투빅스 11기 정규과정

ToBig's 10기 박성진

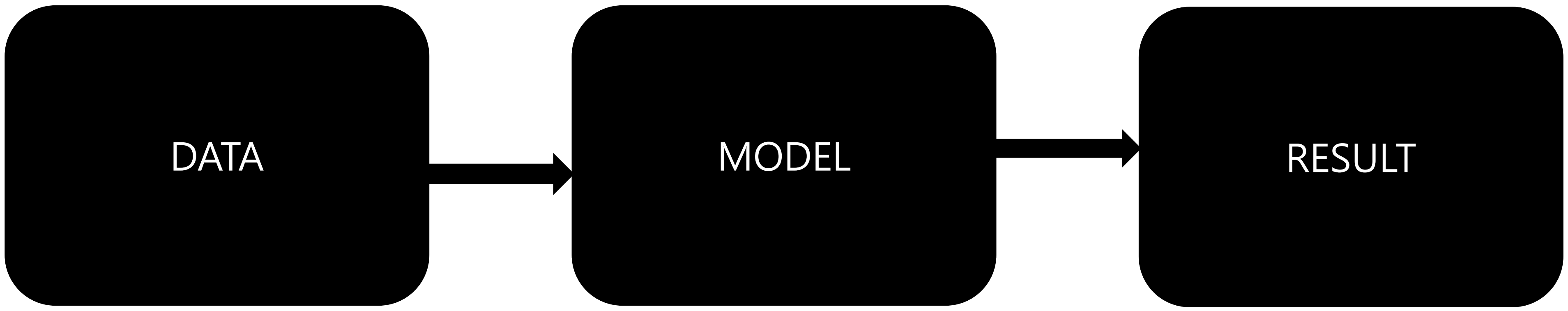
# Logistic Regression & Classification

Contents

# ◆ 머신러닝의 종류

y값

- Label의 유무에 따른 구분: Unsupervised Learning VS Supervised Learning
- 학습목적에 따른 구분 : 분류/ 회귀/ 군집화/ 연관규칙분석...



## ◆ 머신러닝의 종류 – Label의 유무에 따른 구분

- Supervised Learning

classification, Regression Model

✓ Label이 존재하는 데이터를 이용해 독립변수 X와 종속변수 Y사이의 관계를 찾아 Test 데이터의 Y를 예측하기 위한 학습

✓ 분류, 회귀 모델이 Supervised Learning에 해당됨

| | X_1 | X_2 | … | X_d |
|---|---|---|---|---|
| Data_1 | … | … | … | … |
| Data_2 | … | … | … | … |
| … | … | … | … | … |
| Data_n | … | … | … | … |

→

Y=f(x)

정답이 주어진 경우!

| Y |
|---|
| … |
| … |
| … |
| … |

## ◆ 머신러닝의 종류 – Label의 유무에 따른 구분

- Unupervised Learning

  ✓ Label이 존재하지 않는 데이터를 이용해 데이터에 내재된 특징을 분석하기 위한 학습

  ✓ 군집화, 연관규칙분석이 비지도학습에 해당됨

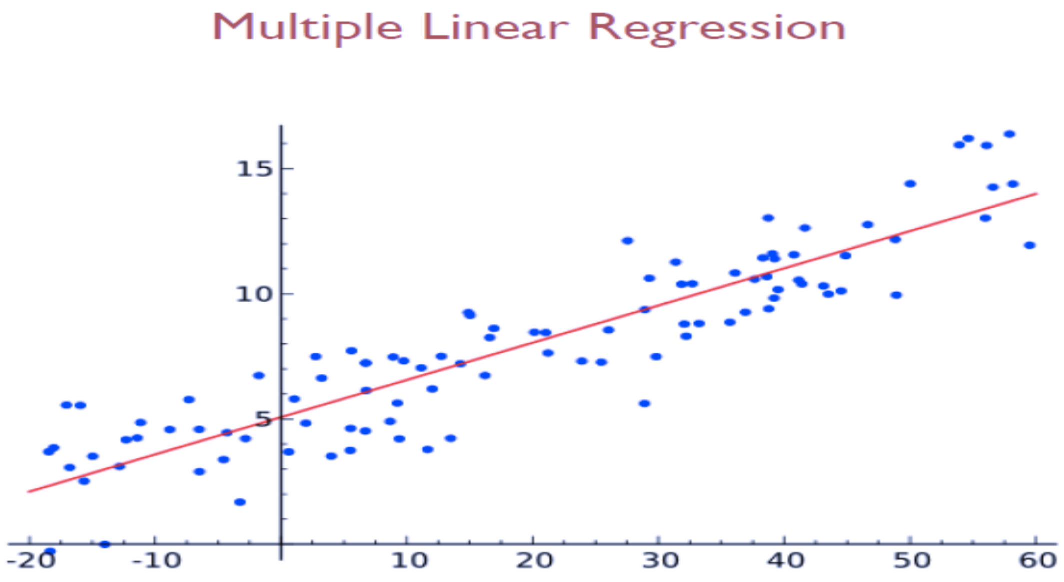
특성을 추출해줌

| | X_1 | X_2 | … | X_d |
|---|---|---|---|---|
| Data_1 | … | … | … | … |
| Data_2 | … | … | … | … |
| … | … | … | … | … |
| Data_n | … | … | … | … |

## ◆ 머신러닝의 종류 – 학습목적에 따른 구분

- Regression - 회귀 (y 값이 연속형)  연속형 변수 예측

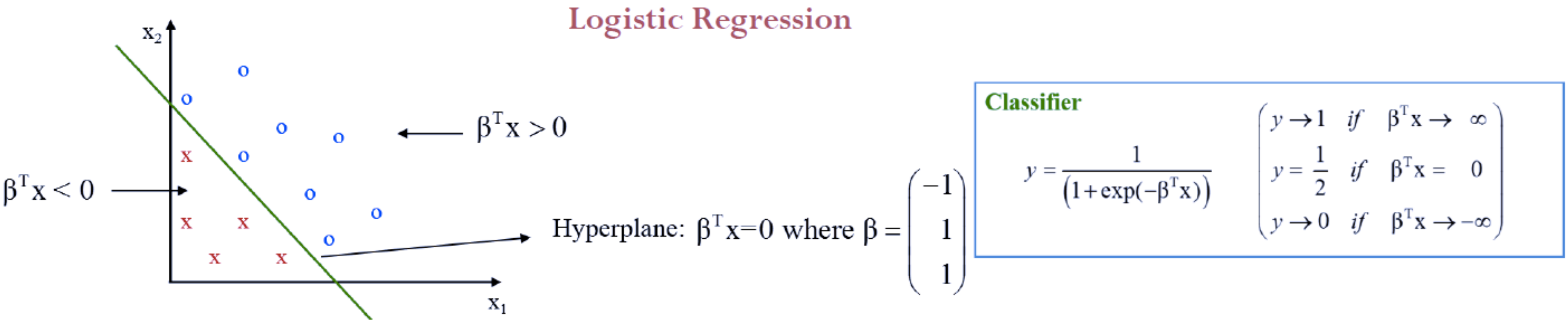
  ✓ 연속형(Continuous)변수를 예측하는 방법론
  ✓ 다중선형회귀분석, 인공신경망, SVR(Support Vector Regression) 등



# ◆ 머신러닝의 종류 – 학습목적에 따른 구분

- Classification    logistic Regression,    y가 catagorical.

