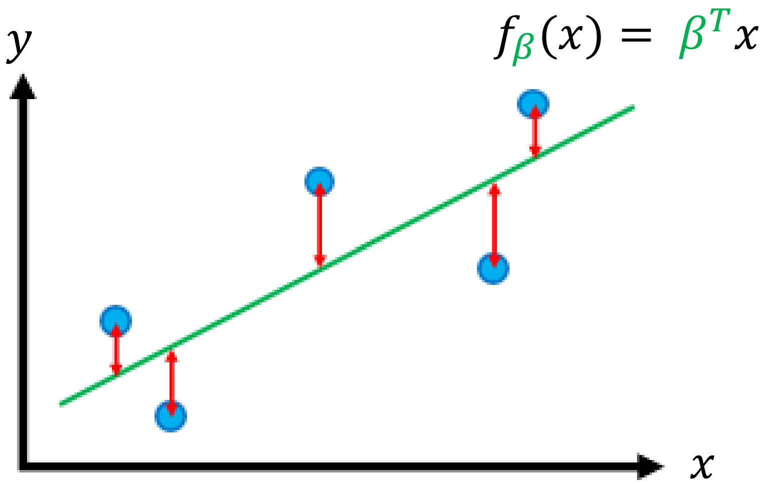
Choice of Loss Function

- $y_i \approx \beta^T x_i$ if $(y_i - \beta^T x_i)^2$ small

- **Mean squared error(MSE) :**

$$\hat{R}(\beta; Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^T x_i)^2$$

- Computationally convenient and works well in practice



$f_\beta(x) = \beta^T x$

$$\hat{R}(\beta; Z) = \frac{\updownarrow^2 + \updownarrow^2 + \updownarrow^2 + \updownarrow^2 + \updownarrow^2}{n}$$

Choice of Loss Function

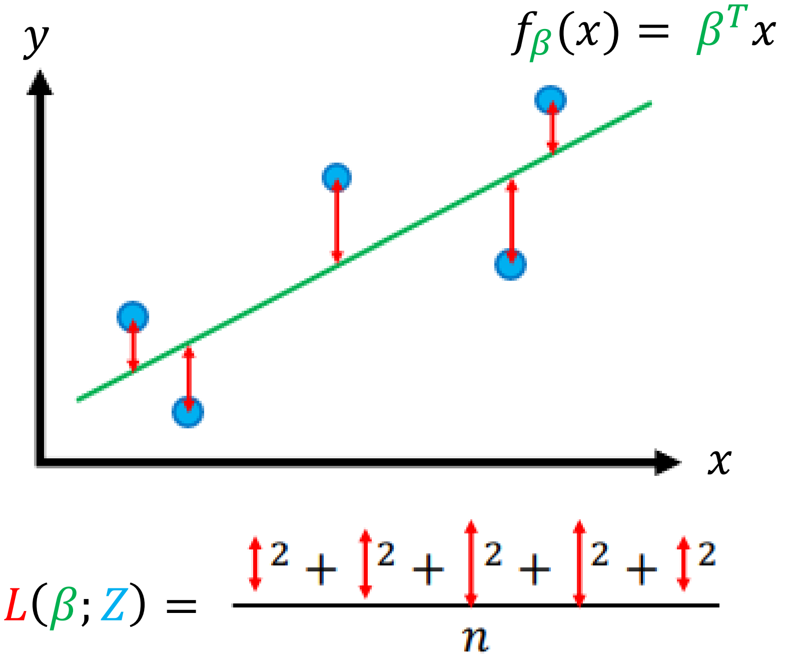
f_β 가 주어졌을 때 i번째 관측치에 대한 loss (squared error loss)

Choice of Loss Function

- $y_i \approx \beta^T x_i$ if $(y_i - \beta^T x_i)^2$ small
- **Mean squared error(MSE) :**

$$L(\beta; Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^T x_i)^2$$

- 편의상 위 값을 loss 라고 표현하기도 한다.



Linear Regression Algorithm

- **Input** : Dataset $Z = \{(x_1, y_1), \cdots, (x_n, y_n)\}$
- Compute

$$\hat{\beta}(Z) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} L(\beta; Z)$$

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2$$

최소제곱추정(LSE):
minimizing
squared error loss

- **Output** : $f_{\hat{\beta}(Z)}(x) = \hat{\beta}(Z)^T x$
- $\hat{\beta}$ 은 다음 식의 ***solution***

$$(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})\beta = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{Y}$$

예측 (Prediction)

- **Prediction:** $X = x$ 값이 주어질 때 이와 대응되는 Y의 값 예측

- **Idea**: 조건부 **기댓값(평균)**: $X_i = x_i$ 라고 값이 주어졌을 경우, (x_i는 상수)

$$Model\colon\ Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \epsilon_i$$

$$E(Y_i | X_i = x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

- $X_i = x_i$ 라고 값이 주어졌을 경우 가능한 Y의 값 중 **평균**으로 예측.

- **평균으로의 회귀(Regression)**

예측 (Prediction)

- **Idea**: 조건부 **기대값**: $X_i = x_i$ 라고 값이 주어졌을 경우, (x_i는 상수)

$$Model:\ Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \epsilon_i$$

$$E(Y_i|X_i = x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

조건부 기대값을 취하면 평균이 0인 오차항 제거

- **예측 (Prediction):** 적절한 추정값 $\hat{\beta}_j, j = 0, \dots, p$ 을 구했다면,

새로운 $X = x^* = \left(x_1^*, \dots, x_p^*\right)$값에 대응되는 Y의 예측은 조건부 기대값

$$\hat{y} = E(Y\widehat{|X =} x^*) = \hat{\beta}_0 + \hat{\beta}_1 x_1^* + \hat{\beta}_2 x_2^* + \cdots + \hat{\beta}_p x_p^*$$

- **적합 값 (Fitted values):** 이미 관측된 $x_i, i = 1, \dots, n$ 에 대응 하는 y값

$$\widehat{y_i} = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip}, \qquad i = 1, \dots, n$$

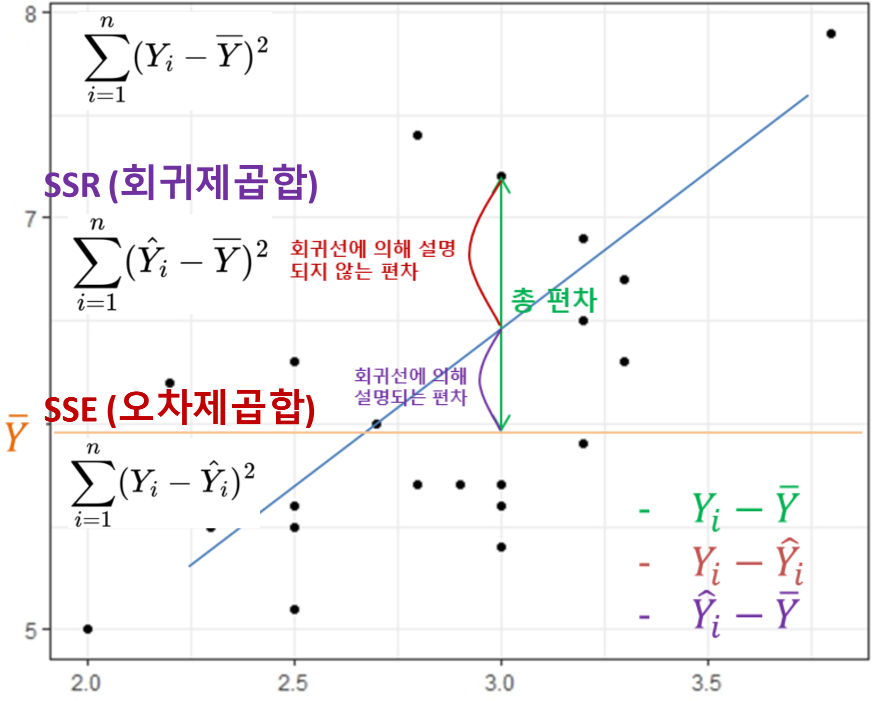
추론

- X들이 Y 에 어떻게 영향을 미치는가? 각 변수별로 Positive? Negative? 얼마나?
- 해당 데이터가 Linear Regression 하는게 적합한가?
- Y 와 관계가 있는 X 변수들이 모두 다 모델에 필요한 변수들인가?
- Linear regression 결과가 믿을 만 한가?

