

Statistics II

for Machine Learning

Chap. 3: 회귀분석 & 로지스틱 함수

Oh, Hyung Sool

Regression Analysis

- "회귀(regression)": 극단값이 다음 세대에서 평균 또는 평균에 더 가까워지는 경향을 설명하기 위해 사용



Regression Analysis

회귀분석의 정의

- 독립변수가 종속변수에 미치는 영향력의 크기를 파악하여 독립변수의 특정한 값에 대응하는 종속변수값을 예측하는 선형모형을 산출하는 방법

회귀분석을 필요로 하는 문제의 예

- 매출액은 광고횟수에 따라 어떻게 변하는가?
- 제조환경을 개선함에 따라 생산량은 어느 정도 증가할 것인가?
- 담배판매량과 폐암환자수와의 관계는 어떠한가?

변 수

- 광고횟수
- 제조환경
- 담배 판매량

독립변수

- 매출액
- 생산량
- 폐암환자수

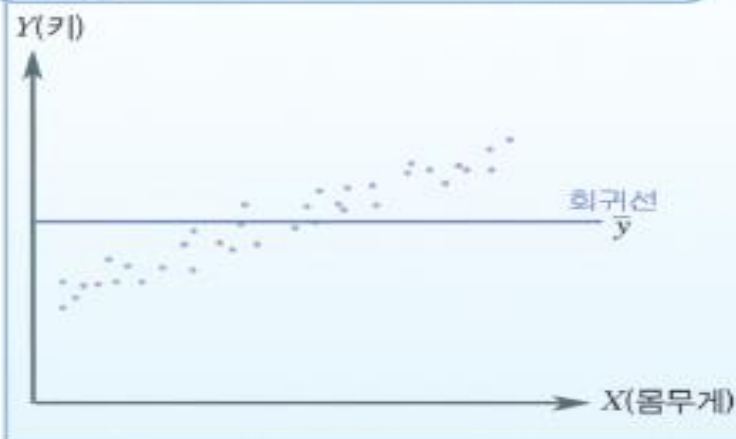
종속변수

- 독립변수와 종속변수의 관계를 파악함(예측, 설명)

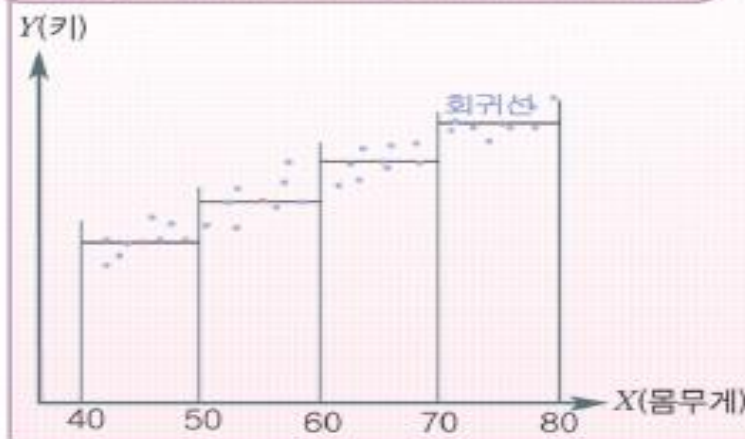
회귀분석

Regression Analysis

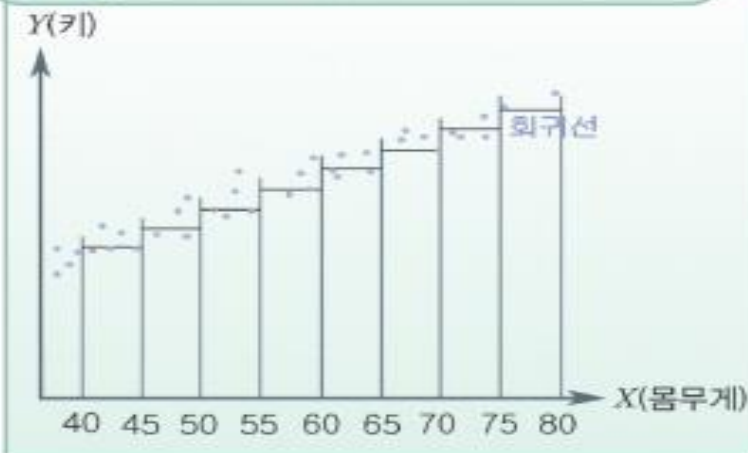
1 몸무게(X)를 구분하지 않음



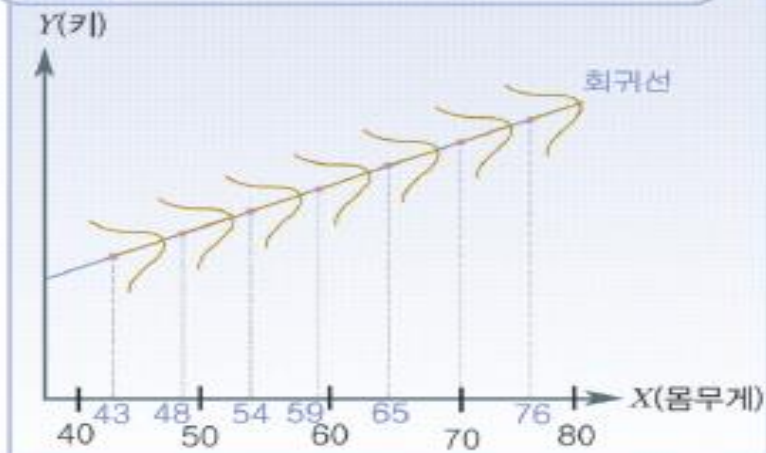
2 몸무게(X)를 10kg 단위로 구분함



3 몸무게(X)를 5kg 단위로 구분함



4 몸무게(X)를 모든 정수 단위로 구분함



The figure consists of three distinct data visualizations arranged horizontally. On the left is a bar chart with approximately 15 vertical bars of varying heights. In the center is a line graph with a jagged, fluctuating line. On the right is a scatter plot showing a series of data points connected by lines, forming a wave-like pattern. The entire figure is set against a blue background with a faint grid and a large, semi-transparent 'X' shape.

