




3. Modeling

KUBIG 학술부



Index

데이터 탐색(전처리) 후
알맞은 모델을 정한다.

1. 여러가지 모델의 종류

: Linear Base Model, Classification, Ensemble,
Clustering, Neural Network

모델을 학습시킨다.
데이터를 가장 잘 설명하는
모델을 찾는다.

2. Loss Function

3. Optimization

: Gradient Descent Algorithm, Newton-Raphson's Method

1. 여러가지 모델의 종류

Linear Base
Model

Classification

Ensemble

Clustering

Neural
Network

1.1. Linear Base Model

➡ Y와 x의 선형적 관계를 기반으로 하는 모델. Simple Model

➡ 1.1.1. Linear Regression

1.1.2. Generalized Linear Model

1.1.3. Penalized Linear Model

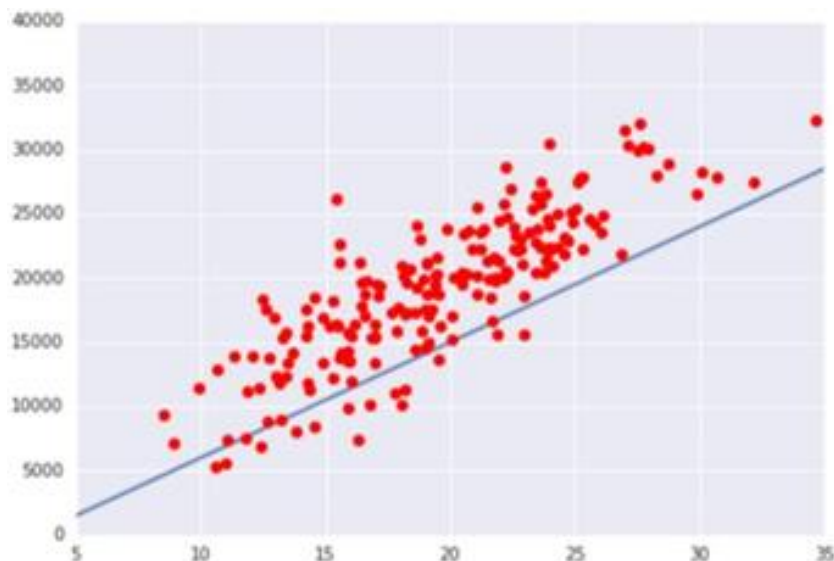
1.1.1. Linear Regression

$$y_i = \alpha + \beta x_i + \varepsilon_i.$$

“Simple Model”

- 해석이 쉽다.
- 가정과 제약이 많다.

EX) ‘오차항이 정규분포를 따른다.’
‘예측 변수 x끼리 선형적으로 독립이다.’



1.1.2. Generalized Linear Model (GLM)

“오차항이 정규분포를 따르지 않는 경우를 포함하는 선형 모형의 확장”

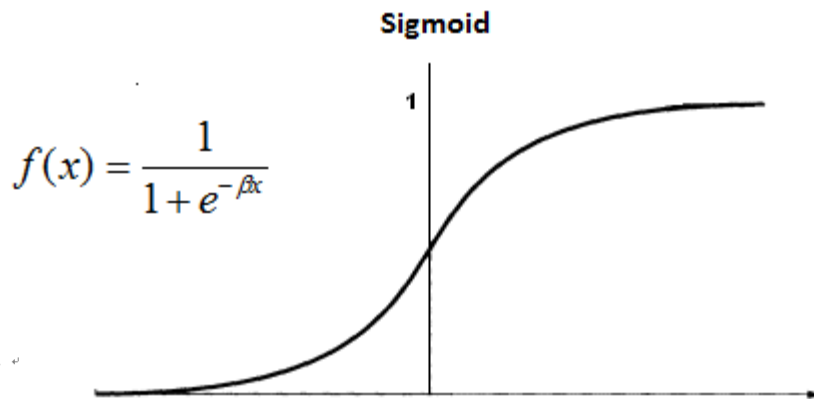
- Logistic Regression : y 가 이항반응 변수일 때

$$p(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p : \text{범위의 문제 발생}$$

↓

$$\ln \frac{p(\mathbf{x})}{1 - p(\mathbf{x})} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

$$p(\mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}} = \frac{1}{1 + e^{-\beta \mathbf{x}}}, \quad 0 < p(\mathbf{x}) < 1$$



1.1.3. Penalized Linear Model

Penalty가 부여된 선형 모델  Overfitting 방지

(Overfitting: 학습 데이터를 설명하는데 너무 집중한 나머지, 실제 데이터에는 예측력이 떨어지는 현상.
bias는 작은데, 예측치들 간 variance는 큰 현상.)