  ✓ 명목형(Categorical) 변수를 예측하는 방법론
  ✓ Logistic Regression, Naïve Bayes, Decision Tree, SVM 등

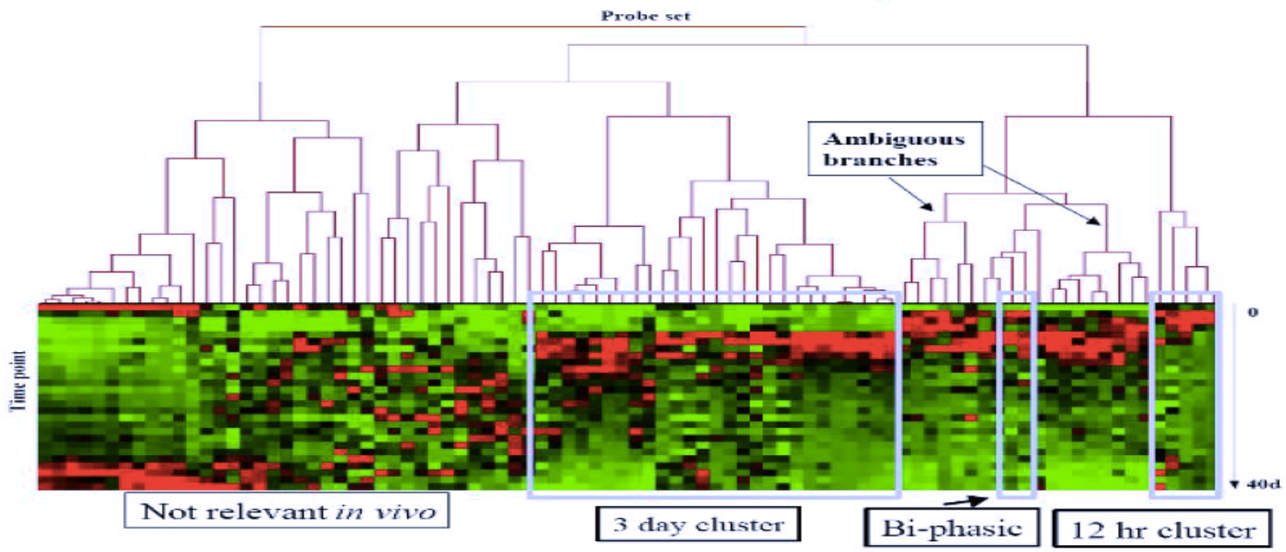


## ◆ 머신러닝의 종류 – 학습목적에 따른 구분

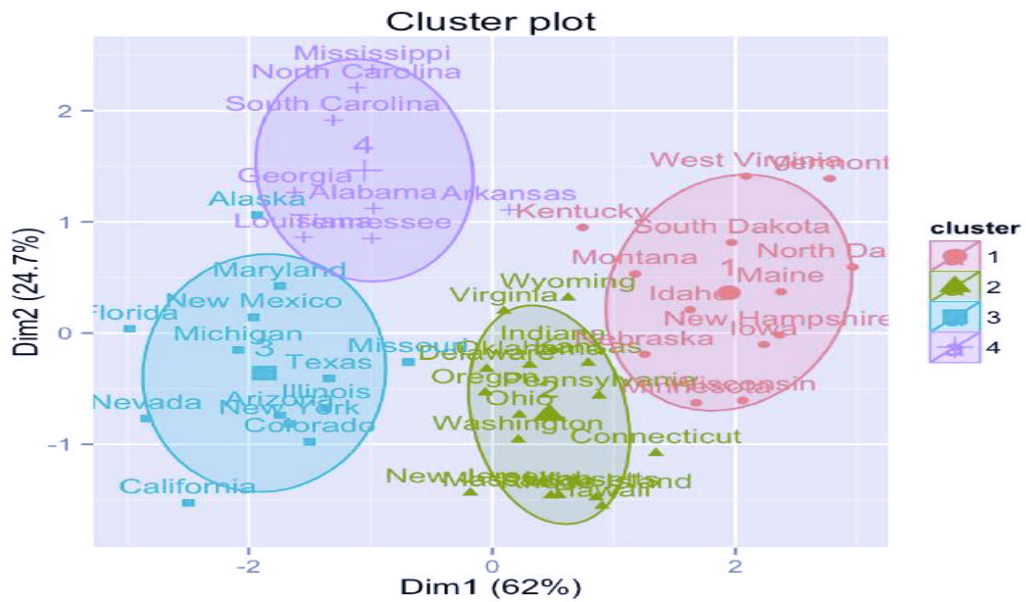
- Clustering 군집화 unsupervised

✓ 유사한 개체들의 집단을 판별하는 방법론
✓ Hierarchical Clustering, K-Means Clustering 등

Hierarchical Clustering




K-Means Clustering



Contents

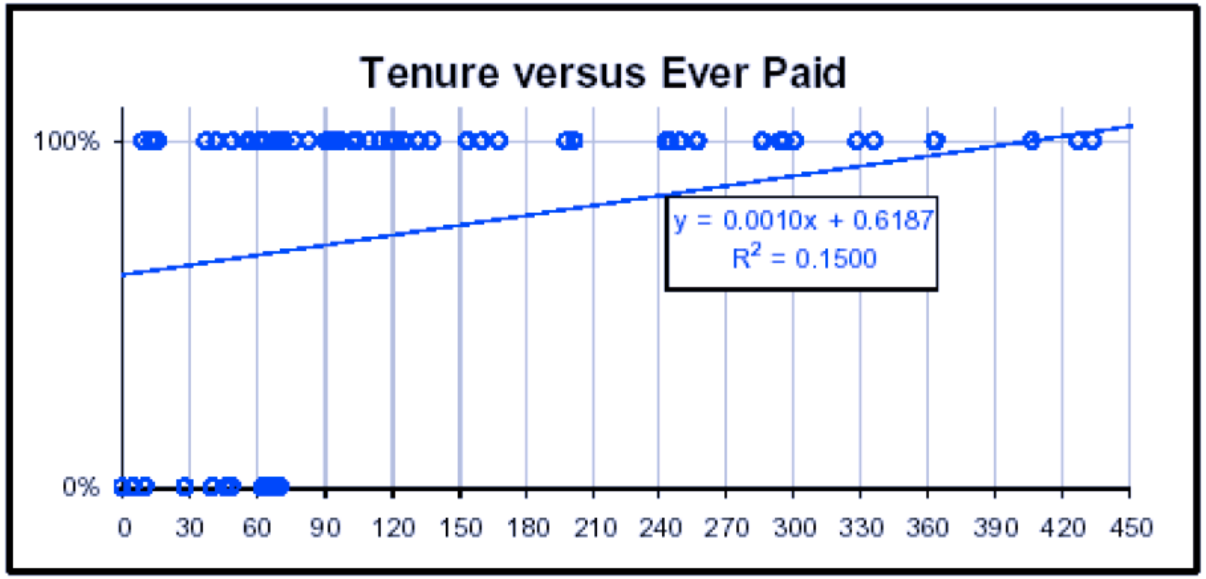
## ◆ Logistic Regression

– y가 Binary 일때.
1, 0
2진형 종속변수

- Logistic Regression 의 목적

  ✓ Binary한 종속변수에 대하여 회귀식의 형태로 모형을 추정함
  ✓ Machine Learning의 종류 중 Supervised Learning, Classification에 속함