추론: 결정계수 R^2

- SST (총 편차제곱합)

$$\sum_{i=1}^{n}(Y_i - \overline{Y})^2$$



- SSR (회귀제곱합)

$$\sum_{i=1}^{n}(\hat{Y}_i - \overline{Y})^2$$

- SSE (오차제곱합)

$$\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

결정계수$\boldsymbol{R^2}$(coefficient of determination)
회귀직선의 적합도를 평가하는 방법
전체변동에서 회귀로 설명되는 부분이 차지하는 비율

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$\sum_{i=1}^{n}(Y_i - \overline{Y})^2 = \sum_{i=1}^{n}(\hat{Y}_i - \overline{Y})^2 + \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

SST(var) = SSR + SSE (training loss)

추론: Statistical Hypothesis Test

- (모델전체)모델이 유의한가? F-Test: 기각 시 (p-value<0.05)
  - $\beta_1, \ldots, \beta_p$ 중 어느 하나라도 0이 아닌 값이 있다

- (개별변수) 독립변수별 유의성 검정: $\mathrm{H}_0\!: \beta_1 = 0, \ldots,\ H_0\!: \beta_p = 0$ 각각에 대해 시행
  - T-Test or Z-Test: 기각시 (p-value<0.05) β_j가 0이 아닌 충분한 근거 획득
  - X_j는 Y 와 선형관계가 있다

추론: 다중공선성(Multicollinearity) 확인

- X 변수들이 서로 표본 상관관계가 1, or p>n (high-dimensional problem)
    - $(\boldsymbol{X}^T\boldsymbol{X})\hat{\beta} = \boldsymbol{X}^T\boldsymbol{Y}$: 유일한 해가 존재하지 않음.

- X 변수들이 서로 상관관계가 매우 높다면
    - $(\boldsymbol{X}^T\boldsymbol{X})$ 의 determinant 가 0에 가까움.
    - $\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ <- 이 값이 매우 불안정함 (분산이 매우 높음!)
    - 계산이 가능하더라도 결과값에 대한 신뢰도는 매우 낮게 됨

- 통계모델 학습 자체는 가능. 하지만 결과값이 j번째 변수에 의한 것인지 아니면 다른 변수에 의한 것인지 판단이 어려움. 해석에 유의.

Feature Mapping

Feature Maps

General strategy

- Model family F = $\{f_\beta\}_\beta$
- Loss function $L(\beta; Z)$

Linear regression with feature map

- Linear functions over a given **feature map** $\phi: X \rightarrow \mathbb{R}^d$

$$F = \{f_\beta(x) = \beta^T \phi(x)\}$$

- MSE $L(\beta; Z) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta^T \phi(x_i))^2$

Quadratic Feature Map

- Consider the feature map $\phi: \mathbb{R} \to \mathbb{R}^2$ given by

$$\phi(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$$

- Then, the model family is

$$f_\beta(x) = \beta_1 x + \beta_2 x^2$$

Examples of Feature Maps

- **Feature Mapping Techinique**
  - **Input X를 X의 nonlinear 함수공간으로 보냄.** $\phi(x)$
  - **Y와** $\phi(x)$ **간의 linear 방법론 Fitting**
- **Polynomial features**
  - $f_\beta(x) = \beta_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_1^2 + \beta_5 x_1 x_2 + \beta_6 x_2^2 + \cdots$
  - Quadratic features are very common; capture “feature interactions”
  - Can use other nonlinearities (exponential, logarithm, square root, etc.)
- **Basis expansion approach**
  - $f_\beta(x) = \beta_0 + \beta_1 \phi_1(x) + \cdots + \beta_d \phi_d(x)$
  - Fit the data in a more general way

Feature Mapping for Qualitative Predictor

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

$$y_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female,} \\ \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

- β_1: difference of E(Y|X) male v.s. female

More than two levels

- With more than two levels for each variable, we create additional **dummy variables.**

- $Y \sim X$
  - Y: credit card balance
  - X: ethnicities (Asian, Caucasian, African American)

More than two levels

- $Y \sim X$
  - Y: credit card balance
  - X: ethnicities (Asian, Caucasian, African American)

$$x_{i1} = \begin{cases} 1 & if\ ith\ person\ is\ Asian, \\ 0 & if\ ith\ person\ is\ not\ Asian. \end{cases}$$

And the second could be

$$x_{i2} = \begin{cases} 1 & if\ ith\ person\ is\ Caucasian, \\ 0 & if\ ith\ person\ is\ notCaucasian. \end{cases}$$

More than two levels

- $Y \sim X$
  - Y: credit card balance
  - X: ethnicities (Asian, Caucasian, African American)

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$$

$$y_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & if\ ith\ person\ is\ Asian, \\ \beta_0 + \beta_2 + \varepsilon_i & if\ ith\ person\ is\ Caucasian, \\ \beta_0 + \varepsilon_i & if\ ith\ person\ is\ African\ American. \end{cases}$$

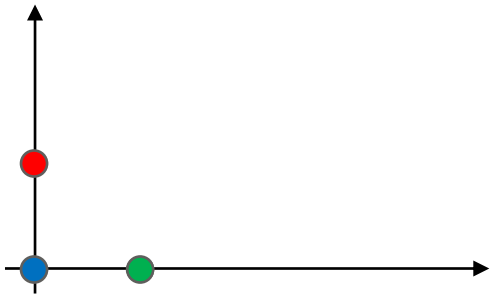
- **Baseline** category: African American
- β_1: difference of E(Y|X) between African American and Asian
- β_2: difference of E(Y|X) between African American and Caucasian