- Ridge Regression : 가중치(계수)들의 값을 감소시킴.
- Lasso Regression: 변수선택 기능이 있음.
- Elastic Net: Ridge와 Lasso의 혼합형 모델

1.2. Classification

➡ Y가 범주형 변수일 때, 이를 분류하는 모델.

➡ 1.2.1. K-nn

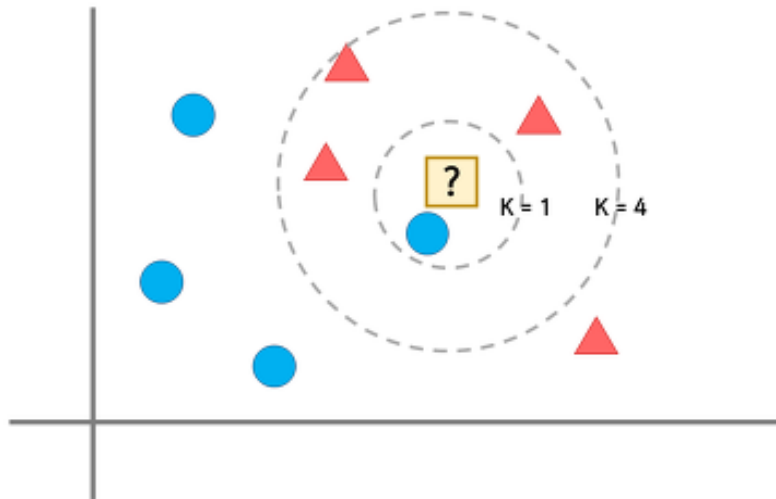
1.2.2. Decision Tree

1.2.3. Support Vector Machine

1.2.1. K-nn (nearest neighbor)

- 어떤 데이터 포인트의 종류를 결정할 때, 그 점에서 가장 가까운 것이 무엇이냐를 중심으로 결정.

K: 주변의 개수



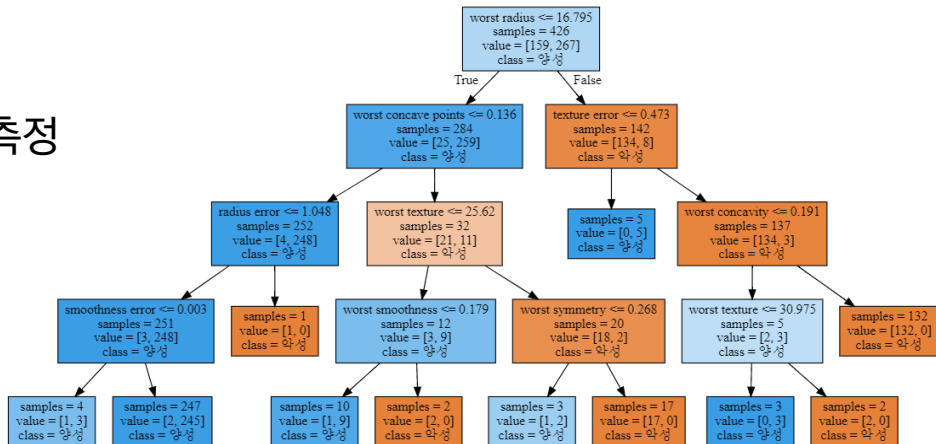
1.2.2. Decision Tree

- 질문에 질문을 이어 데이터를 분류해 나가는 기법.

아래 단계로 내려갈수록 동질적이어야 함.