응답자	변수 X 몸무게	변수 Y 키
1	72	176
2	72	172
3	70	182
4	43	160
5	48	163
6	54	165
7	51	168
8	52	163
9	73	182
10	45	148
11	60	170
12	62	166
13	64	172
14	47	160
15	51	163
16	74	170
17	88	182
18	64	174
19	56	164
20	56	160
평균	60.1	168

회귀선: $\hat{y}_i = \beta_0 + \beta_1 x_i$

$\bar{x} = 60.10\text{kg}$

$\bar{y} = 168\text{cm}$

관측치

회귀선

$\epsilon = \text{잔차}$

β_0

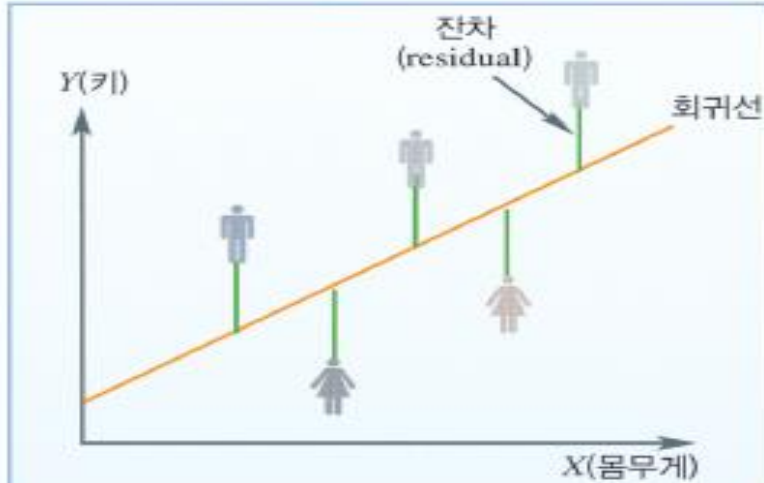
β_1

$Y(\text{키})$

$X(\text{몸무게})$

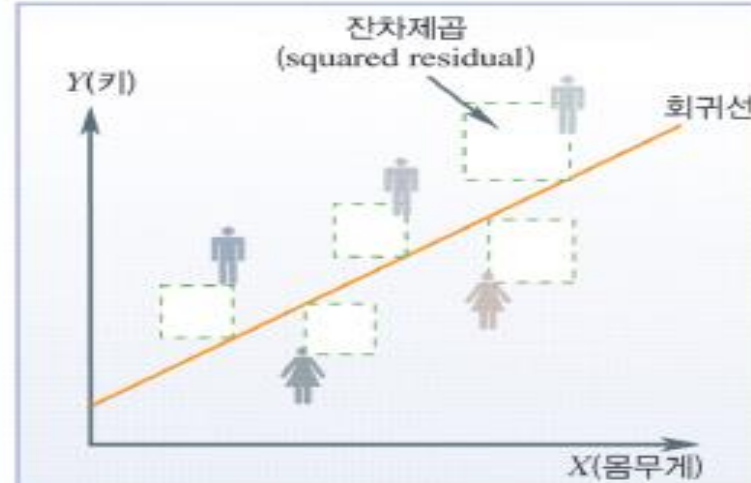
- 몸무게와 키에 관한 자료를 이용하여 선형 회귀식을 도출함
- 회귀식을 이용하여 몸무게에 따른 키를 예측함
- 도출된 회귀선은 두 변수의 평균이 교차하는 점을 통과함

Regression Analysis



- 가상의 회귀선과 관측치들 간의 거리, 즉 잔차의 절대값을 모든 관측치에 대하여 구한 다음 그 합을 최소화하는 직선이 가장 적합한 회귀선임
- 일반적인 잔차는 양수나 음수가 될 수 있으므로 단순한 잔차의 합은 회귀식을 구하는 기준으로 적합하지 않음
- 이를 해소하기 위해 잔차에 절대값을 취하여 합하는 방법이 있으나, 절대값은 컴퓨터를 이용한 계산이 어려움

$$\text{Min} \sum_{i=1}^n |e_i| = \text{Min} \sum_{i=1}^n |y_i - \hat{y}_i|$$



- 관측치와 회귀선 간의 잔차를 제곱한 값을 모든 관측치에 대하여 구한 다음 그 합을 최소화하면 최적의 회귀선을 도출할 수 있음
- 그림에서 사각형 면적의 합이 가장 작은 회귀선이 최적의 회귀선임

$$\begin{aligned} \text{Min} \sum_{i=1}^n e_i^2 &= \text{Min} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ &= \text{Min} \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]^2 \end{aligned}$$

Regression Analysis

최소자승법

- 관측치(y_i)와 회귀선(\hat{y}_i)과의 거리인 잔차(e_i)제곱의 합을 최소화하는 직선식을 찾는 방법
- 잔차의 합($\sum_{i=1}^n e_i$)을 최소화하는 것이 아니고, 잔차제곱의 합($\sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$)을 최소화하는 직선식을 구하는 방법

1 잔차제곱합

$$E = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]^2$$

2 잔차제곱합의 최소화

$$\text{Min}[E] = \text{Min} \left[\sum_{i=1}^n e_i^2 \right] = \text{Min} \left[\sum_{i=1}^n (y_i - \hat{y}_i)^2 \right] = \text{Min} \left[\sum_{i=1}^n \{y_i - (\beta_0 + \beta_1 x_i)\}^2 \right]$$

3 절편과 기울기의 편미분

- 절편(β_0)과 기울기(β_1)로 편미분한 값을 0으로 하는 등식을 풀면 직선식의 기울기와 상수항을 구할 수 있음

$$\frac{\partial E}{\partial \beta_0} = 2 \sum_{i=1}^n (-1) [y_i - (\beta_0 + \beta_1 x_i)] = 0$$

$$\frac{\partial E}{\partial \beta_1} = 2 \sum_{i=1}^n (-x_i) [y_i - (\beta_0 + \beta_1 x_i)] = 0$$