Dummy v.s. One-Hot Encoding

Color
Red
Green
Blue

Dummy encoding

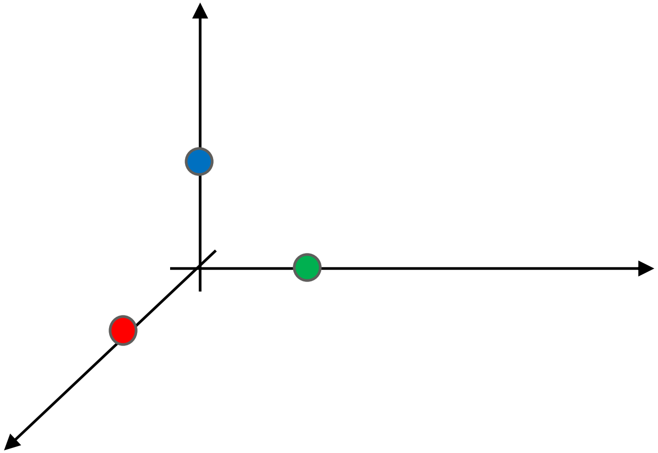
d1	d2
1	0
0	1
0	0



Color
Red
Green
Blue

One-hot encoding

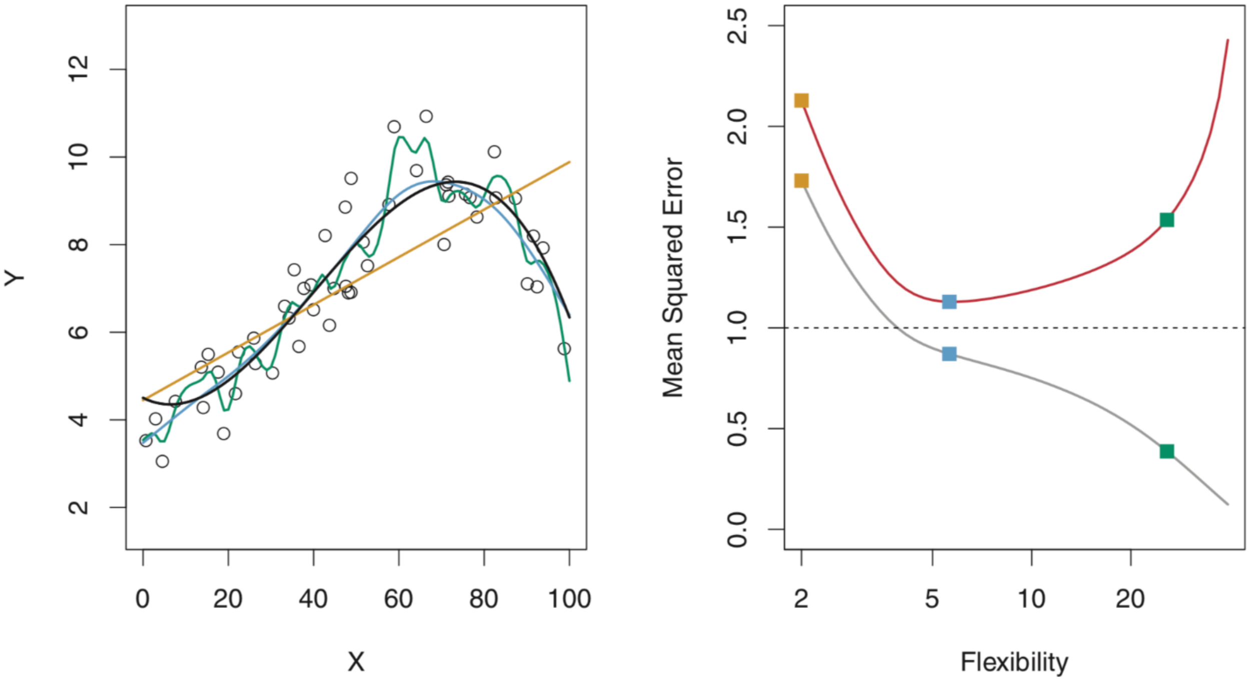
d1	d2	d3
1	0	0
0	1	0
0	0	1



Feature Selection & Regularization

Training Loss (MSE) v.s. Test Loss (MSE)

- Training MSE (grey) decreases monotonically as the model flexibility increases and Test MSE (red) has U-shape



Bias-Variance Tradeoff

- Increasing number of examples n in the data…
  - Tends to **increase bias** and **decrease variance**

- **General strategy**
  - **High bias:** Increase model capacity d
  - **High variance:** Increase data size n (i.e., gather more labeled data)

Bias-Variance Tradeoff

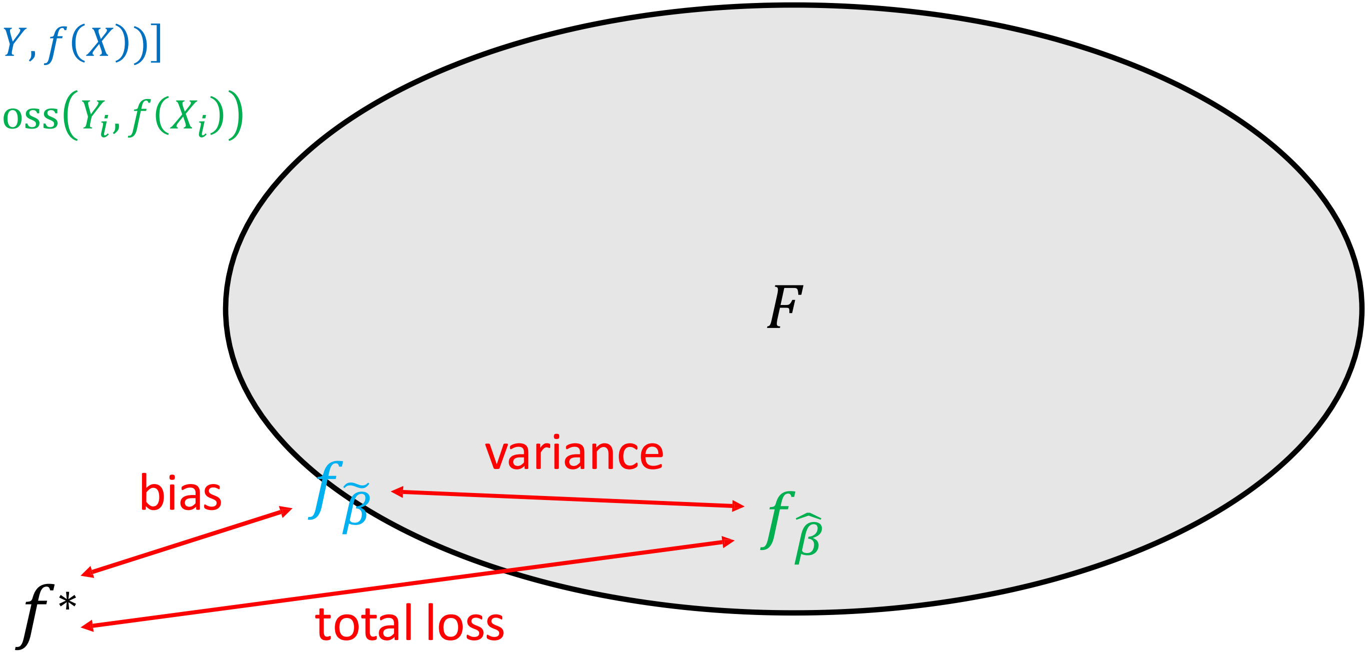
Ideal goal: Construct prediction rule $f^* : \ \mathcal{X} \rightarrow \mathcal{Y}$

$$\boldsymbol{f^* = argmin_f \mathbb{E}_{XY}[loss(Y, f(X))]}$$

$$\tilde{f} = \arg\min_{f \in F} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]$$

$$\widehat{\boldsymbol{f_n}} = \arg\min_{f \in F} \sum_{i=1}^{n} \text{loss}\big(Y_i, f(X_i)\big)$$