불순도: 동질적이지 않은 정도

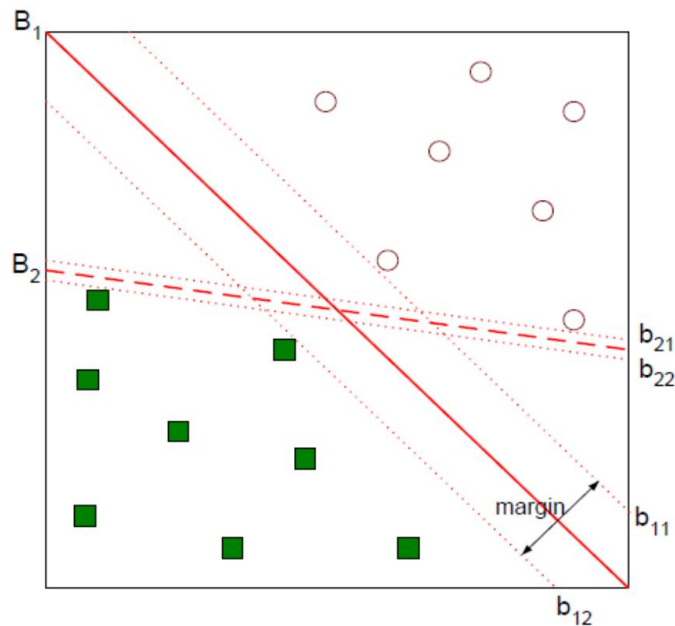
→ 엔트로피, 지니 불순도 등으로 측정



1.2.3. Support Vector Machine (SVM)

(1) Linear SVM

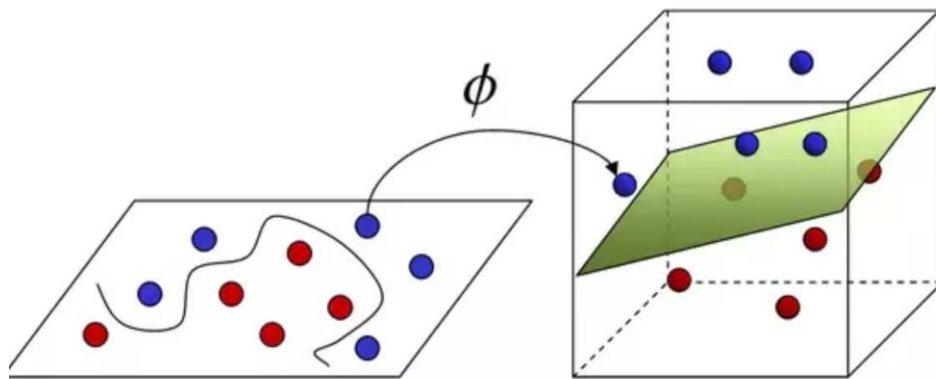
- 데이터를 선형적인 분류 경계면으로 분류.
특히, 분류 경계면 중에서도 margin을 최대화하는
분류경계면을 찾는 기법.
(margin: 분류 경계면과 가장 가까운 점들과
분류 경계면 사이의 거리)
- 데이터를 선형적으로 분리할 수 없을 때는?



1.2.3. Support Vector Machine (SVM)

(2) Kenelized SVM

- 데이터를 선형적으로 분리할 수 있도록 고차원의 feature space로 변환시킨다.