$$P(Y=1)=\beta_0+\beta_1x_1+\beta_2x_2+\ldots+\beta_px_p+\varepsilon$$



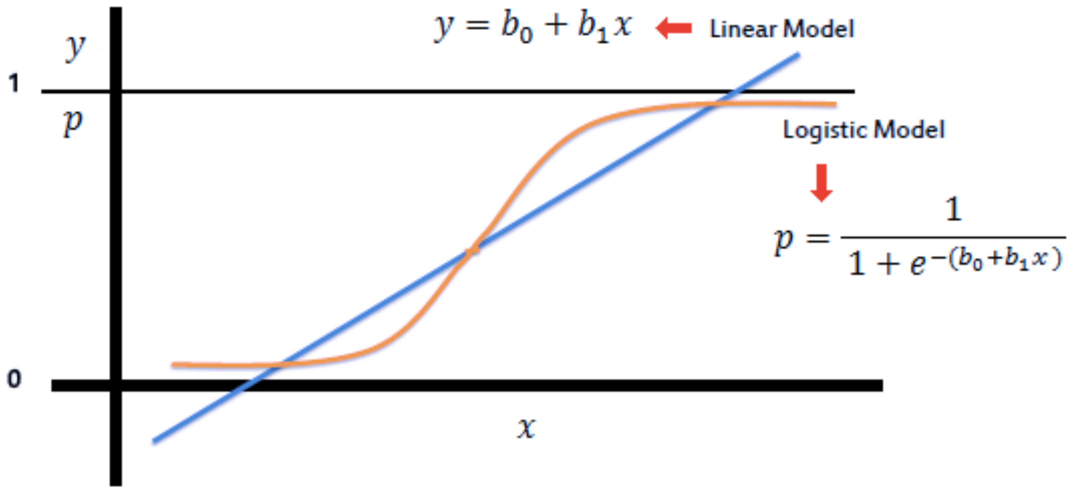
## ◆ Logistic Regression

- Logistic Regression vs Linear Regression

  (범주형 = y)

  ✓ Linear Regression 과 다르게 Categorical Variable을 예측
  Ex) 양성 vs 음성, 성공 vs 실패

  ✓ 주어진 데이터에 직선을 fitting 하는게 아니라 S모양의 logistic function을 fitting

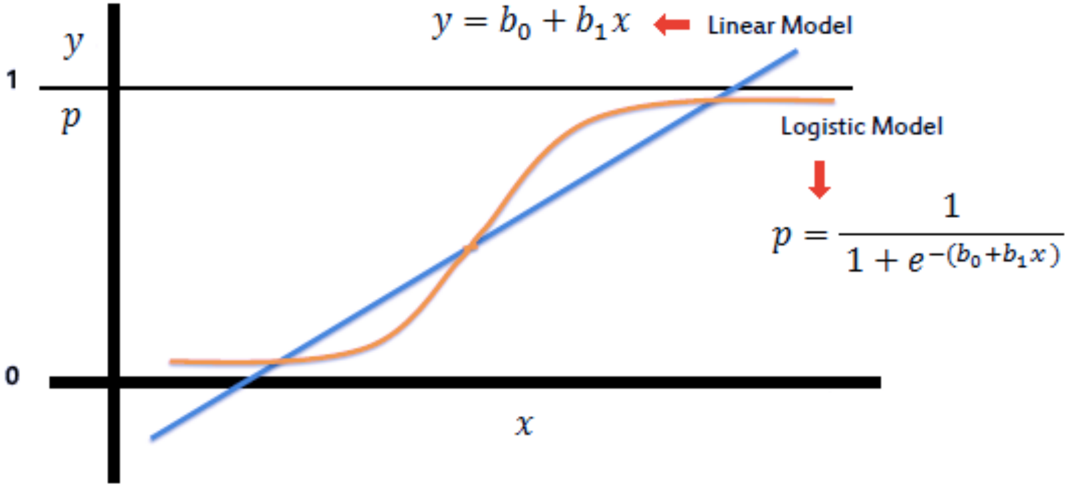


# ◆ Logistic Regression

- Logistic Regression vs Linear Regression

✓ P의 의미
설명변수(X)를 고려했을때 종속변수가 True(1)일 확률
EX) X – 몸무게 , Y – 비만 or Not 비만

0과 1중!
1로 예측할 확률이 얼마나 되는지가 y.




$-\infty$ $\infty$ 인가?

## ◆ Logistic Regression

종속변수가 아닌

- Logistic Regression – Logit을 종속변수로 하는 선형모형을 만드는 것이 Logistic Regression

  ✓ Odds의 Logit값을 종속변수로 사용한 회귀 식을 추정한 뒤, 역산을 통해 구한 성공 확률 값을 기준으로 분류 값을 예측함

$$log(Odds) = log\left(\frac{p}{1-p}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

↓ 양변 exp 취함

$$\frac{p}{1-p} = e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d}$$

→

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}}$$

P(성공확률) 에 대해 정리 $= \sigma(\mathrm{x}, \beta)$

우리가 원하는 값은 p이기 때문

# ◆ Logistic Regression

- Odds & Logit

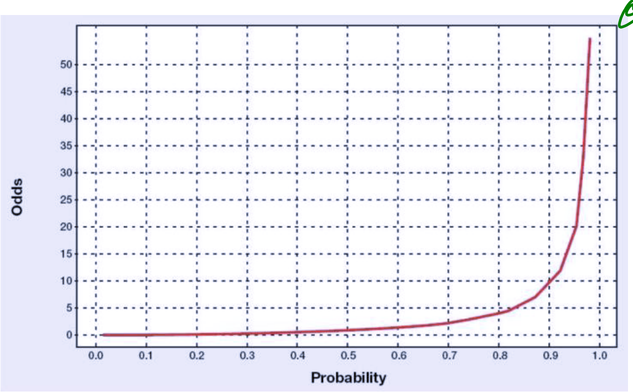
  ✓ Odds : 실패확률 대비 성공확률로 0에서 양의 무한대 사이의 값을 가짐
  ✓ Logit : Odds의 Log값으로 음의 무한대에서 양의 무한대 사이의 값을 가짐

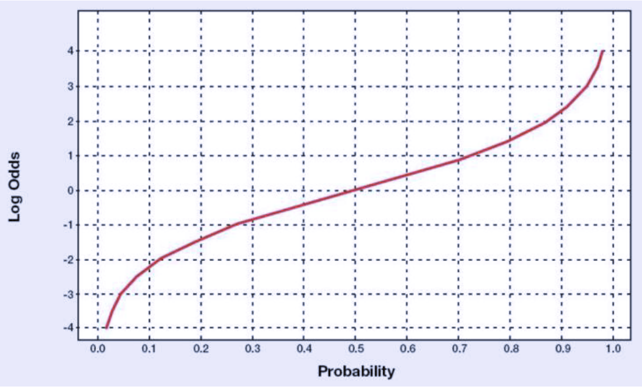
$$Odds = \frac{p}{1-p}$$

비대칭적이고 음수가 없기 때문에

$$\log(Odds) = \log\left(\frac{p}{1-p}\right)$$

Input $[0, \infty)$
output $(-\infty, \infty)$




합성함수라고 생각하면 된다!