4 절편과 기울기값 도출

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Regression Analysis

◆ 편차, 편차곱, 편차제곱의 도출

ID	변수X 몸무게	변수Y 키	편차X ($x_i - \bar{x}$)	편차Y ($y_i - \bar{y}$)	편차곱 ($(x_i - \bar{x})(y_i - \bar{y})$)	편차제곱 ($(x_i - \bar{x})^2$)
1	72	176	11.9	8.0	95.2	141.6
2	72	172	11.9	4.0	47.6	141.6
3	70	182	9.9	14.0	138.6	98.0
4	43	160	-17.1	-8.0	136.8	292.4
5	48	163	-12.1	-5.0	60.5	146.4
6	54	165	-6.1	-3.0	18.3	37.2
7	51	168	-9.1	0.0	0.0	82.8
8	52	163	-8.1	-5.0	40.5	65.6
9	73	182	12.9	14.0	180.6	166.4
10	45	148	-15.1	-20.0	302.0	228.0
11	60	170	-0.1	2.0	-0.2	0.0
12	62	166	1.9	-2.0	-3.8	3.6
13	64	172	3.9	4.0	15.6	15.2
14	47	160	-13.1	-8.0	104.8	171.6
15	51	163	-9.1	-5.0	45.5	82.8
16	74	170	13.9	2.0	27.8	193.2
17	88	182	27.9	14.0	390.6	778.4
18	64	174	3.9	6.0	23.4	15.2
19	56	164	-4.1	-4.0	16.4	16.8
20	56	160	-4.1	-8.0	32.8	16.8
총합	1202	3360	0	0	1673	2693.8
평균	60.1	168				

◆ 회귀계수 산출

$$\bar{x} = 60.10 \quad \bar{y} = 168$$

$$\sum (x_i - \bar{x})(y_i - \bar{y}) = 1673$$

$$\sum (x_i - \bar{x})^2 = 2693.80$$

■ 회귀선의 기울기

$$\begin{aligned} \beta_1 &= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \\ &= \frac{1673}{2693.80} = 0.621 \end{aligned}$$

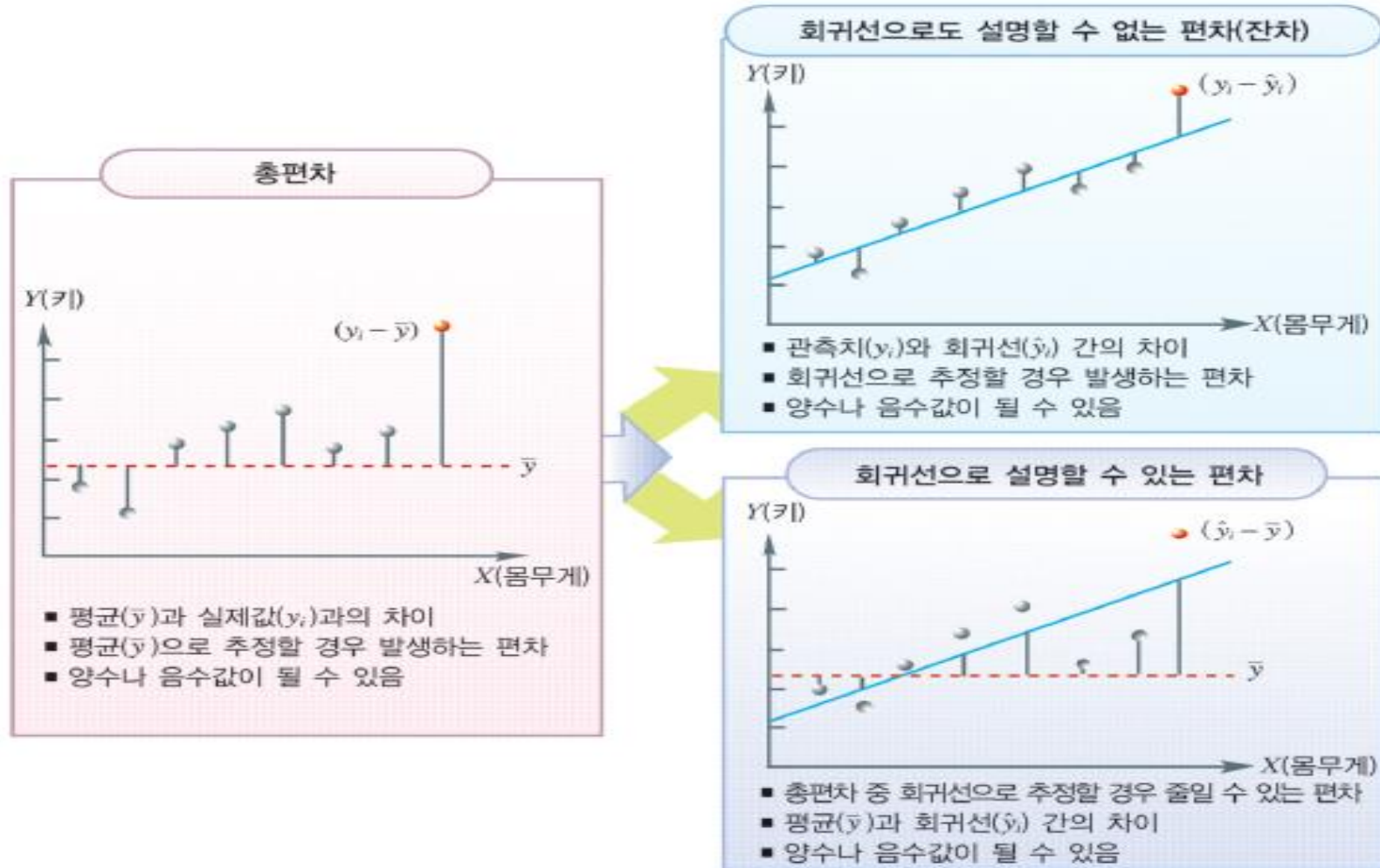
■ 회귀선의 Y절편

$$\begin{aligned} \beta_0 &= \bar{y} - \beta_1 \bar{x} = 168 - (0.621 \times 60.1) \\ &= 130.678 \end{aligned}$$