1.3. Ensemble

➡ 자료로부터 여러 개의 예측 모델을 만든 후 예측 모델들을 종합하여 하나의 최종 예측 모델을 만드는 방법

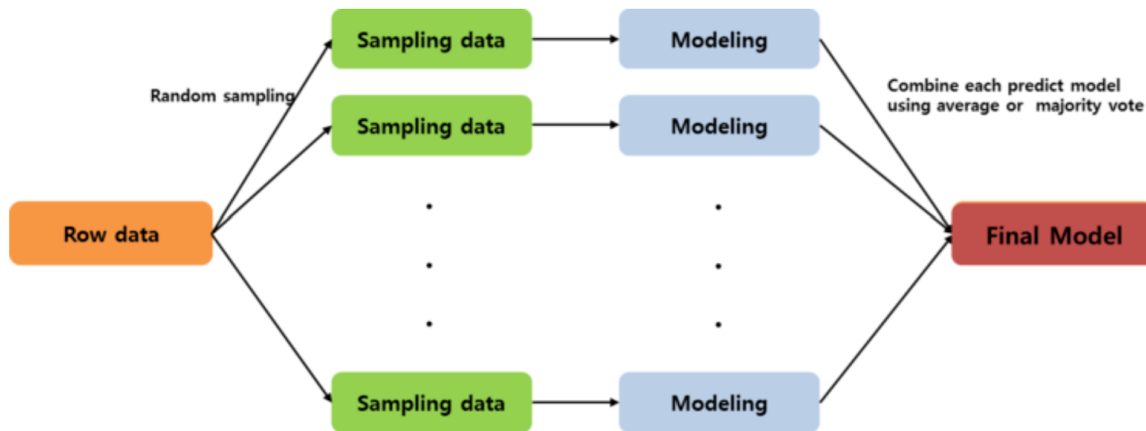
➡ 1.3.1. Bagging

1.3.2. Random Forest

1.3.3. Boosting

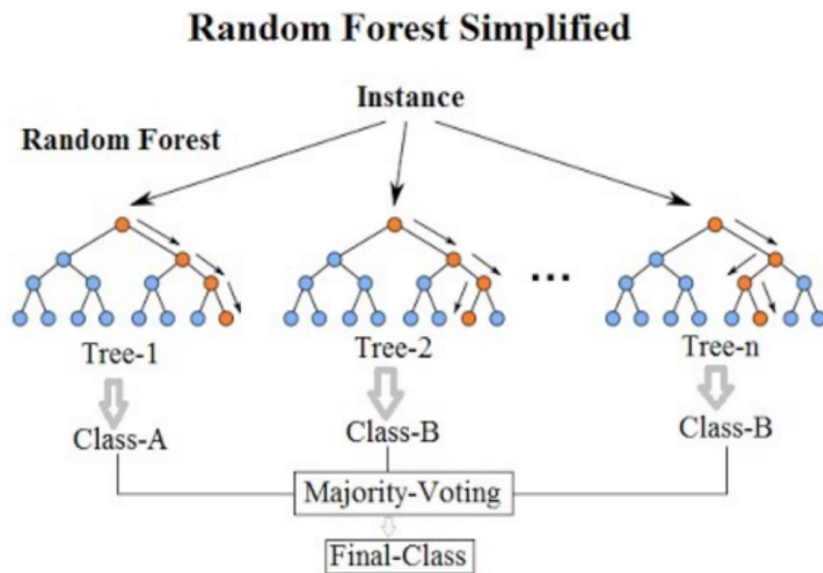
1.3.1. Bagging

- 표본 데이터로부터 세부 표본들을 랜덤 추출한 다음에, 각 세부 표본에 대해 모델을 생성하고, 각 모델의 결과를 다수결 또는 평균을 통해 최종 결과를 예측하는 방법.



1.3.2. Random Forest

- 다수의 Decision Tree를 결합.
- Bagging의 개념에 착안하여 만들어졌지만, 차이점이 존재한다!
 - ➡ randomness를 표본 추출 뿐만 아니라 feature 선택에도 부여하였다는 것.
 - ➡ 그리하여, tree들의 이질성 강화, 상관성 감소.



1.3.3. Boosting

Boosting

- 직렬적인 구성.
- 이전 시행의 결과가 다음 시행으로 이어짐.