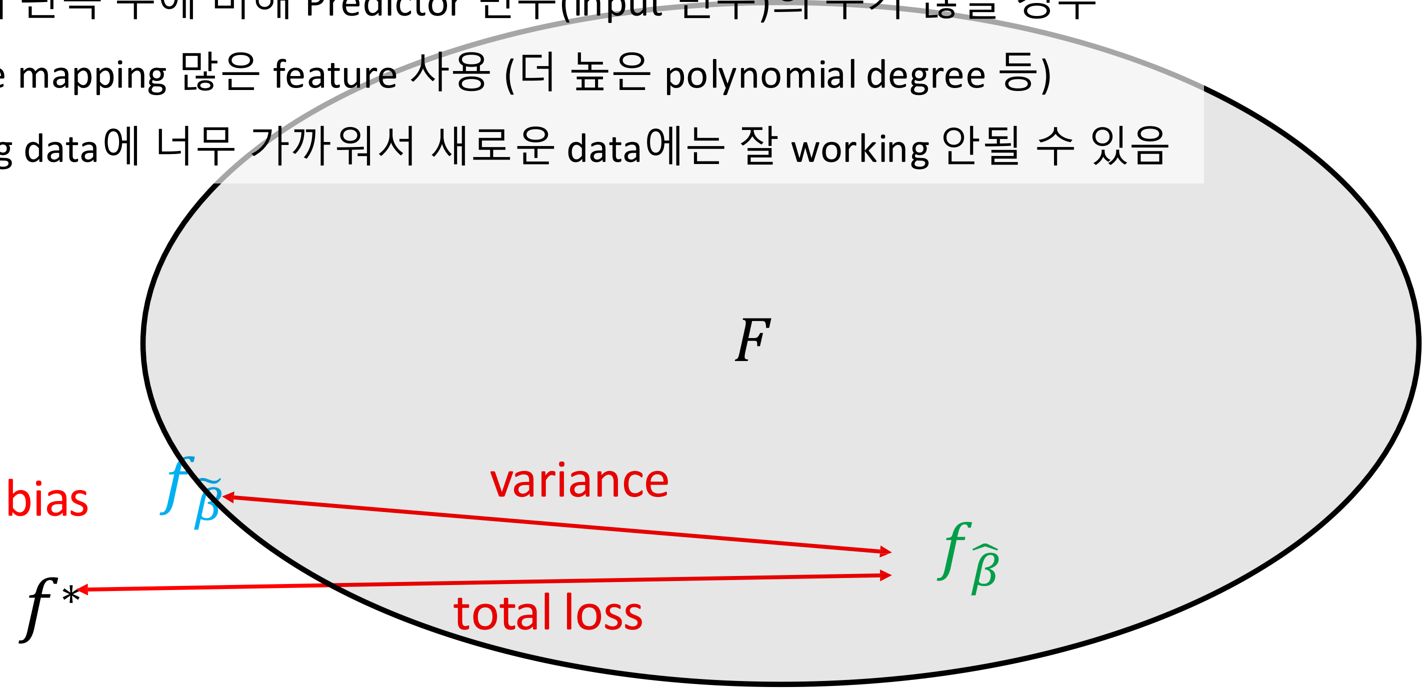
Overfitting?
Underfitting?



Bias-Variance Tradeoff(Overfitting)

주로 모델이 너무 복잡한 경우

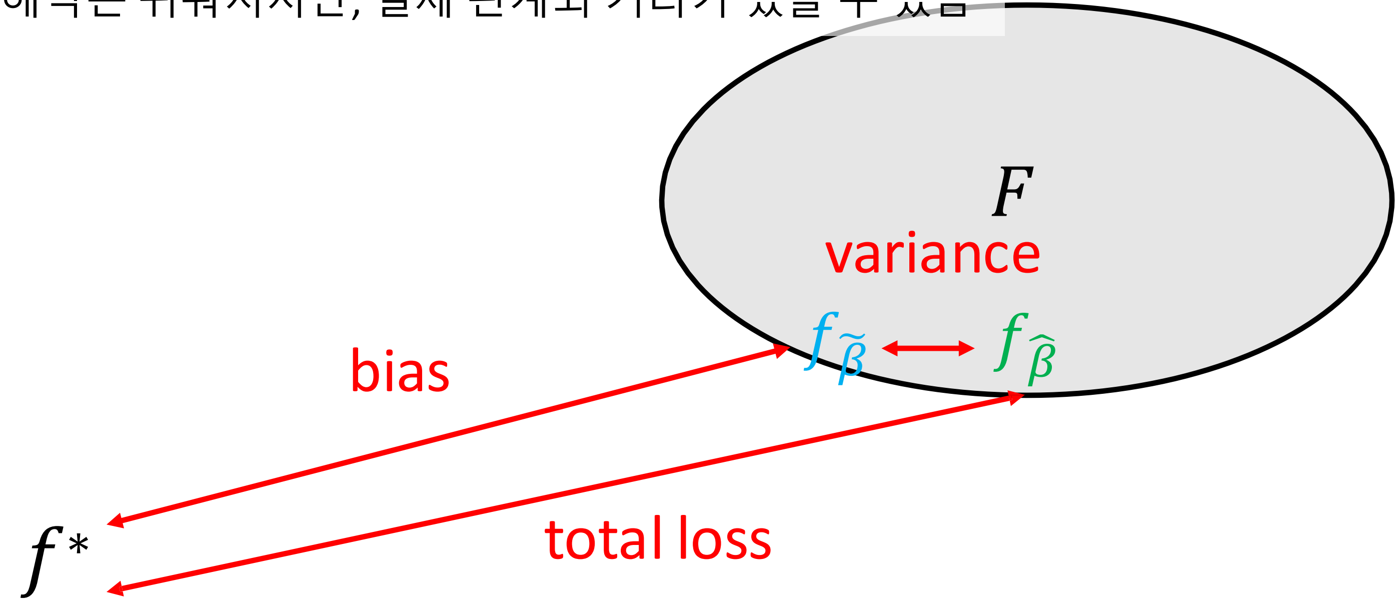
- 데이터 관측 수에 비해 Predictor 변수(input 변수)의 수가 많을 경우
- feature mapping 많은 feature 사용 (더 높은 polynomial degree 등)
- training data에 너무 가까워서 새로운 data에는 잘 working 안될 수 있음



Bias-Variance Tradeoff(Underfitting)