## ◆ Logistic Regression

- 회귀 계수의 해석

✓ Linear Regression
- 해당 변수가 1만큼 증가함에 따른 종속변수의 변화량

→ 회귀분석은 직관적으로 변수의 영향력을 알 수 있어 기업에서 좋아함

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

✓ Logistic Regression
- 해당 변수가 1만큼 증가함에 따른 로그 승산의 변화량

→ X가 1 변할때 log(Odds)의 변화량

$$log(Odds) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

# ◆ Logistic Regression

- Maximum Likelihood Estimation
  - ✓ Likelihood Function을 최대화하는 회귀 계수를 추정함
  - ✓ Likelihood Function과 Log Likelihood Function는 회귀 계수에 대한 비선형 함수이므로 Linear Regression과 다르게 명시적인 해가 존재하지 않음
  - ✓ 따라서 Logistic Regression의 경우 MLE를 쓴다.

$$P(x_i, y_i \mid \beta) = \begin{cases} \sigma(\mathrm{x}, \beta) & \text{if } y = 1 \\ 1 - \sigma(\mathrm{x}, \beta) & \text{if } y = 0 \end{cases} = \sigma(\mathrm{x}, \beta)^{y} \left(1 - \sigma(\mathrm{x}, \beta)\right)^{1-y}$$

$$L(\mathrm{X}, y, \beta) = \prod_{i=1}^{R} \sigma(\mathrm{x}_i, \beta)^{y_i} \left(1 - \sigma(\mathrm{x}_i, \beta)\right)^{1-y_i}$$

$$\ln L(\mathrm{X}, y, \beta) = \sum_{i=1}^{R} y_i \ln\left(\sigma(\mathrm{x}_i, \beta)\right) + (1 - y_i) \ln\left(1 - \sigma(\mathrm{x}_i, \beta)\right)$$

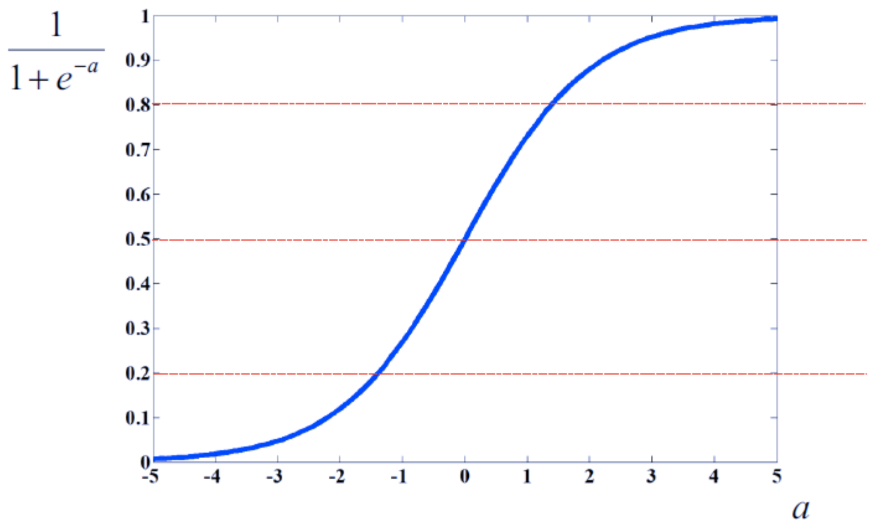
# ◆ Logistic Regression

- Cut-off 설정

  ✓ 회귀계수가 추정되면 설명변수가 주어졌을 때, 성공확률을 구할 수 있고, 해당 확률이 cut-off 이상이면 1, 아니면 0으로 분류할 수 있음

  ✓ 사전확률을 고려한 cut-off 또는 검증 데이터의 성능을 최대화하는 cut-off 등을 사용함

label 중 1인 데이터가 더 많으면 1을 많이 뽑아주고 판단미스 가능 → 0.8로 cutoff 설정 가능!
반대 경우는 cut off 내려서 설정



1의 비중이 높을때

일반적인 cut-off

1의 비중이 낮을때

1이면 0을 나오게함
0.5이상 0.5이하
커지면 조정가능!

# ◆ Logistic Regression

- Performance evaluation index

| Confusion Matrix | | Predict | |
|---|---|---|---|
| | | Positive | Negative |
| Actual | Positive | a | b |
| | Negative | c | d |

✓ Recall(재현율) = $\frac{a}{a+b}$

✓ Precision(정밀도) = $\frac{a}{a+c}$

✓ F1-measure = $\frac{2*recall*precision}{recall+Precision}$

Recall과 Precision 의 조화평균
B/C Ⓡ Ⓟ가 한쪽으로 쏠릴 가능성 O

✓ BCR(균형정확도) = $\sqrt{\frac{a}{a+b} * \frac{d}{c+d}}$ 기하평균

✓ ACC(정분류율)=$\frac{a+d}{a+b+c+d}$

# ◆ Logistic Regression

- Performance evaluation index

| Confusion Matrix | | Predict | |
|---|---|---|---|
| | | Positive | Negative |
| Actual | Positive | TP | FN |
| | Negative | FP | TN |

특이도

✓ *Specificity* = $\frac{TN}{TN+FP} = \frac{d}{c+d}$ = Negative로 판단한 것 중 실제 Negative 비율

✓ *Sensitive (Recall)* = $\frac{TP}{TP+FP} = \frac{a}{a+b}$ = 원래 Positive 중에서 Positive로 예측한 비율

= True Positive Rate

✓ *1 - Specificity = False Positive Rate*

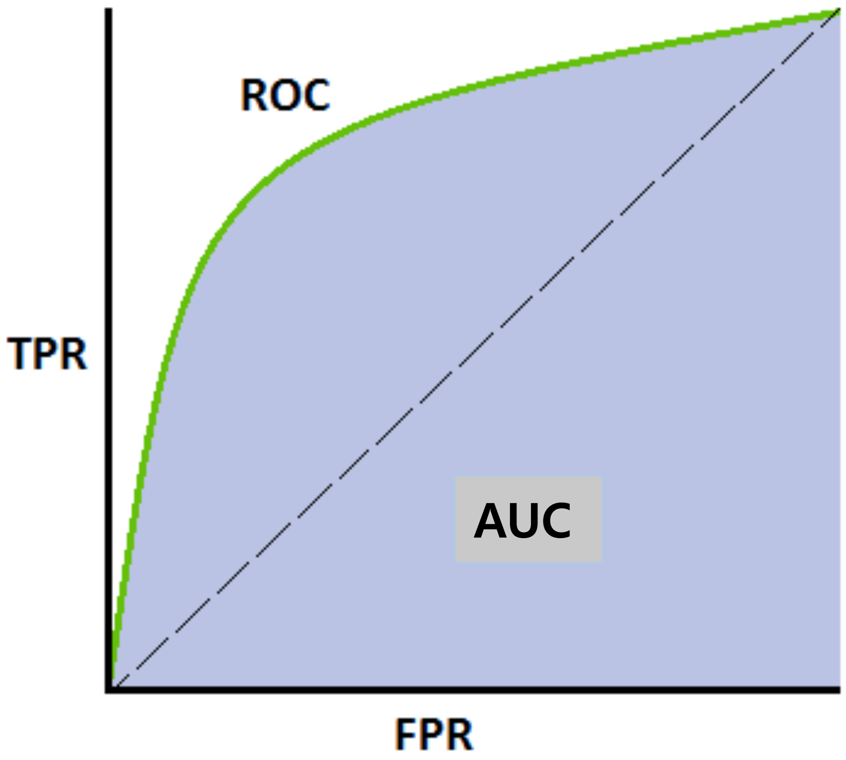
# ◆ Logistic Regression

- ROC Curve

✓ TPR과 FPR은 서로 반비례적인 관계에 있으므로 둘다, **어떤 기준**(언제 1이라고 예측 할 지)을 연속적으로 바꾸면서 측정 해야한다.