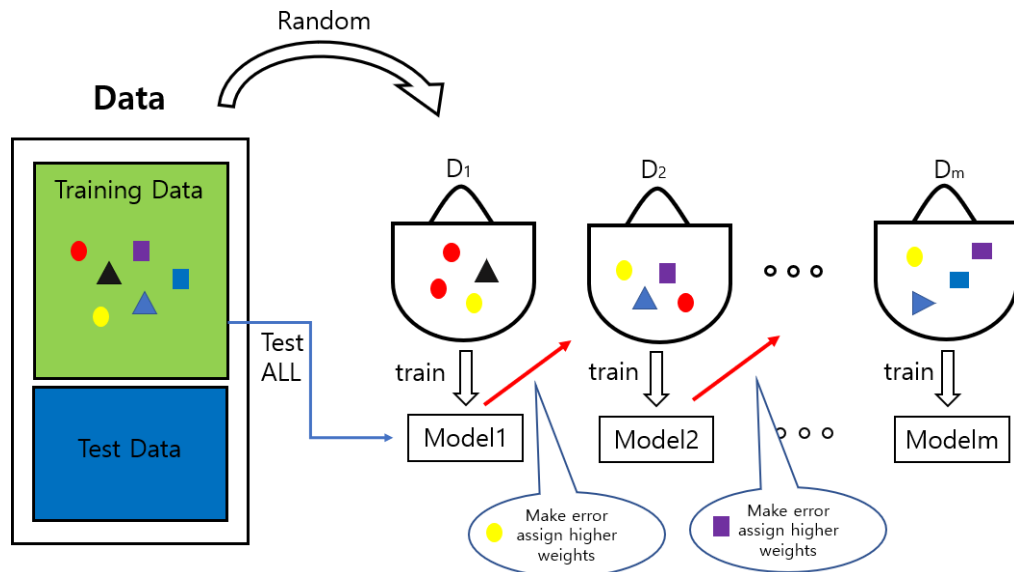
Bagging

- 병렬적인 구성.
- 각 시행(bag)마다 독립적으로 모델을 구성.

➡ Boosting의 종류: Ada Boost, Gradient Boost, XG Boost

1.3.3.1. Ada Boost

- 간단하고 성능이 떨어지는 모델(weak learner)들이 상호보완하도록 단계적으로 학습해서, 최종적으로는 성능을 매우 높이는 원리.

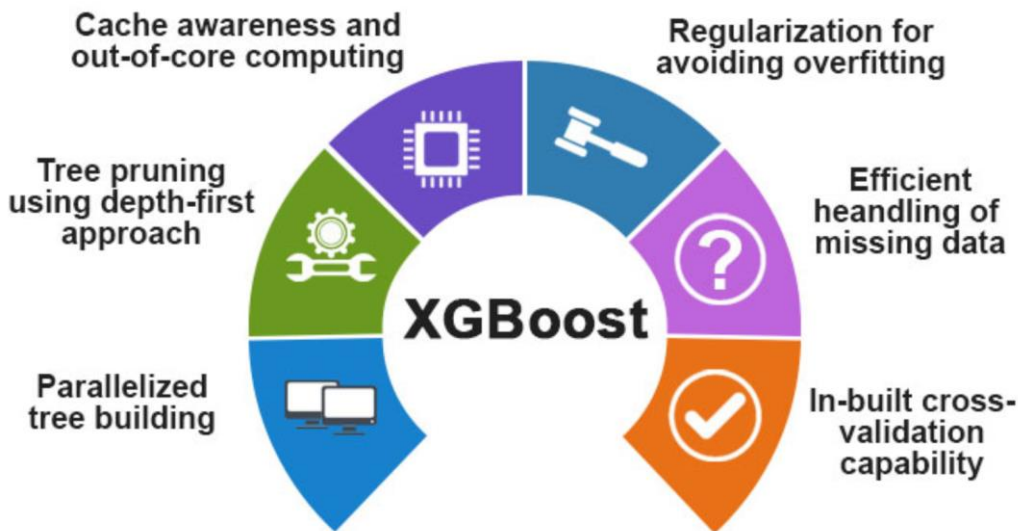


1.3.3.2. Gradient Boosting

- Gradient Descent Algorithm을 사용해서 Ada Boost의 성능을 개선시킴.
- 이전 모델에서 발생한 오차를 Loss function으로 표현해서, Gradient Descent 알고리즘을 통해 오차를 최소화하는 방향으로 다음 모델을 구성.

1.3.3.3. XG Boost

- Gradient Boosting 대비 속도와 성능의 비약적인 향상.
- 시스템 자원을 효율적으로 활용.



1.4. Clustering

➡ 비슷한 데이터 포인트들을 하나의 큰 cluster로 묶어서 여러가지 cluster로 나타내는 방법.