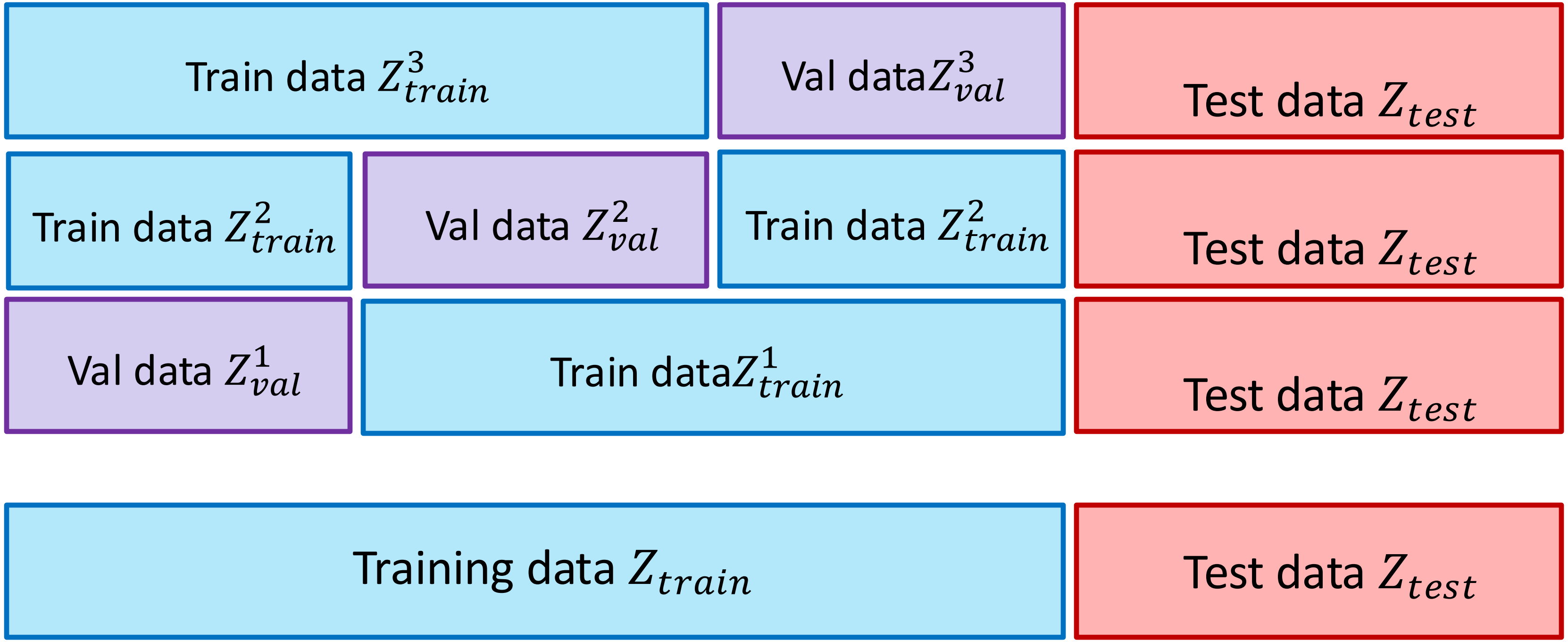
모델이 너무 단순한 경우

- 예: Linear Regression, 적은 변수의 수
- 추론과 해석은 쉬워지지만, 실제 관계와 거리가 있을 수 있음



k-fold CV: to estimate test loss(error)

- Goal: Find a hyperparmeter (model complexity, some tuning parameter)
- Split train & test. Apply k-fold CV to train!



Train data Z_{train}^{3} Val dataZ_{val}^{3} Test data Z_{test}

Train data Z_{train}^{2} Val data Z_{val}^{2} Train data Z_{train}^{2} Test data Z_{test}

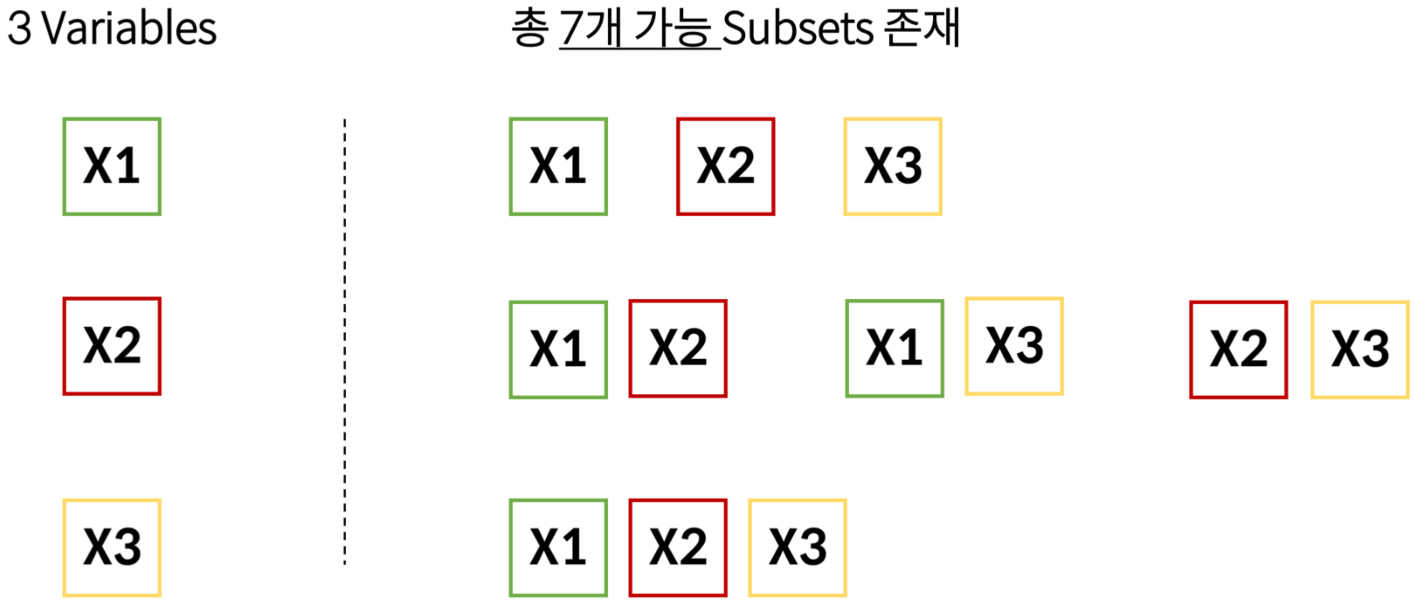
Val data Z_{val}^{1} Train dataZ_{train}^{1} Test data Z_{test}

Training data Z_{train} Test data Z_{test}

For reporting test loss

Feature Selection: Exhaustive Search

- Best combination of features
- 경우의 수: $2^p - 1$ - 시간이 너무 많이 걸림
- 예제: p=3



Feature Selection: Sequential Selection

- Forward Selection (Addition)
  - 0 variable부터 시작. 하나씩 추가
  - 한번 추가된 변수는 다시 지우지 않음

- Backward Selection (Elimination)
  - Full model에서 시작. 하나씩 제거
  - 한번 제거된 변수는 다시 추가하지 않음

- Step-wise Selection
  - 위 기법 혼합

High Variance in Linear Regression

- Multicollinearity
  - $\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ <- 이 값이 매우 불안정함 (**분산이 매우 높음**!)
  - 계산이 가능하더라도 결과값에 대한 신뢰도는 매우 낮게 됨

- High-dimensional data (n < p)
  - $\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ <- 이 값이 매우 불안정하거나 무수히 많은 해(해가 유일하지 않음!) (**분산이 매우 높음**!)

Linear Regression with L_p Regularization

loss without regularization

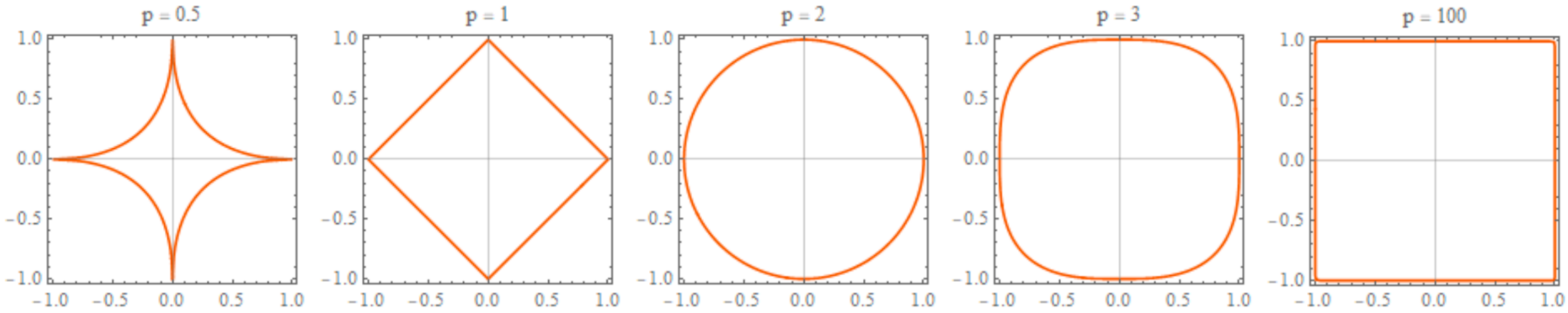
- **Original loss + regularization:**

$$L(\beta; Z) = \boxed{\frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta^T x_i)^2} + \lambda \cdot \|\beta\|_p$$