✓ 그러면 결국 TPR과 FPR의 여러가지 상황을 고려해서 성능을 판단해야 하는데, 이것을 한눈에 볼 수 있게 한 것이 바로 ROC 커브이다.

✓ AUC : ROC Curve의 밑면적 값, 높을수록 좋다



Contents

# ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 최대우도추정(maximum likelihood estimation)이란 모수(parameter)가 미지의 $\theta$ 인 확률분포에서 뽑은 표본(관측치)들을 바탕으로 확률분포의 모수인 $\theta$ 를 추정하는 기법

✓ 여기에서 우도(likelihood)란 이미 주어진 표본들에 비추어 봤을 때 모집단의 모수 $\theta$ 에 대한 추정이 그럴듯한 정도를 가리킴

✓ 따라서, Likelihood $L(\theta|x)$는 $\theta$ 가 전제되었을 때 표본 $x$가 등장할 확률인 $p(x|\theta)$에 비례함

# ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 우리는 절대로 모수를 알 수 없다. 따라서 표본(sample)을 통해서 모집단의 특성인 모수를 파악해야한다.

✓ 이 때, 우리가 가정하는 것은 각각의 표본을 추출할 때의 확률밀도함수 혹은 확률질량함수의 양상을 알고 있다는 것이다. (즉, 수학적으로 각 sample을 뽑을 확률밀도함수에 대해서 어떤 양상을 따르지 알고 있을 때 모수를 추정하고자 한다면 최대우도법을 쓸 수 있다.)

Unit 03 | MLE(Maximum Likelihood Estimation)

# ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 어떤 모수 $\theta$ 로 결정되는 확률변수들의 모임 $D_\theta=(X_1, X_2,\cdots, X_n)$ 이 있고 $D_\theta$의 확률밀도함수나 확률질량함수가 $f(x)$이고, 그 확률변수들에서 각각 값 $X_1, X_2,\cdots, X_n$을 얻었을 경우 Likelihood $\boldsymbol{L}(\theta)$는 다음과 같다.

$$\boldsymbol{L(\theta) = f_\theta(x_1, x_2, \cdots, x_n)}$$

✓ 여기에서 Likelihood를 최대로 만드는 $\theta$ 는

$$\boldsymbol{\hat{\theta} = argmax_\theta L(\theta)}$$ 가 된다.

Unit 03 | MLE(Maximum Likelihood Estimation)

# ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 이 때 $X_1, X_2, \cdots, X_n$ 이 모두 독립적이고 같은 확률분포를 가지고 있다면 $\boldsymbol{L}$ 은 다음과 같이 표현이 가능하다

$$\boldsymbol{L(\theta)} = \prod_i^n \boldsymbol{f_\theta(x_i)}$$

✓ 또한, 로그함수는 단조 증가하므로, $\boldsymbol{L}$ 에 로그를 씌운 값의 최댓값은 원래 값 $\hat{\boldsymbol{\theta}}$와 같고, 이 경우 계산이 비교적 간단해진다.

$$\boldsymbol{L^*(\theta)} = \log(\boldsymbol{L(\theta)}) = \sum_i^n \log \boldsymbol{f_\theta(x_i)}$$

## ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 예를 들어 투빅스 10기 11기 중에서 한명을 표본으로 추출하는데 추출된 사람이 남자인지 여자인지를 알려고 한다고 하면, 이 때 표본 확률변수가 갖는 확률분포는 베르누이 분포

$$f(X) = p^X(1-p)^{1-X}$$

✓ 그러면 총 n명에 대해 추출했을 때의 likelihood는 다음과 같이 정해진다.

$$L(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n | p) = \prod_{i=1}^{n} p^{x_i} (1-p)^{1-x_i} \qquad (a)$$

# ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 가령 10명의 사람을 추출했는데 1번부터 10번 사람까지의 성별이 각각 {남, 여, 남, 남, 여, 여, 남, 남, 여, 남} 이라고 해보자.

✓ 남자라면 $\boldsymbol{X_i = 0}$이라고 하고 여자라면 $\boldsymbol{X_i = 1}$이라고 결정한다고 했을 때, 현 상태에서 $\boldsymbol{x_i (i = 1, 2, \dots, 10)}$ 은 {0, 1, 0, 0, 1, 1, 0, 0, 1, 0} 이라고 할 수 있다.

그러면 식 (a)는 다음과 같을 것이다.

$$\boldsymbol{L(X_1 = 0, X_2 = 1, \dots, X_{10} = 0 | p) = \prod_{i=1}^{10} p^{x_i} (1-p)^{1-x_i} \dots}$$

# ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 그렇다고 해서 우리가 p(표본이 여성일 확률)를 알수 있냐면 그렇진 못하다. 즉, 우리가 정작 알아야 하는 것은 이 p를 어떻게 결정할 것인가이다.

✓ 이 p를 잘 결정하려면 최대한 모집단의 p와 비슷할수록 잘 결정한 것이라고 할 수 있다. 최대우도법에서의 기본 가정은 지금 이러한 상황은 랜덤하게 부여된 것인데,

✓ **이 상황이 나온 것은 이렇게 나올 가능성이 가장 높았기 때문일 것이라는 것이다.**

✓ 따라서, '이 상황'을 설명하는 확률 $f$를 최대화 할 수 있는 모수 $p$를 찾는 것이 최대우도법이 시행하고자 하는 바이다.

Unit 03 | MLE(Maximum Likelihood Estimation)

## ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 예시에서 제기된 문제를 계속 풀어보도록 하자. 그런데, 식 (a)에서 함수 $f$를 $p$에 대해 편미분 하려면 쉽지 않다.

✓ 우리는 여기서 로그 함수의 단조증가 성질을 활용하여 $L^* = \log(L)$라는 보조 방정식을 도입하도록 하자.

그러면 $L^*$는 다음과 같다.

$$\boldsymbol{L^* = log(L)} = \sum_{i=1}^{n} \log(\boldsymbol{p^{x_i}(1-p)^{1-x_i}}) = \sum_{i=1}^{n} \{\boldsymbol{x_i} \log(\boldsymbol{p}) + (\boldsymbol{1 - x_i}) \log(\boldsymbol{1-p})\}$$

Unit 03 | MLE(Maximum Likelihood Estimation)

## ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 이제 보조방정식 $L^*$을 $p$에 대해 편미분 하는 것이 쉬워진다.

✓ 그런 다음 $L^*$ 의 $p$ 에 대한 편미분이 0이 되는 $p$ 를 찾으면 최대우도를 만족하는 모수 $p$ 를 추정할 수 있다.

$$L^* = \frac{\sum_{i=1}^{n} x_i}{p} - \frac{\sum_{i=1}^{n}(1-x_i)}{1-p}$$

$$= \frac{n\bar{X}}{p} - \frac{n(1-\bar{X})}{1-p} = 0$$

따라서

$$\Rightarrow \frac{n\bar{X}}{p} = \frac{n(1-\bar{X})}{1-p}$$

$$\Rightarrow (1-p)\bar{X} = (1-\bar{X})p$$

$$\Rightarrow \bar{X} - p\bar{X} = p - \bar{X}p$$

$$\therefore p = \bar{X}$$

# ◆ Logistic Regression

- Maximum Likelihood Estimation

✓ 따라서, $p = \overline{X}$로 모수 $p$를 추정하는 것이 적절하다는 것을 알 수 있다.