◆ 회귀선의 도출과 활용

- 최종 회귀선: $\hat{y}_i = 130.678 + 0.621x_i$
- 몸무게(x_i)가 67kg의 학생의 키는 172.282cm로 추정할 수 있음
 $\hat{y}_i = 130.678 + 0.621 \times 67 = 172.285\text{cm}$

Regression Analysis

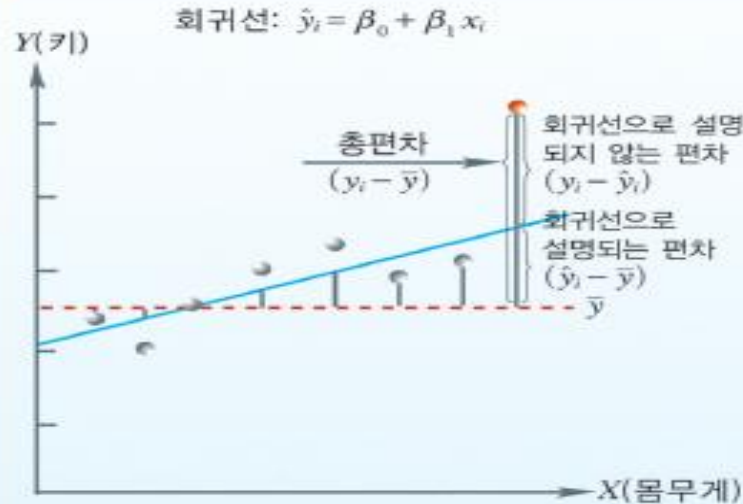


● 총편차($y_i - \bar{y}$) = 회귀선으로 설명되지 않는 편차($y_i - \hat{y}_i$) + 회귀선으로 설명되는 편차($\hat{y}_i - \bar{y}$)

Regression Analysis

결정계수

- 추정된 회귀선(\hat{y}_i)이 실제값(y_i)과 평균(\bar{y}) 사이의 편차를 얼마나 줄여주는가를 나타내는 지수
- 일반적으로 R^2 으로 나타냄



편차는 양수나 음수값을 가질 수 있으므로 편차를 제곱한 값을 활용하여 분석함

$$(y_i - \bar{y}) = (y_i - \hat{y}_i) + (\hat{y}_i - \bar{y})$$

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

$$SST = SSE + SSR$$

$$(\text{총제곱합}) = \boxed{\text{회귀선으로 설명되지 않는 제곱합}} + \boxed{\text{회귀선으로 설명되는 제곱합}}$$

$$R^2 = \frac{\text{회귀선에 의해 설명되는 제곱합}}{\text{총제곱합}}$$

$$= 1 - \frac{\text{회귀선에 의해 설명되지 않는 제곱합}}{\text{총제곱합}}$$

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}, \quad (0 \leq R^2 \leq 1)$$

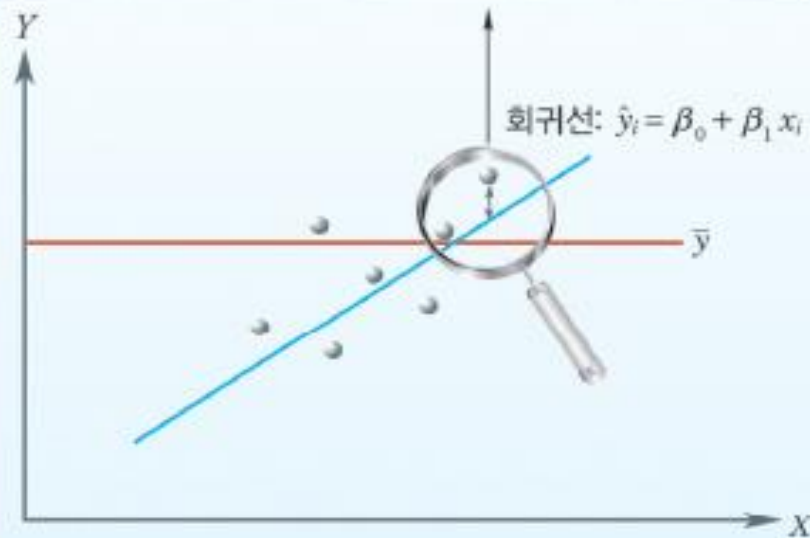
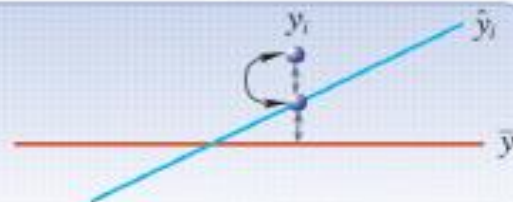
- 결정계수(R^2)는 총제곱합 중에서 회귀선으로 설명되는 제곱합의 비율임
- 결정계수(R^2)가 1에 가까울수록 회귀선은 설명력이 높은 바람직한 회귀선이 됨

Regression Analysis

추정값의 표준오차

- 실제 관측치(y_i)와 추정된 회귀선의 예측값(\hat{y}_i)과의 차이, 즉 오차 혹은 잔차(e_i)의 표준편차를 말함
- 잔차는 음의 값과 양의 값이 있기 때문에 서로 상충되지 않도록 잔차의 합($\sum_{i=1}^n e_i$) 대신 잔차제곱합($\sum_{i=1}^n e_i^2$)을 사용함

추정값의 오차
= 잔차(e_i)
= ($y_i - \hat{y}_i$)