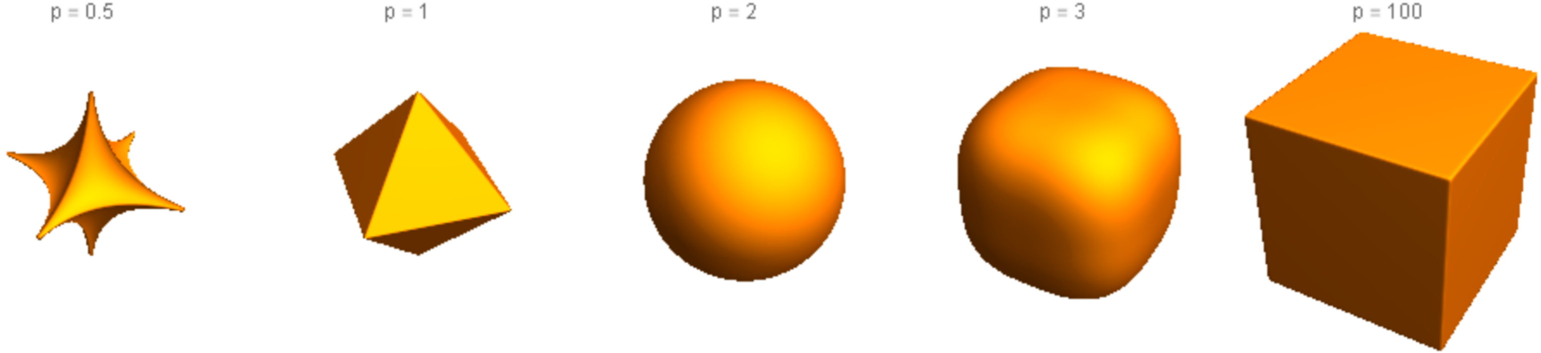
- λ is a **hyperparameter** that must be tuned (satisfies $\lambda \geq 0$)

L_p Norm?

When $x \in \mathbb{R}^2$, $\{x \in \mathbb{R}^2 : \|x\|_p = 1\}$ is



When $x \in \mathbb{R}^3$, $\{x \in \mathbb{R}^2 : \|x\|_p = 1\}$ is



Linear Regression with L_p Regularization

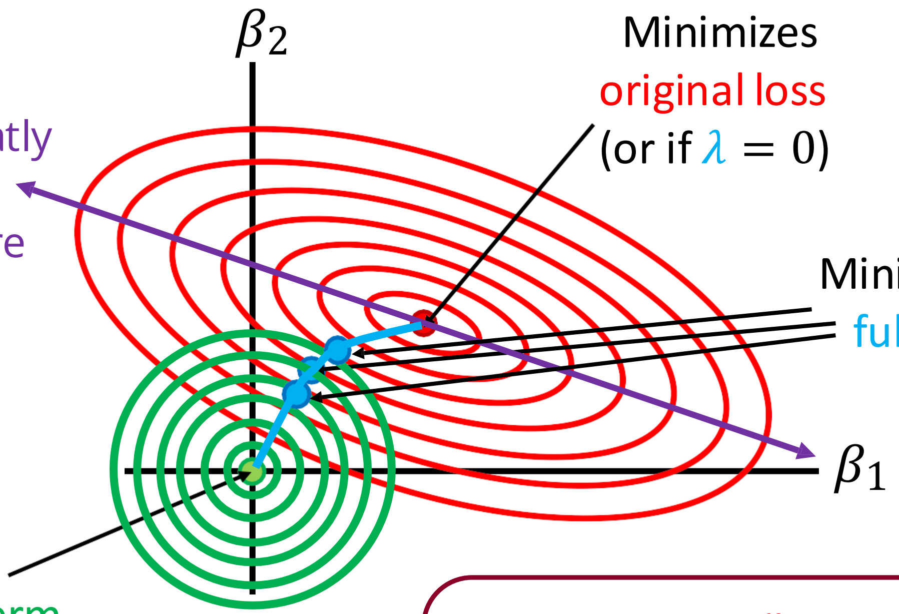
- **Original loss** + **regularization:**

loss without regularization

$$L(\beta; Z) = \boxed{\frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta^T x_i)^2} + \lambda \cdot \|\beta\|_p$$

- λ is a **hyperparameter** that must be tuned (satisfies $\lambda \geq 0$)
- Penalty term: we want to reduce the loss. If λ is large, more penalty on $\|\beta\|_p$
  - A large λ encourages “simple” function.
  - Tuning λ = Tuning **bias-variance tradeoff**

Intuition L_2 Regularization



Loss varies greatly in this direction
□ Penalizes more

Minimizes
original loss
(or if $\lambda = 0$)

Minimizes
full loss

Minimizes
regularization term
(or if $\lambda \to \infty$)

- At this point, the gradients are **equal**
- (with opposite sign)
- Tradeoff depends on choice of λ

$$L(\beta; Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta^T x_i)^2$$

subject to $\|\beta\|_p \le c$

Ridge Regression(L_2 Regularization)

Ridge Regression is the linear regression with L2 penalty

- Minimize

$$\hat{\beta}^{Ridge} = \arg\min_{\beta\in\mathbb{R}^p} L(\beta;Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta^T x_i)^2 + \lambda \cdot \|\beta\|_p$$

- The objective function has a closed-form solution (analytic solution) as below

$$\hat{\beta}^{Ridge} = \boxed{\left(X^T X + \lambda I_p\right)^{-1}} X^T Y$$

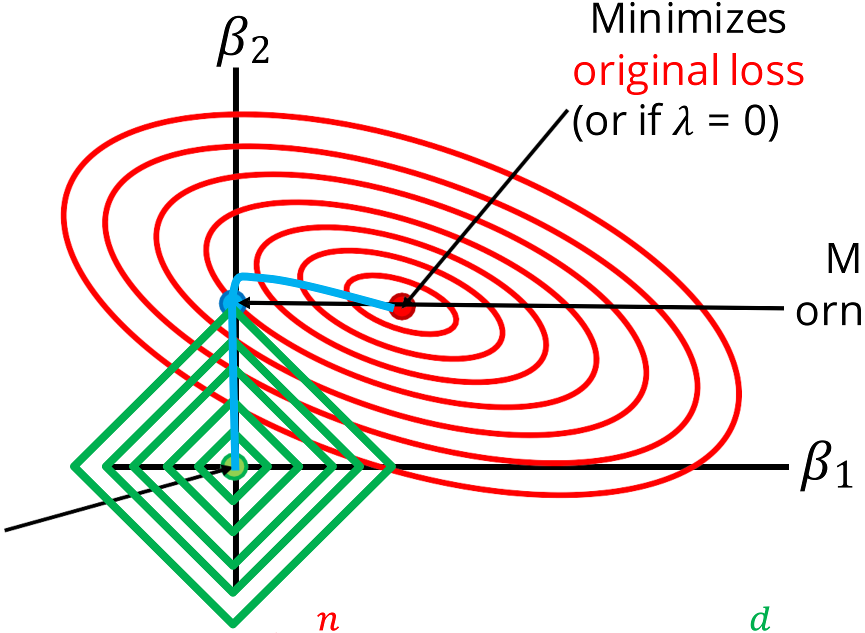
inverse is stable

- Remark: if the predictors are orthonormal, (variables are not correlated), it has a form of

$$\hat{\beta}^{Ridge} = \boxed{\frac{\hat{\beta}}{1+\lambda}}$$

coefficients are shrunken

Intuition on L_1 Regularization



Minimizes
original loss
(or if $\lambda = 0$)

Minimizer of full loss at c
orner sparse ($\beta_1 = 0$)!