✓ 생각해보면 자연스러운 것이 모비율 추정 시 현재 모여있는 사람의 성비를 가지고 모비율을 추정할 수 밖에 없고, 아마 그런 모비율이 있었기 때문에 현재 상태가 만들어 진 것은 아닐까? 라고 추정하는 것은 자연스럽다.

Contents

## ◆ Optimizer

optimizing의 ex) 최소제곱법 OLS로 가기

- Optimizer
  - ✓ Machine learning 문제를 풀다보면 Objective function을 만들고 그 objective function을 optimize 해야하는 경우가 매우 빈번하게 발생함.
  - ✓ 간단히 생각해서 loss function을 minimize하는 것도 optimization이다. 그렇다면 그런 optimization은 도대체 어떻게 해야하는 것일까.
  - ✓ 여러가지 방법이 있지만, 이번 시간에서는 간단한 optimization이라는 것에 대한 컨셉을 다루고, 그 중 특수 케이스인 convex optimization에 대해 다룸

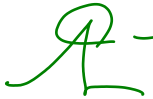
## ◆ Optimizer

- Optimizer
  - ✓ 대부분의 Optimization은 아래와 같은 식으로 표현할 수 있을 것이다.

$$\min f(x)\ s.t.\ g(x) = c$$

  - ✓ 여기에서 $f(x)$, $g(x)$는 함수이다. 어떤 함수의 optimum point, 즉 그것이 최소이거나 혹은 최대인 지점을 찾는 과정을 optimization이라고 한다고 생각하면 간단
  - ✓ 엄청 간단하게 생각해보면 $f(x)$는 loss function이고, $g(x)$는 일종의 제약조건으로 생각하면 됨

optimize 해야하는 ~

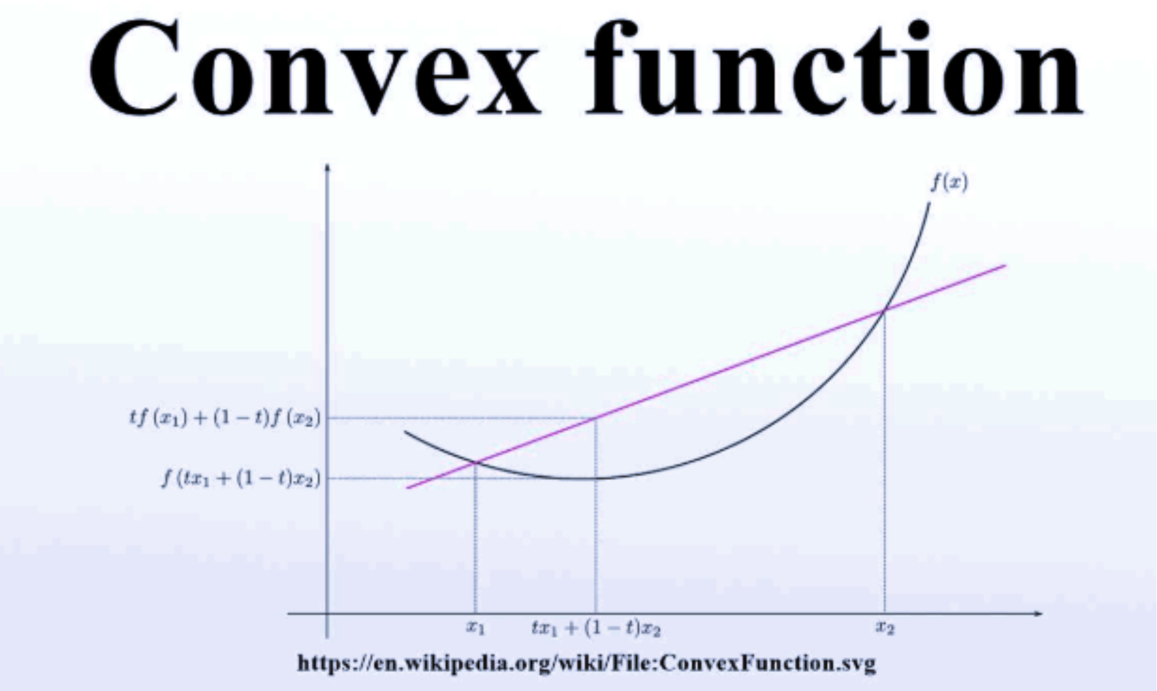
미분이 안되는 함수
or
 이럴때

## ◆ Optimizer

- Optimizer
  - ✓ Machine learning 문제를 풀다보면 이렇게 optimization을 해야하는 일이 아주 빈번하게 발생하는데, 안타깝게도 항상 이런 function들의 optimum point를 찾을 수 있는 것은 아님
  - ✓ 가장 간단하게 생각했을 때 이런 point를 찾는 방법은 미분을 하고 그 값이 0이 되는 지점을 찾는 것인데, 안타깝게도 미분 자체가 되지 않는 함수가 존재할 수도 있다
  - ✓ 또한 미분값이 0이라고 해서 반드시 극점인 것은 아니기 때문이다. (saddle point를 생각해보자.)

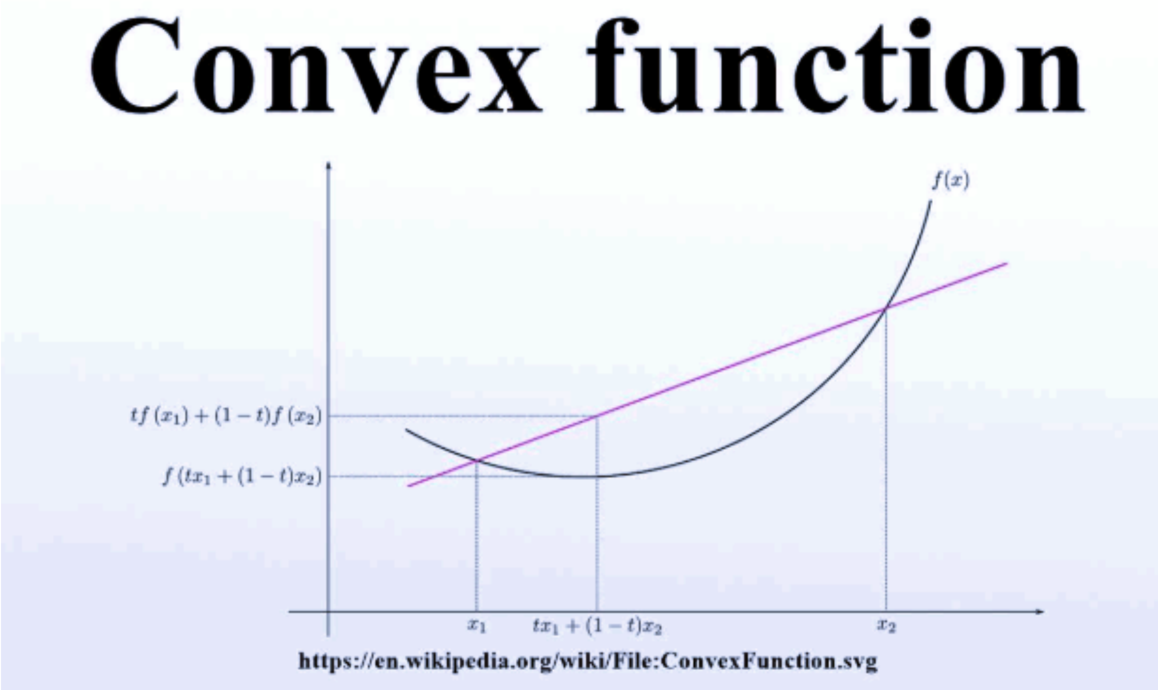
# ◆ Optimizer

- Optimizer
  - ✓ 따라서 대부분의 경우에 이런 방법으로 극점을 구하는 것은 불가능하며, 매우매우 특수한 일부 경우에 대해서 완전한 optimum을 찾는 것이 알려져 있다. 그리고 그 경우가 바로 convex optimization이다.