잔차제곱의 합(sum of squared error)

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]^2$$

- \hat{y}_i 을 구하는 과정에서 β_0 와 β_1 을 사용하기 때문에 잔차제곱합(SSE)은 2개의 자유도를 잃게 되어 잔차제곱합(SSE)의 자유도는 $n-2$ 가 됨
- 잔차제곱합(SSE)을 자유도인 $n-2$ 로 나누어 주면 잔차평균제곱(MSE)이 됨

잔차평균제곱(mean of squared error)

$$MSE = \frac{SSE}{n-2} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}$$

Regression Analysis

단순회귀분석의 분산분석표

- 객관적으로 도출된 회귀식이 통계적으로 유의한가를 평가하는 방법
- 회귀선의 설명력(R^2)이 아무리 높아도 통계적으로 유의하지 않으면 일반화하여 사용하기 어려움
- 분산분석에서와 같은 방법으로 회귀식의 통계적 유의성을 검정함

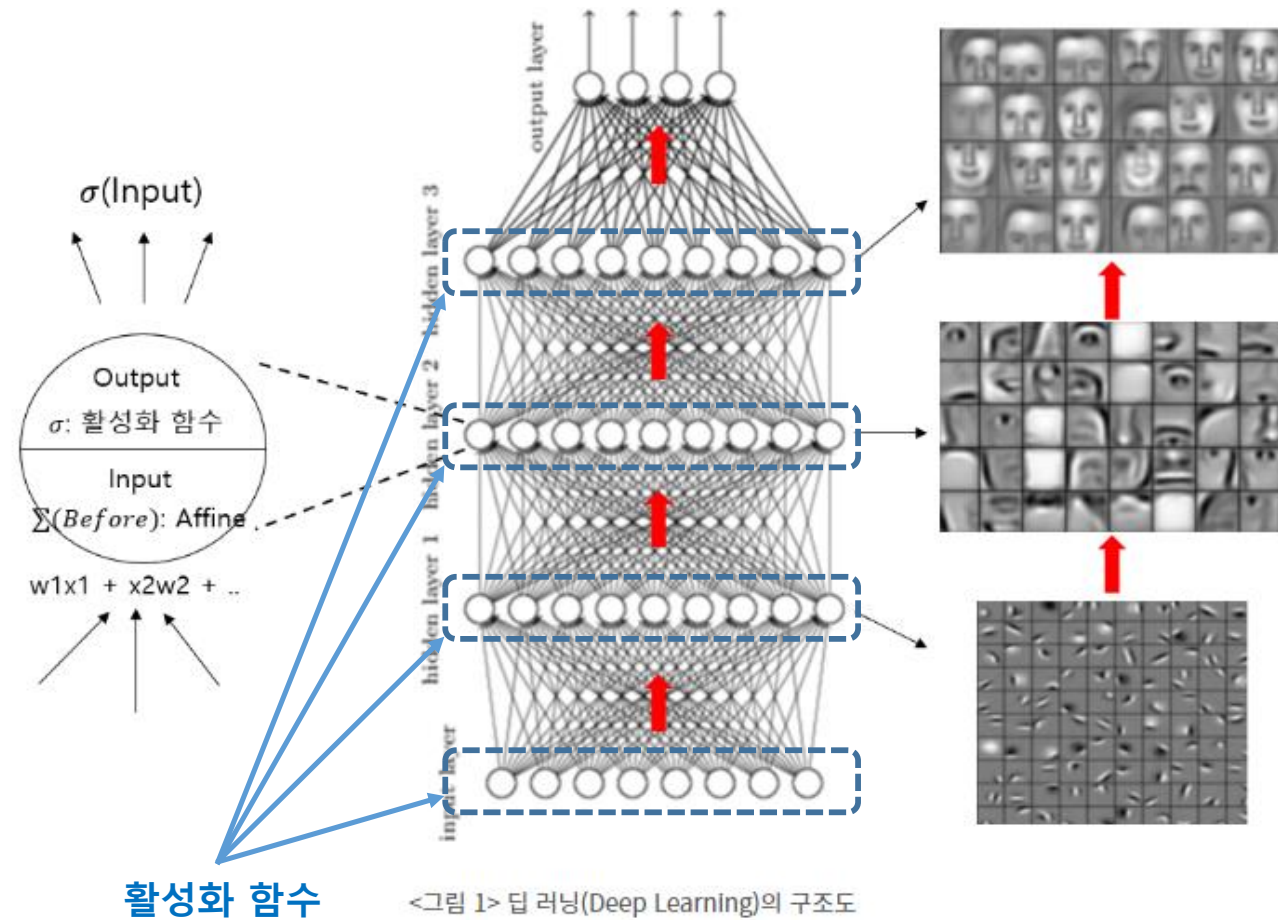
원 천	제곱합(SS)	자유도(df)	평균제곱(MS)	검정통계량 F
회 귀	$SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$	$(k+1)-1$	$MSR = \frac{SSR}{(k+1)-1}$	$\frac{MSR}{MSE}$
잔 차	$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$	$n-(k+1)$	$MSE = \frac{SSE}{n-(k+1)}$	
총(합계)	$SST = \sum_{i=1}^n (y_i - \bar{y})^2$	$n-1$	여기서, n : 관측치의 수 k : 독립변수의 수	