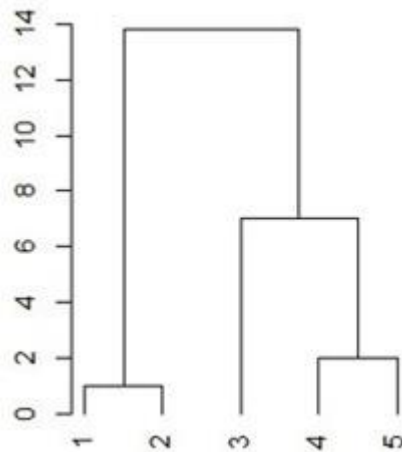
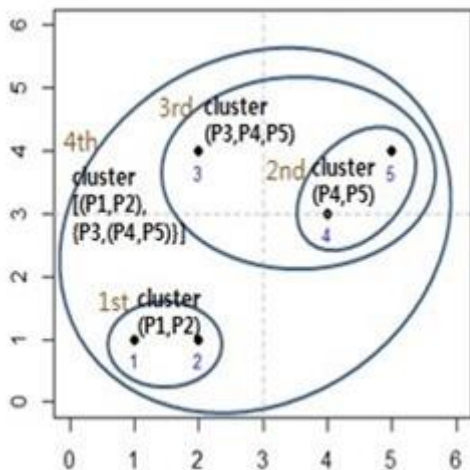
➡ 1.4.1. Hierarchical Clustering

1.4.2. K-means

1.4.3. DBSCAN

1.4.1. Hierarchical Clustering

- 하나의 cluster가 다른 cluster를 포함하는 구조로 cluster들을 형성하는 기법.
EX) '동물'이라는 cluster 안에, '개'라는 cluster 안에, '웰시코기'라는 데이터 포인트.



1.4.2. K-Means

- 비계층적 Clustering.
- 거리를 기반으로 cluster 형성.
군집 내 점들 간의 거리는 최소화,
군집 간의 거리를 최대화.
- K: cluster의 개수
(사전에 지정해야 하는 hyperparameter)

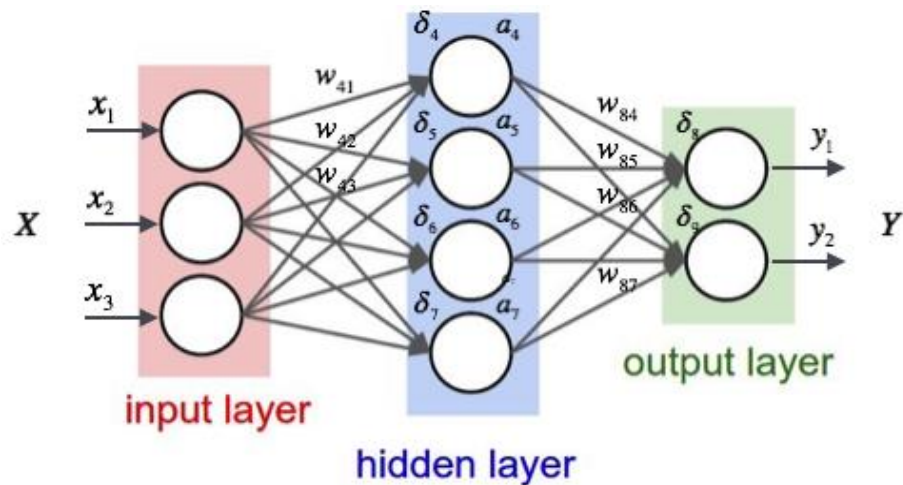


1.4.3. DBSCAN

- 비계층적 Clustering.
- 밀도를 기반(Density-Based)으로 cluster 형성.
밀도가 높은 곳에 군집을 형성하고, 밀도가 낮은 부분은 noise로 취급.
- Outlier에 취약하지 않다는 장점이 있음.

1.5. Neural Network

- 입력층, hidden layer, 출력층의 구조.
hidden layer의 개수가 2개 이상일 때가 Deep Neural Network, 즉 딥러닝.