# ◆ Optimizer

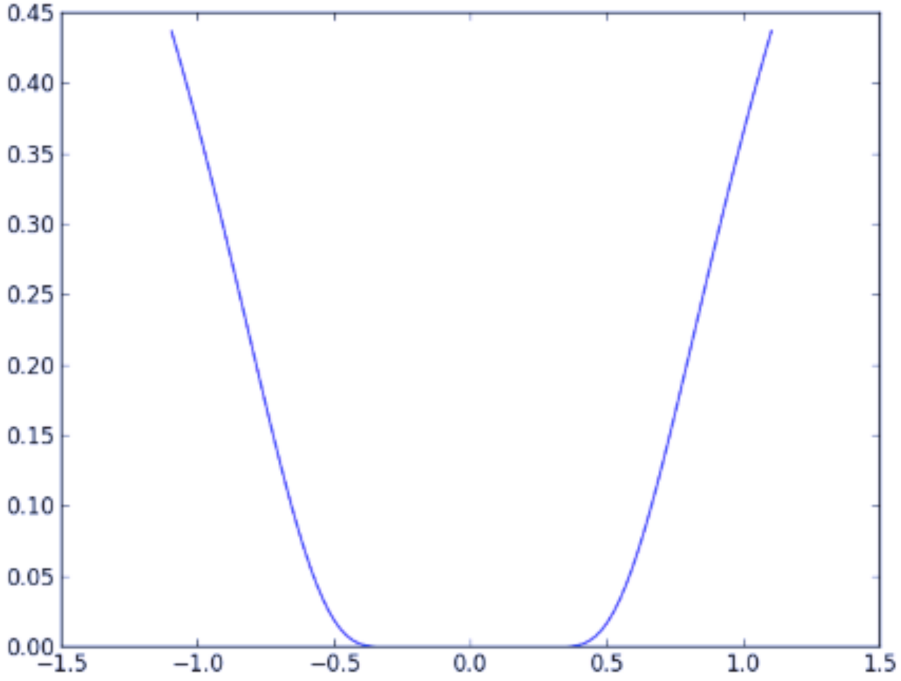
- Optimizer




$$f(\lambda x_1 + (1 - \lambda x_2) \le \lambda f(x_1) + (1 - \lambda) f(x_2)\ for\ \forall x_1,\ x_2 \in X,\ \forall \lambda \in [0,1]$$

# ◆ Optimizer

- Optimizer
  - ✓ Convex function이 좋은 이유는 반드시 optimal value가 하나 밖에 존재하지 않는다는 것이다. 여기에서 optimal point가 하나라고 얘기하지 않은 이유는 아래와 같은 예가 있기 때문이다.



# ◆ Optimizer

- Optimizer

✓ 이 함수는 optimal value는 unique하게 존재하지만, 그 값을 가지는 point가 unique하지는 않다.

✓ 따라서 unique한 optimal point를 찾기 위해서는 하나의 조건이 더 필요한데, 바로 strictly convex라는 조건이다.

$$f(\lambda x_1 + (1 - \lambda x_2) < \lambda f(x_1) + (1 - \lambda) f(x_2) for\, \forall x_1, x_2 \in X, \forall \lambda \in [0,1]$$

✓ 이 조건은 위의 식에서 = 이 빠진 형태이다. 즉 ≤ 가 < 으로 바뀌는 것이다.

✓ 이런 strictly convex function에 대해서 optimal point가 unique하게 존재한다는 것을 증명할 수 있으며, 증명과정은 크게 어렵지 않으니 링크 등을 참고하면 될 것 같다.

✓ 아무튼 strictly convex function은 minimum point가 unique하게 존재하기 때문에, 이런 convex function에 대해서 우리는 어떤 optimization algorithm을 design할 수 있다.

# ◆ Optimizer

- Optimizer
  - ✓ 이런 convex optimization의 subset으로 linear programming, quadratic programming, semidefinite programming 등이 존재한다.
  - ✓ 여기서는 그런 특수한 경우는 다루지 않고, 일반적인 convex optimization에서 사용할 수 있는 알고리즘들을 다룸.
  - ✓ Gradient descent method, Newton method and Lagrange multiplier 등이 있다.

# ◆ Optimizer

- Gradient Descent Algorithm
  - ✓ 최적해에 가까워지는 방향으로 조금씩 움직이면서 최적의 해를 찾는 최적화 알고리즘
  - ✓ 산 정상 위에 우리가 서있다고 가정
  - ✓ 우리가 알 수 있는 정보는 내 위치와 내 주변 위치들의 높이 차이밖에 없다고 가정
  - ✓ 만약 내가 산의 가장 낮은 위치로 내려가야하는 상황이라면 어떻게 내려가면 낮은 위치에 도달할 수 있을까?
  - ✓ 가장 간단한 방법은 가장 기울기가 가파른 방향을 골라서 내려가는 것
  - ✓ 그러다보면 언젠가는 기울기가 0이 되는 지점에 도달하게 될 것이고, 그 지점이 주변에서는 가장 낮은 지점이 될 것이다.