분산분석표

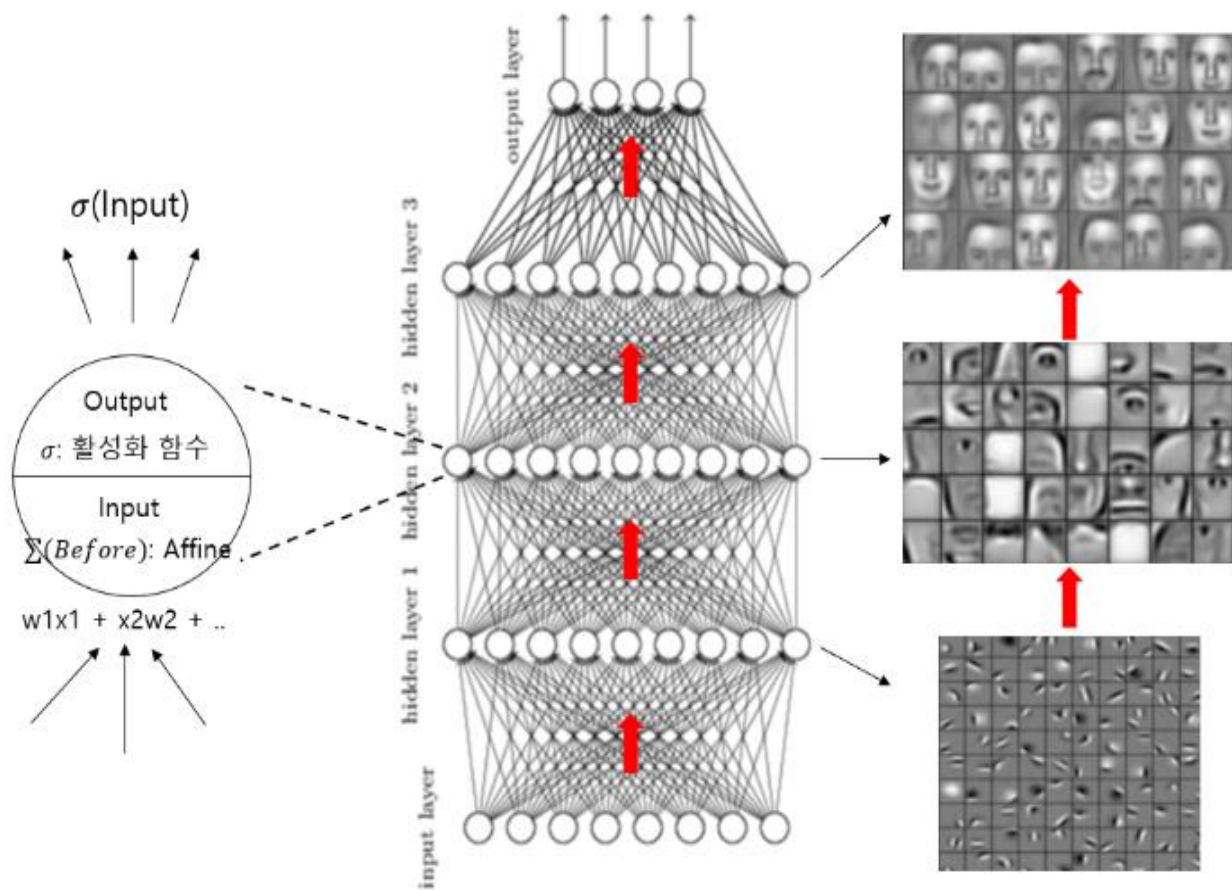
원 천	제곱합(SS)	자유도(df)	평균제곱(MS)	검정통계량 F	유의확률(p-value)
회 귀	1039.026	1	1039.026	48.581	.000
잔 차	384.974	18	21.387		
총(합계)	1424.000	19			

Regression Analysis & Deep Learning

- 딥러닝에서는 오차를 최소화하도록 **파라미터의 가중치(Weights)**를 **갱신**하는데, 이 개념이 회귀분석과 상당히 유사하다.
- 학습이 반복되면서 가중치가 계속 바뀌게 되는데, 이 과정이 가중치(w_1, w_2, \dots)로 표현된 **회귀 방정식($w_1x_1 + w_2x_2 + \dots$)**의 **최적화** 과정에서 이루어진다.
- 인공 신경망(ANN)은 무수히 많은 가중치들을 갖기 때문에 신경망(ANN)은 **무수히 많은 회귀식**으로 이루어졌다고 볼 수 있다.



Regression Analysis & Deep Learning

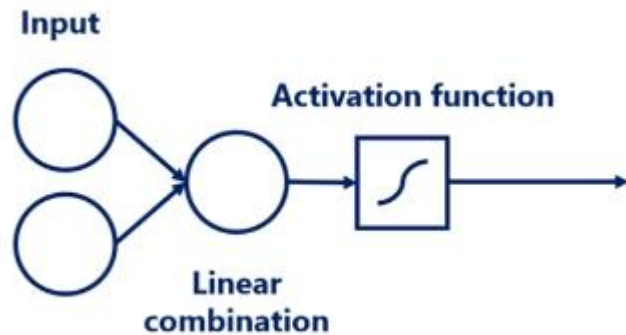



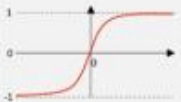
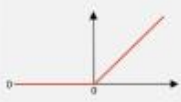
<그림 1> 딥 러닝(Deep Learning)의 구조도

- 사람 얼굴을 인식하는 과제를 인공 신경망이 수행해야 한다고 가정하자. 심층 신경망은 **복잡한 과제**(사람 얼굴 인식)를 **작은 업무의 조합**(사람 얼굴 특징)으로 구분하고 이를 각 Layer가 분할하여 담당하도록 한다. 이러한 기법을 **앙상블(ensemble)**이라고 합니다.
- 아래의 작은 네모칸에는 신경망 모델이 찾아낸 사람 얼굴의 특징을 가장 단순하게 점과 선들로 표현한 것이다. 첫 Layer는 가장 단순한 표현을 찾고, 과제의 남은 부분을 다음 Layer로 전달한다.
- 중간층의 작은 네모칸들에는 사람 얼굴의 특징들(눈, 코, 입)이 그려져 있다. 이전 Layer에서 전달받은 결과 값을 조합하여 보다 선명한 표현을 학습한다. 그리고 최종 Layer로 결과 값을 전달한다.