2. Loss Function (Cost Function)

- Loss Function : 모델이 데이터를 얼마나 설명을 못하는지, 모델의 예측치와 실제 값 간의 차이가 얼마나 있는지를 수학적으로 계산한 함수.

$$loss(f) = (y - \hat{y})^2$$

- $Cost Function = \frac{1}{m} \sum_{i=1}^m Loss Function^{(i)}$
- EX) Linear Regression: MSE, Logistic Regression: Cross Entropy

3. Optimization

- Optimization : Cost Function을 최소화하는 모수(parameter)들의 조합을 찾는 문제.

3.1. Gradient Descent Algorithm

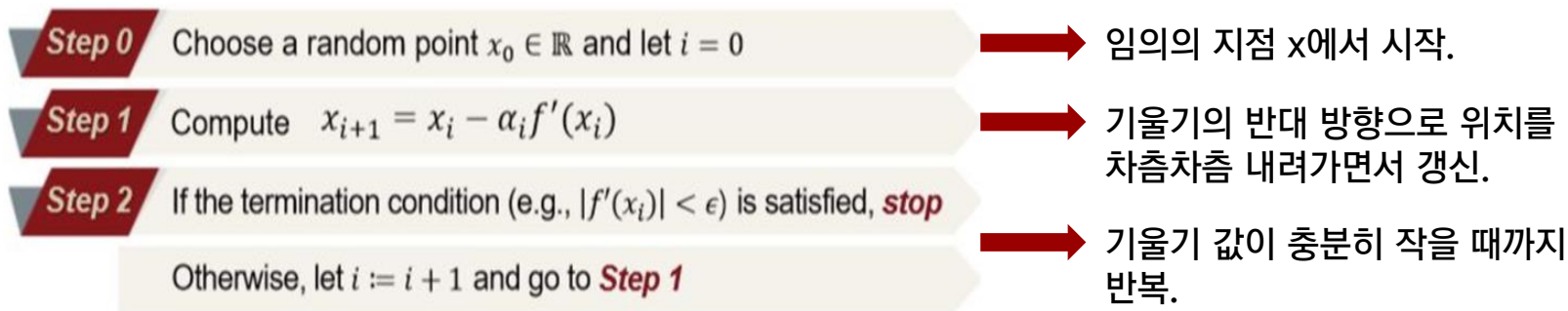
3.2. Newton Raphson's Method

3.1. Gradient Descent Algorithm

- Gradient Descent Algorithm: 기울기가 0인 부분을 찾아가는 알고리즘.
(Cost function을 최소화하는 지점)

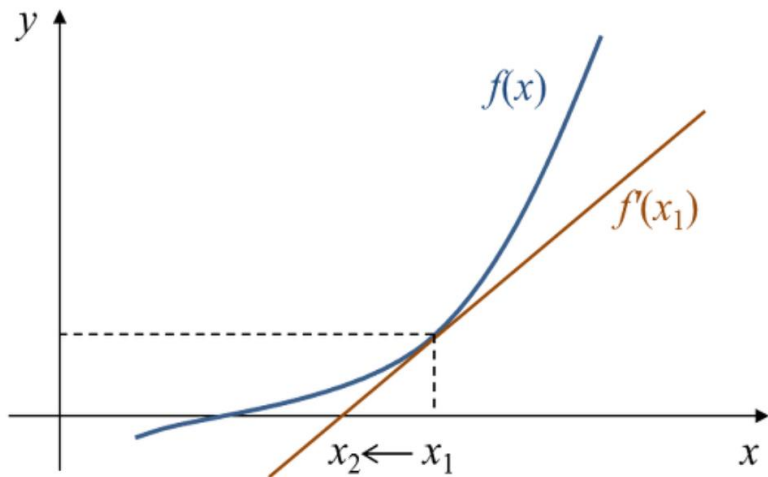
- Single-variable case

$f(x)$: cost function



3.2. Newton Raphson's Method

- Newton Raphson's Method: 간단하고 수렴 속도가 빨라서, $f(x)=0$ 의 근사해를 찾기 유용한 방법.



$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (\because f'(x_n) \neq 0)$$

➡ X의 값을 지속적으로 갱신하는 아이디어가 Gradient Descent Algorithm과 유사.