Minimizes
regularization term
(or if $\lambda \to \infty$)

$$L(\beta;Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^T x_i)^2 + \lambda \sum_{j=1}^{d}|\beta_j|$$

Lasso: Feature Selection via L_1 Regularization

- 전통적인 Sequential Feature Selection 방법은 High-dimension 문제에서 시간이 너무 오래 걸리거나 Full model 계산이 불가능

- **L_1 Regularization:** Model Estimation 과정에서 동시에 Feature Selection

- 다른 performance measure 기반의 선택이 아닌 model train 과정에서 자체 Feature 학습

Feature Standardization

- **Ridge/Lasso: rescaling of features affects the output**

- **Solution:** Rescale features to zero mean and unit variance

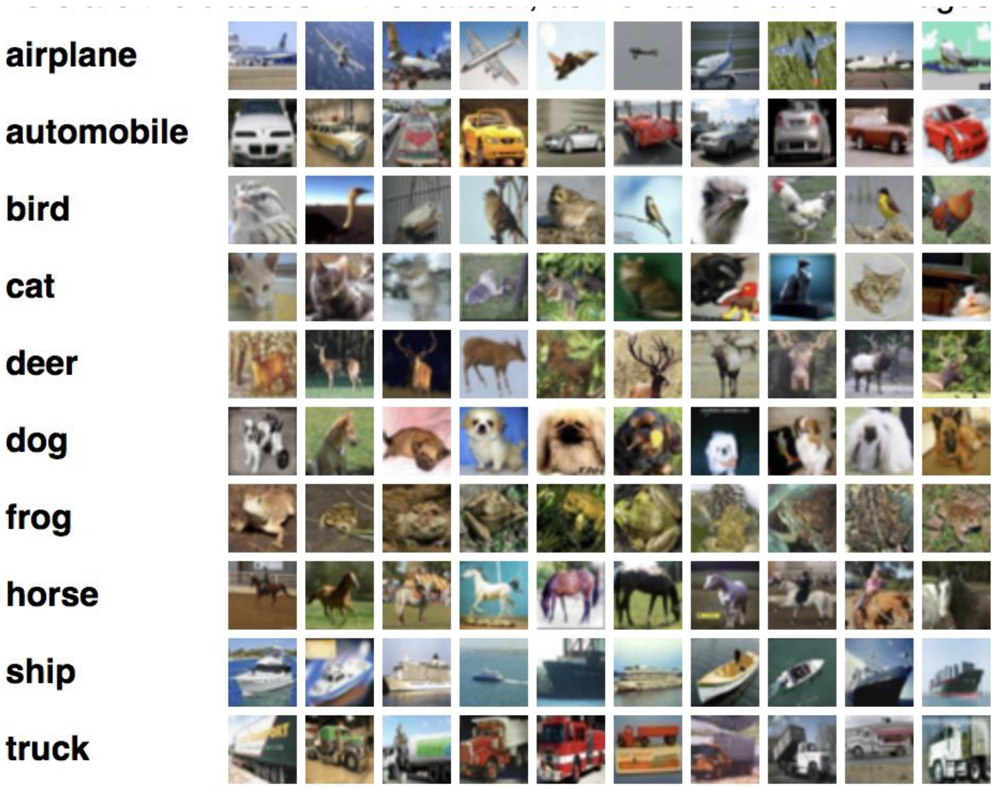
$$x_{i,\mathrm{j}} \leftarrow \frac{x_{i,\mathrm{j}} - \mu_{\mathrm{j}}}{\sigma_{\mathrm{j}}} \qquad \mu_{\mathrm{j}} = \frac{1}{N} \sum_{i=1}^{N} x_{i,\mathrm{j}} \qquad \sigma_{\mathrm{j}} = \frac{1}{N} \sum_{i=1}^{N} \left(x_{i,\mathrm{j}} - \mu_{\mathrm{j}}\right)^2$$

  - **Note:** When using intercept term, do not rescale $x_1 = 1$

- **Must use same transformation during training and for prediction**
  - Compute on standardization on training data and use on test data

Supervised Learning : Classification

Classification y x



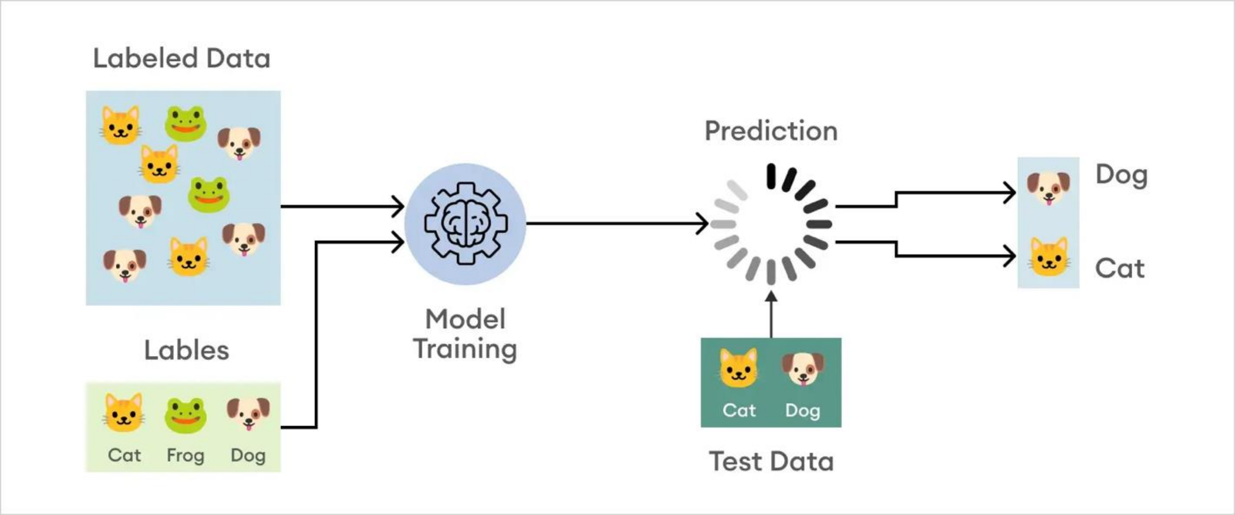
Data
(x,y)
(image1, ‘airplane’),
(image2, ‘airplane’),
.
.
.
(image100, ‘truck’)