Unit 04 | Optimizer

# ◆ Optimizer

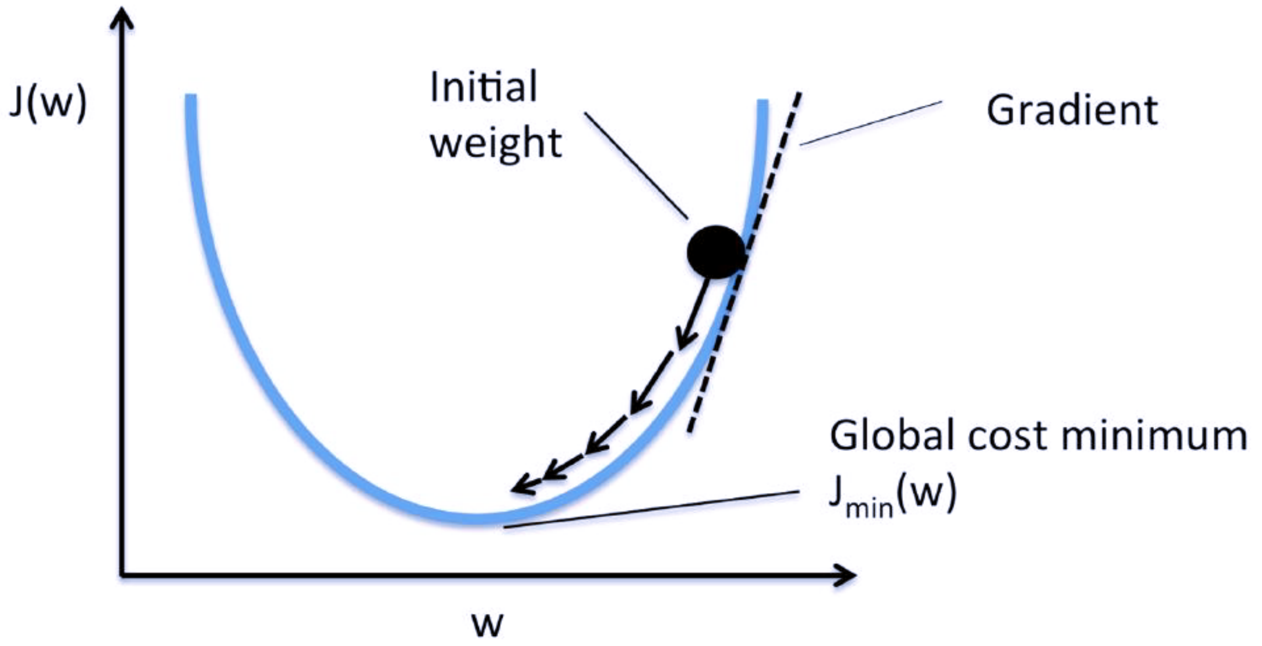
- Gradient Descent Algorithm

  ✓ 그러다보면 언젠가는 기울기가 0이 되는 지점에 도달하게 될 것이고, 그 지점이 주변에서는 가장 낮은 지점이 될 것이다.

  ✓ 만약 산의 높이가 convex function이라면, 즉 가장 낮은 지점이 unique하다면, 그렇게 도달한 지점이 우리가 원했던 가장 optimal한 지점이라는 것을 알 수 있다.

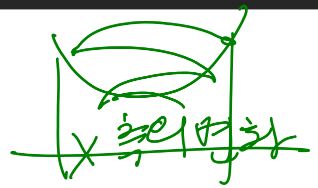
# ◆ Optimizer

- Gradient Descent Algorithm
  - ✓ 파란선 : 가중치에 대한 Loss function
  - ✓ 검은점 : 현재 해의 위치
  - ✓ 화살표 : Loss funstion을 최적화하기 위해 가중치가 이동해야 하는 방향



## ◆ Optimizer

- Gradient Descent Algorithm – [링크]

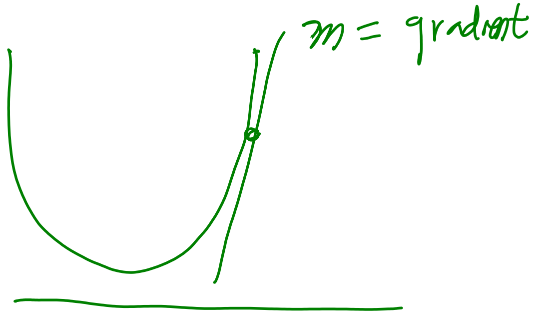



$$w \leftarrow w - \alpha \frac{\partial}{\partial w} J(w)$$

방향알려줌.

### ✓ $\frac{\partial}{\partial w} J(w)$: gradient

- 가중치에 대한 목적함수의 미분값
- 미분 값의 부호는 최적해로 가기 위한 방향과 반대 값을 가짐




### ✓ $\alpha$ : Learning Rate

- 움직이는 정도를 조정하는 계수로 클수록 빠르게 학습하지만 최적해로 수렴하지 못할 수도 있음

Contents

## Unit 05 | Tutorial & Assignment

# ◆ Tutorial

- Logistic Regression 맛보기 with Python(Logistic_Regression_01.ipynb)

# ◆ Assignment

얘네는 python으로 하기! 함수 구현해보자.

1. 튜토리얼 파일(Logistic_Regression_01.ipynb)을 수정하세요.

   * confusion matrix와 ROC curve를 활용하여 cut-off만 조정
   * 새로운 결과값 도출 및 의미해석

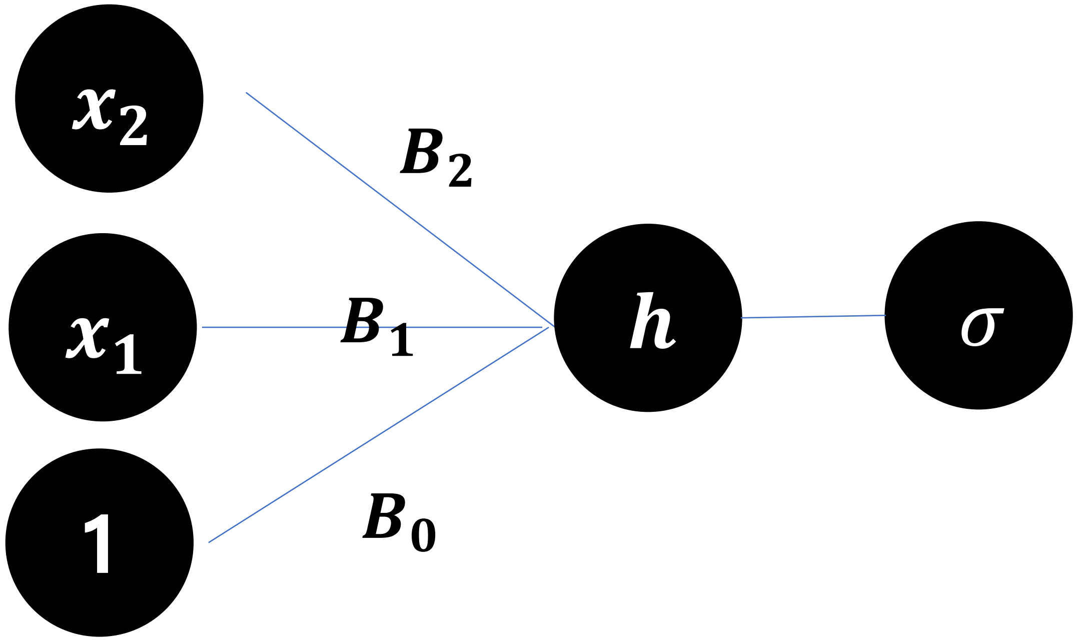
2. MLE를 Gradient Descent의 Loss function으로 사용하여 최적화하기

   * Logistic Regression의 회귀계수를 추정하는 MLE와 Optimizer 중에 하나인 GD를 배웠습니다.
   * MLE를 최적화 해보세요.(MLE에 마이너스를 곱하면 Convex 함수가 됩니다)

# ◆ Assignment 2

- Gradient Descent Algorithm

✓ Logistic Regression에서의 회귀 계수 추정
- Gradient Descent Algorithm을 이용한 방법
- 설명변수가 두개인 경우



$$h = \sum_{i=1}^{2} B_i x_i + B_0$$

$$\sigma = \frac{1}{1 + exp(-h)}$$

$$J = -\frac{1}{n} log(L) \text{ , } log(L)\text{은 } MLE$$

References

## ◆ Reference


- 투빅스 7기 최희정님 자료, 투빅스 7기 전종섭님 자료
- Ratsgo's blog
- 조대협님 blog
- Quora
- 공돌이의 수학정리노트
- Lee, I. Optimization. Retrieved from http://issactoast.com
- SanghyukChun's Blog – Convex Optimization
- Kaggle - 타이타닉

# Q & A