

December 10, 2025

- 강의명: CS109A: 데이터 과학 입문
- 주차: Lecture 13
- 교수명: Pavlos Protopapas, Kevin Rader, Chris Gumb
- 목적: Lecture 13의 핵심 개념 학습

## Contents

<b>1</b>	<b>복습: 선형 회귀와 통계적 추론 (Review)</b>	<b>4</b>
1.1	가설 검정과 p-value	4
1.2	순열 검정 (Permutation Test)	4
1.3	순열 검정 vs. 부트스트랩 (Permutation vs. Bootstrap)	5
1.4	상호작용 항 (Interaction Terms) 해석	5
1.5	신뢰 구간 vs. 예측 구간	5
<b>2</b>	<b>분류 (Classification)란 무엇인가?</b>	<b>6</b>
2.1	회귀 vs. 분류 (Regression vs. Classification)	6
2.2	왜 선형 회귀를 분류 문제에 쓰면 안 되는가?	6
2.2.1	문제 1: 다중 클래스의 잘못된 순서 (False Ordering)	6
2.2.2	문제 2: 확률 범위를 벗어남 (Probability Bounds Violation)	6
<b>3</b>	<b>로지스틱 회귀 (Logistic Regression)</b>	<b>8</b>
3.1	핵심 아이디어: S-커브 (The S-Curve)	8
3.2	계수 (Coefficient)의 해석: 오즈 (Odds)와 로그-오즈 (Log-Odds)	8
3.3	모델 추정: 최대가능도추정 (MLE)	9
<b>4</b>	<b>다중 로지스틱 회귀와 결정 경계</b>	<b>11</b>
4.1	다중 로지스틱 회귀 (Multiple Logistic Regression)	11
4.2	결정 경계 (Decision Boundaries)	11
4.3	결정 경계의 형태: 선형과 비선형	11
<b>5</b>	<b>실습 코드 예제 (Python)</b>	<b>13</b>
5.1	순열 검정 (Permutation Test) 예제 (Numpy)	13
5.2	상호작용 모델 (Statsmodels)	13
5.3	단순 로지스틱 회귀 (Scikit-learn)	14
5.4	비선형 결정을 위한 다항 로지스틱 회귀 (Scikit-learn)	14

## 개요 (Overview)

### □ 핵심 요약

이 문서는 데이터 사이언스의 핵심 주제인 '분류(Classification)'를 다룹니다.

지금까지 다룬 '회귀(Regression)'가 숫자(예: 주택 가격)를 예측하는 문제였다면, '분류'는 범주(예: 심장병 유무, 전공)를 예측하는 문제입니다.

이를 위해, 선형 회귀를 분류 문제에 바로 적용할 때 발생하는 문제점들을 살펴보고, 분류를 위한 핵심적인 파라메트릭 모델인 로지스틱 회귀(Logistic Regression)를 배웁니다.

#### 주요 학습 목표:

- 회귀와 분류의 근본적인 차이를 설명할 수 있습니다.
- 왜 선형 회귀를 분류 문제에 사용하면 안 되는지 이해합니다.
- 로지스틱 회귀가 '확률'을 모델링하기 위해 시그모이드(Sigmoid) 함수를 사용하는 원리를 배웁니다.
- 로지스틱 회귀의 계수가 로그-오즈(Log-Odds) 관점에서 어떻게 해석되는지 설명할 수 있습니다.
- 로지스틱 회귀가 최대가능도추정(MLE)과 이진 교차 엔트로피(BCE)를 통해 어떻게 학습되는지 이해합니다.
- 모델의 결정 경계(Decision Boundary)가 어떻게 형성되는지 이해합니다.

또한, 이 주제에 앞서 중간고사에 포함될 수 있는 가설 검정, 순열 검정, 상호작용 항, 예측 구간 등에 대한 핵심 내용을 복습합니다.

## 주요 용어 정리 (Terminology)

본격적인 학습에 앞서, 오늘 다룰 핵심 용어들을 정리합니다.