Activation Function

- 활성화 함수: 신경망에서 입력신호를 총합하여 출력신호로 변환하는 역할을 한다. 활성화 함수는 **비선형 함수**를 사용
- 활성화 함수가 비선형 함수를 사용하기 때문에 신경망이 더 **복잡한 패턴**을 학습할 수 있도록 **표현력**을 높여준다.
- 딥러닝 모델을 학습할 때, 활성화 함수의 종류에 따라 **모델 성능**이 크게 달라질 수 있기 때문에 활성화 함수를 적절하게 선택하는 것이 중요



Name	Formula	Derivative	Graph	Range
sigmoid (logistic function)	$\sigma(a) = \frac{1}{1+e^{-a}}$	$\frac{\partial \sigma(a)}{\partial a} = \sigma(a)(1 - \sigma(a))$		(0,1)
TanH (hyperbolic tangent)	$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$	$\frac{\partial \tanh(a)}{\partial a} = \frac{4}{(e^a + e^{-a})^2}$		(-1,1)
ReLu (rectified linear unit)	$\text{relu}(a) = \max(0, a)$	$\frac{\partial \text{relu}(a)}{\partial a} = \begin{cases} 0, & \text{if } a \leq 0 \\ 1, & \text{if } a > 0 \end{cases}$		(0,∞)
softmax	$\sigma_i(a) = \frac{e^{a_i}}{\sum_j e^{a_j}}$	$\frac{\partial \sigma_i(a)}{\partial a_j} = \sigma_i(a)(\delta_{ij} - \sigma_j(a))$ Where δ_{ij} is 1 if $i=j$, 0 otherwise	different every time	(0,1)

Activation Function: *sigmoid fun.*

- 신경망에 사용되는 회귀식을 **분류(classification)**에 활용하기 위해, 간단히 y 를 p 로 바꿔보자.

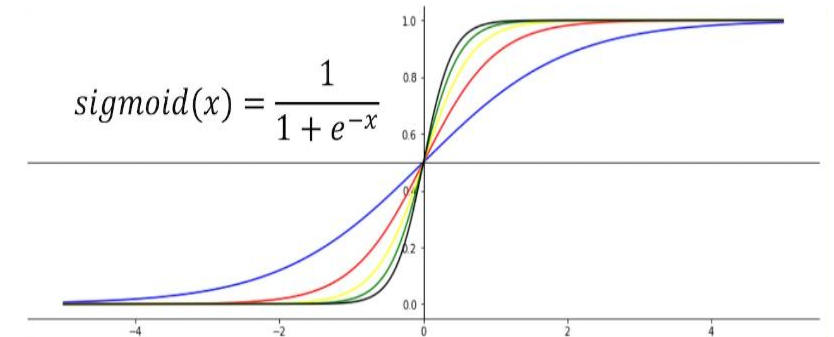
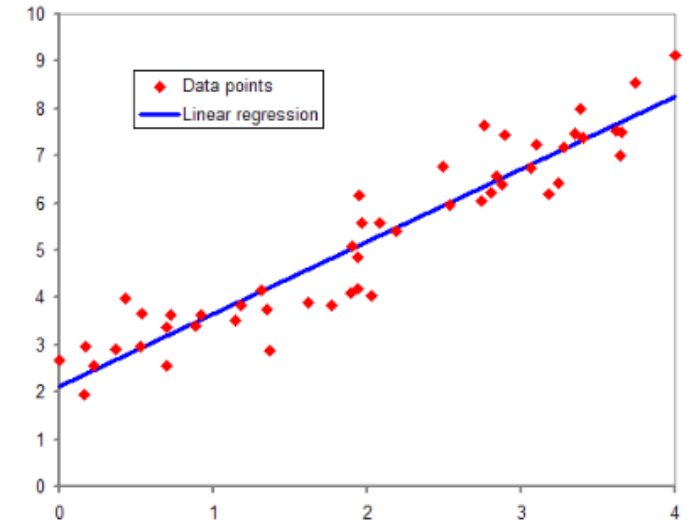
$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$



$$p = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$

- p 로 표현한 식을 분류 문제에 적용하기 위해서는 p 값이 제한되어 있지 않다는 문제점을 해결해야만 한다.
- 확률 값으로 만들기 위해 p 값을 0 ~ 1로 제한 시켜야 한다. 그래서 우리가 익히 알고 있는 **시그모이드 함수**를 적용한다.

$$f(x) = \frac{1}{1 + e^{-x}}$$



Odds & Logit

- 선형관계가 있는 실수의 입력 값들을 토대로 확률을 예측하는 회귀모델은 다음과 같다.

$$p = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$

- 여기서 문제 발생. 좌변의 확률 값 p 는 확률이므로 $[0,1]$ 값만 가능하지만, 우변의 선형 회귀식은 연속형 실수 공간이므로 $[-\infty, +\infty]$ 값을 갖는다.
- 이 때문에 좌변과 우변 간에 값의 범위가 다른 **미스 매치가 발생**
그래서, 우변의 확률 값을 $[-\infty, +\infty]$ 사이의 실수 값으로 변환해줄 수 있는 특별한 식의 필요성이 발생
이로 인해 등장하는 것이 **승산(Odds)의 개념**이다.
- 승산(Odds)는 실패 확률에 대한 성공 확률의 비이다. Odds=4 란 "성공확률이 실패확률보다 4배 높다" 라는 의미

$$\text{승산(Odds)} = \frac{p}{(1-p)} = \frac{\text{관심 있는 사건이 발생함}}{\text{관심 있는 사건 발생하지 않음}}$$

Odds & Logit

- 승산에 자연로그를 취하면 확률의 의미를 가지면서 $[-\infty, +\infty]$ 사이의 범위 값을 변환해줄 수 있는 함수를 얻게 됩니다.

$$\ln[\text{Odds}] = \ln\left[\frac{p}{(1-p)}\right]$$

- **로짓(Logit) 변환:** 어떤 사건이 벌어질 확률 p 가 $[0,1]$ 사이의 값일 때, 이를 $[-\infty, +\infty]$ 사이 실수 값으로 변환하는 과정을 **로짓(Logit) 변환**이라고 한다.
- 로짓변환으로 확률과 실수 값 사이의 선형 관계성을 찾는 로지스틱 회귀가 가능하게 된다.
(승산의 경우는 1보다 큰지가 기준점이었다면, **Logit 은 0보다 큰지가 기준점**이 된다.)