Supervised Learning : Classification

Classification

fitted f : image -> class

Input → Output



Supervised Learning

Regression vs. Classification

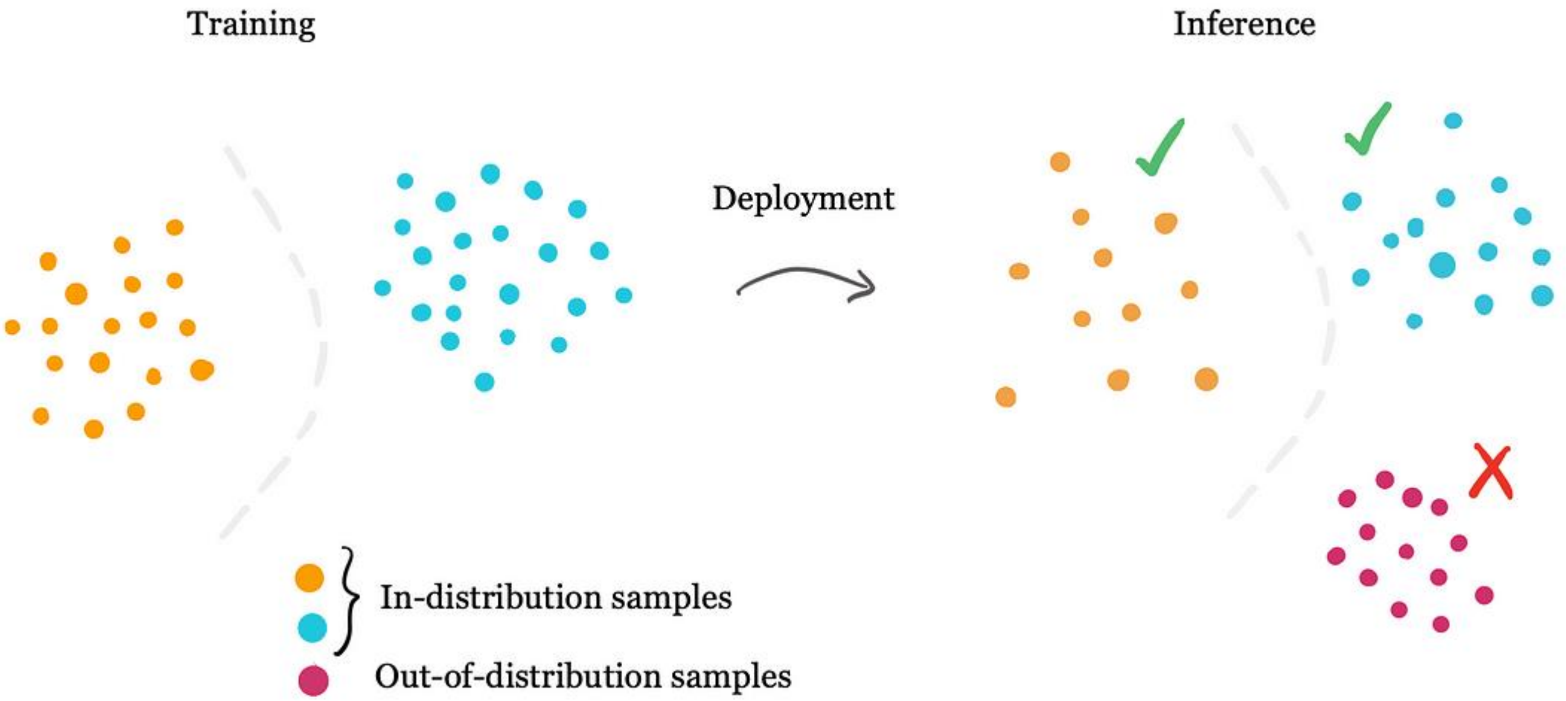
- Where does Y reside?
    - **Regression** : Real vector space
    - **Classification**: A finite set. {c1,c2, …, ck}

- Real Number: Math operations!(+, -, *, /)
- finite set don't have the math operations. cat+dog? cat-dog?

- Differently treated in
    - modeling
    - (E)data coding
    - (T)developing a method to do the task
    - (P)measuring the performance of the method
    - etc

Things to Consider

Danger of Out-of-Domain Application

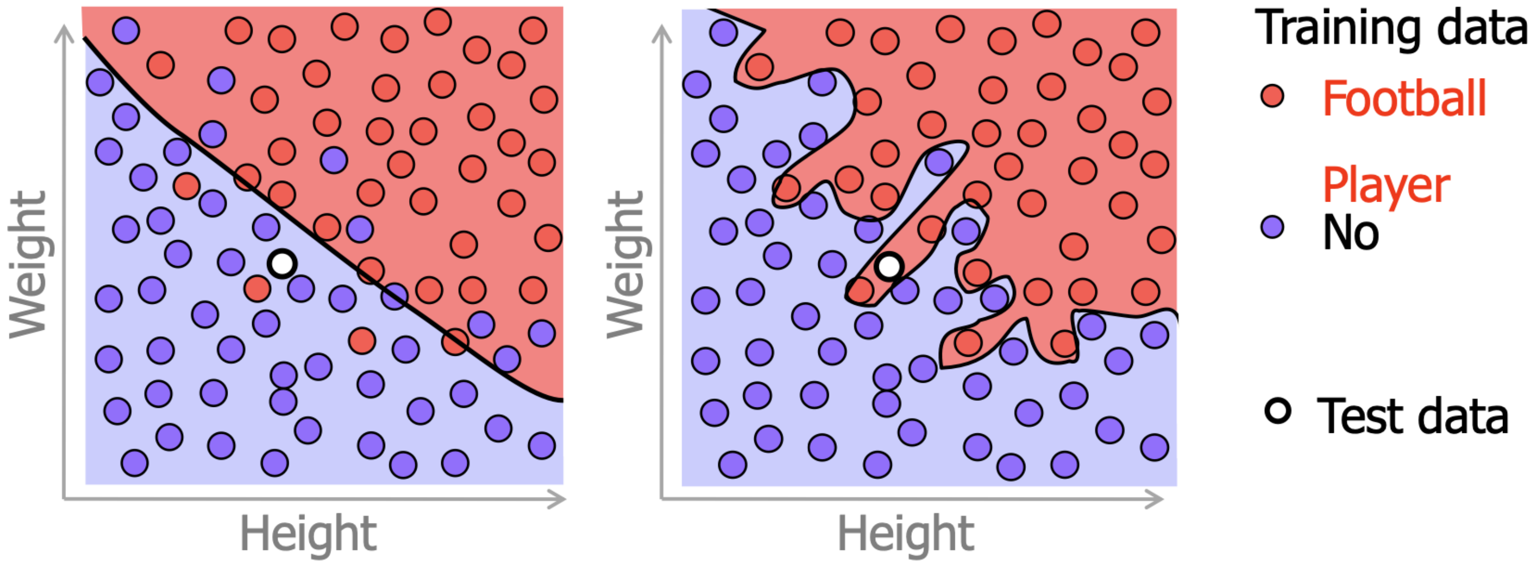


This can easily happen with high-dimensional data in ML algorithms.

Overfitting Problem

A good machine learning algorithm

- Does not **overfit** training data
- **Generalizes** well to test data



Ethical Consideration

- **편향과 차별**: What if we have biased data? Is it okay to learn the algorithm with this?
  - 입학/채용 서류절차에 ML 적용시: 과거의 인종/국적/성별에 대한 편향이 포함된 데이터로 훈련되었을 경우 특정 인종/국적/성별에 불이익을 줄 수 있음.
- **개인정보 보호**
- **안정성과 보안**
  - 시스템에 결함이나 취약점에 이해 예기지 않은 행동으로 인해 사고 발생 가능 (특히 Blackbox-type learning algorithm 을 사용할 때)
- **의사결정의 투명성과 설명가능성**
  - ML/AI 모델은 종종 '블랙 박스'로 작동하여, 그 결정 과정이 불투명
  - 예를 들어, 은행이 ML을 사용하여 대출 승인을 결정할 경우, 모델이 어떻게 그 결정에 도달했는지 설명하기 어려울 수 있고 이는 고객의 불만을 초래