**Table 1: 분류 및 로지스틱 회귀 핵심 용어**

용어 (Korean)	원어 (English)	쉬운 설명	비고
가설 검정	Hypothesis Testing	데이터가 특정 가설(주장)을 지지하는지 통계적으로 판단하는 과정.	예: $\beta_1 = 0$ 인가?
p-value	p-value	'귀무가설(예: 관계가 없다)'이 맞다고 할 때, 현재 데이터만큼 극단적인 결과가 우연히 나올 확률.	낮을수록(보통 $< 0.05$ ) 관계가 있다고 봄.
순열 검정	Permutation Test	데이터의 라벨(Y)을 무작위로 섞어(순열), 우연만으로 원본 결과가 나오기 힘든 일인지 검증하는 기법.	$t$ -test의 가정(정규성 등)이 깨졌을 때 유용.
부트스트랩	Bootstrap	원본 데이터에서 중복을 허용하여(복원추출) 여러 번 샘플링하는 기법.	주로 신뢰구간 '추정(Estimation)'에 사용.
상호작용 항	Interaction Term	한 변수(X1)의 효과가 다른 변수(X2)의 수준에 따라 달라지는 효과를 나타내는 항.	예: $sqft : type$
예측 구간	Prediction Interval	새로운 개별 관측치(single point)가 존재할 것으로 예상되는 범위.	신뢰 구간(평균)보다 항상 넓음.
분류	Classification	데이터가 어떤 '범주(Category)'에 속하는지 예측하는 문제.	예: 스팸(S) vs. 정상(N)
회귀	Regression	데이터로부터 '연속적인 숫자(Number)'를 예측하는 문제.	예: 주택 가격 예측
시그모이드	Sigmoid Function	모든 입력을 0과 1 사이의 S자 곡선으로 매핑하는 함수.	로지스틱 함수의 별명.
로지스틱 회귀	Logistic Regression	시그모이드 함수를 사용해 데이터가 특정 범주(예: 1)에 속할 확률을 모델링하는 기법.	이름은 '회귀'지만 '분류' 모델임.
오즈	Odds	성공 확률(p)을 실패 확률(1-p)로 나눈 값. $(\frac{p}{1-p})$	$p=0.8$ 이면 $Odds = 4$ (성공이 4배)
로그-오즈	Log-Odds (Logit)	오즈에 자연로그(ln)를 취한 값. $\ln(\frac{p}{1-p})$	로지스틱 회귀는 로그-오즈를 선형 모델링함.
최대가능도추정	MLE (Max Likelihood)	주어진 데이터가 관측될 '가능성(Likelihood)'을 최대로 만드는 모델 파라미터를 찾는 방법.	로지스틱 회귀의 학습 원리.
이진 교차 엔트로피	BCE (Binary Cross-Entropy)	로지스틱 회귀의 손실 함수(Loss Function). (음의 로그 가능도)	BCE를 최소화 = 가능도를 최대화.
결정 경계	Decision Boundary	모델이 클래스 0과 1을 구분하는 경계선. (즉, $P(Y=1) = 0.5$ 가 되는 지점)	기본은 선형, 다항식 함수가 시 곡선 가능.

## 1 복습: 선형 회귀와 통계적 추론 (Review)

분류 모델을 배우기에 앞서, 선형 회귀 모델의 통계적 추론 방식을 복습합니다.

### 1.1 가설 검정과 p-value

- **가설 검정 (Hypothesis Testing)**이란, 우리가 모델에서 발견한 관계(예: 주택 크기와 가격의 관계)가 '진짜'인지, 아니면 '단순한 우연'인지 통계적으로 판단하는 공식적인 절차입니다.
- **귀무가설 ( $H_0$ ):** "관계가 없다." (즉,  $\beta_1 = 0$  이다.)
- **대립가설 ( $H_A$ ):** "관계가 있다." (즉,  $\beta_1 \neq 0$  이다.)

이때 사용되는 핵심 도구가 **t-통계량(t-statistic)**과 **p-value**입니다.

- **t-통계량:** 우리가 추정한 계수( $\hat{\beta}_1$ )가 표준 오차(SE)에 비해 얼마나 큰지 나타내는 값입니다. (즉, 0에서 얼마나 멀리 떨어져 있는가?)

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$$

- **p-value:** 만약 귀무가설( $H_0$ )이 사실(관계가 없음)이라면, 우리가 관찰한 t-통계량만큼 극단적인 값이 순전히 '우연'에 의해 관찰될 확률입니다.