$$\ln\left[\frac{p}{1-p}\right] = b_0 + b_1x_1 + b_2x_2 + \dots + a_kx_k + \varepsilon$$

$$\frac{p}{(1-p)} = e^{b_0+b_1x^1+b_2x^2+\dots+a_kx^k+\varepsilon}$$

$$p = e^{b_0+b_1x^1+b_2x^2+\dots+a_kx^k+\varepsilon} (1-p)$$

$$p(1 + e^{b_0+b_1x^1+b_2x^2+\dots+a_kx^k+\varepsilon}) = e^{b_0+b_1x^1+b_2x^2+\dots+a_kx^k+\varepsilon}$$

$$p = \frac{e^{b_0+b_1x^1+b_2x^2+\dots+a_kx^k+\varepsilon}}{(1+e^{b_0+b_1x^1+b_2x^2+\dots+a_kx^k+\varepsilon})} = \frac{1}{(1+e^{-(b_0+b_1x^1+b_2x^2+\dots+a_kx^k+\varepsilon)})}$$

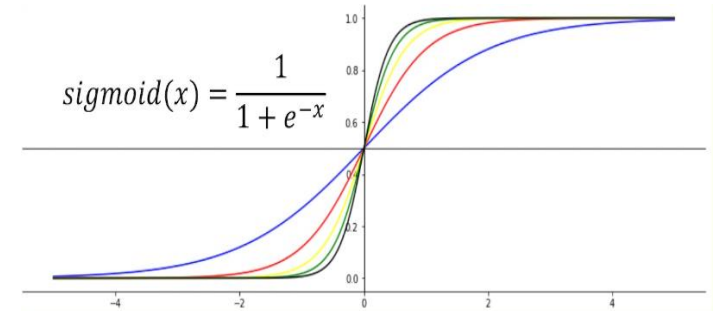
$$f(x) = \frac{1}{1+e^{-x}} : \text{Sigmoid Function}$$

Logistic Function

Activation Function: *Logistic fun.*

- 시그모이드 함수(sigmoid function)에는 0과 1 사이의 값을 출력하는 S자 모양의 함수를 말하며, 보통은 $f(x) = \frac{1}{1+e^{-x}}$ 를 가리킨다.

- 시그모이드 함수의 장점은 다음과 같다.
 1. 실수 전체를 (0, 1)에 매핑시켜 확률처럼 만든다.
 2. 지수 함수를 사용하기 때문에 $x = 0$ 근처에서도 비선형성을 보장한다.
 3. $x = 0$ 에서 함수의 값이 0.5이다.



- 시그모이드 함수의 x 대신 p 로 표현된 회귀식으로 대입하면 다음과 같으며, 이를 **로지스틱 함수**라고 한다.

$$f(x) = \frac{1}{1+e^{-x}} \quad \Rightarrow \quad p = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n \quad \Rightarrow \quad p = \frac{1}{1+e^{-(\beta_0 + \beta_1 x + \dots + \beta_n x_n)}}$$

- 로지스틱 함수를 이용하여 **선형 회귀식이 이진 분류**에 사용하는 것이 가능해졌다. 식의 값이 0.5 이상이면 class A로 분류하고, 0.5 미만이면 class B로 분류하는 등의 규칙을 세우면 된다.

Logit & Softmax

- 로지스틱 함수는 딥 러닝에서 또 다른 말로 **시그모이드 함수** 라고 사용한다.
- 로짓(Logit)은 확률 값으로 변환되기 직전의 최종 결과 값이다. 다른 말로는 **score**라고도 한다.
- 분류(Classification) 계열의 신경망 모델의 마지막 Layer에서는 **소프트맥스(softmax) 함수**가 활용되는데, 로짓 함수는 소프트 맥스 함수에 그 값을 전달해주는 역할을 한다.
- softmax 함수는 로지스틱 함수의 다차원 일반화 개념이다. 종속변수 상태가 **3개 이상**인 멀티 클래스 분류 문제에서 인공 신경망의 최종 Layer로 활용된다.

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)}$$

Logit & Softmax

예) 기상 정보를 토대로 날씨를 예측하는 인공 신경망이 있다고 가정. 날씨의 상태는 맑음, 흐림, 우천 3가지로 구분한다.

- 날씨 상태가 3가지이므로, 인공 신경망의 마지막 Layer에 있는 노드 수가 3개가 된다.
- 3개의 최종 노드에서 출력된 최종 값이 [2.1, 3.5, 5.6]이라고 하자. 이 값이 바로 **Score**이며, **로짓 변환**의 결과 값이다.
- 인공 신경망은 로짓 변환의 결과 값을 활용하여 확률 값을 출력해야 한다. 아래는 Logit 값이 어떻게 softmax 함수를 통과하여 확률 값으로 변하는지를 나타내는 과정이다.

S1: [2.1, 3.5, 5.6]에 지수 함수를 취하면, $[e(2.1), e(3.5), e(5.6)] = [8.2, 33.1, 270.4]$

S2: 모든 값을 더하여 분모로 나누어주면, $[8.2/(8.2+33.1+270.4), 33.1/(8.2+33.1+270.4), 270.4/(8.2+33.1+270.4)]$

S3: 최종 결과: **Logit** [2.1, 3.5, 5.6] → **Softmax** [0.03, 0.11, 0.86]

S4: **[맑음, 흐림, 우천] = [0.03, 0.11, 0.86]** 이므로 신경망은 날씨 상태를 "우천"으로 예측

※ 참고로 Logit 값을 그대로 사용하지 않고, **지수함수를 취하는 이유**는 값들 간의 **차이를 더욱 두드러지게** 하여 신경망 학습이 잘 되도록 하기 위함이다.

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)}$$