#### □ 예제: title

p-value가 0.001이라는 것은, "만약 주택 크기와 가격이 아무 관계가 없다면, 우리가 현재 데이터에서 본 것과 같은 강한 관계가 우연히 나타날 확률이 0.1%밖에 되지 않는다"는 의미입니다. 이 확률이 매우 낮기(보통 0.05 미만), 우리는 "이건 우연이 아니다"라고 결론 내리고 귀무가설을 기각합니다. 즉, "주택 크기와 가격 사이에는 통계적으로 유의미한 관계가 있다"고 말합니다.

### 1.2 순열 검정 (Permutation Test)

t-test는 데이터가 정규분포를 따르고, 분산이 동일하다(등분산성)는 가정이 필요합니다. 만약 데이터가 이 가정을 만족하지 못하면(예: 예러가 한쪽으로 몰려있음), t-test의 p-value를 신뢰할 수 없습니다.

순열 검정은 이러한 가정 없이 p-value를 계산하는 강력한 **재표본추출(Resampling)** 기법입니다.

- **핵심 아이디어:** 귀무가설( $H_0$ )이 "X와 Y는 관계가 없다"는 것이므로, 이 가설을 시뮬레이션하기 위해 Y값(예: 주택 가격)을 무작위로 뒤섞어버립니다(shuffling).
- 이렇게 하면 X와 Y 사이의 실제 관계가 인위적으로 파괴됩니다.
- **절차:**
  1. 원본 데이터에서 t-통계량 (또는  $\hat{\beta}_1$  값)을 계산합니다. (예:  $\hat{\beta}_1 = 0.5898$ )
  2. Y 값을 무작위로 섞은 후, X와 다시 짝지어 모델을 적합하고  $\hat{\beta}_1^*$  값을 계산합니다. (당연히 0에 가까운 값이 나올 것입니다.)
  3. 이 과정을 1,000 번 (또는 10,000 번) 반복하여, '관계가 없을 때' 나올 수 있는  $\hat{\beta}_1^*$  값들의 분포(귀무 분포)를 만듭니다.
  4. 원본 값(0.5898)이 이 귀무 분포(대부분 0 근처)에서 얼마나 극단적인 위치에 있는지 확인하여 p-value를 계산합니다.

### 1.3 순열 검정 vs. 부트스트랩 (Permutation vs. Bootstrap)

두 기법 모두 데이터를 재표본추출하지만, 목적과 방식이 다릅니다.

Table 2: 순열 검정과 부트스트랩 비교

특징	부트스트랩 (Bootstrap)	순열 검정 (Permutation Test)
목적	추정 (Estimation)	가설 검정 (Hypothesis Testing)
주요 산출물	신뢰 구간 (Confidence Interval)	p-value
샘플링 방식	복원 추출 (With Replacement)	비복원 추출 (Shuffling)
기본 가정	원본 데이터가 모집단을 잘 대표함	귀무가설( $H_0$ )이 참 (X-Y 관계 없음)

### 1.4 상호작용 항 (Interaction Terms) 해석

상호작용 항은 "X1의 효과가 X2의 수준에 따라 달라지는" 효과를 모델링합니다. 예를 들어,  $\text{price} \sim \text{sqft} + \text{type} + \text{sqft}:\text{type}$  모델을 살펴봅시다. (여기서 **type**의 기준(reference) 범주는 'Condo'입니다.)

$$\text{price} = \beta_0 + \beta_1 \cdot \text{sqft} + \beta_2 \cdot \text{type}[\text{multifamily}] + \beta_3 \cdot (\text{sqft} \times \text{type}[\text{multifamily}]) + \dots$$

- $\hat{\beta}_1$  (예: 0.6659): 기준 범주(Condo)의 1 평방 피트당 가격 효과입니다.
- $\hat{\beta}_3$  (예: -0.2863): Multifamily의 1 평방 피트당 가격 효과가 Condo에 비해 얼마나 다른지(차이)를 나타냅니다.
- Multifamily의 실제 평방 피트당 가격 효과는  $\hat{\beta}_1 + \hat{\beta}_3$  (즉,  $0.6659 - 0.2863$ ) 입니다.
- 이 상호작용 항의 p-value가 유의미하다면(예:  $< 0.05$ ), "평방 피트에 따른 가격 변화율이 주택 유형(Condo vs. Multifamily)에 따라 통계적으로 유의미하게 다르다"고 결론 내릴 수 있습니다.

### 1.5 신뢰 구간 vs. 예측 구간

- **신뢰 구간 (Confidence Interval):** (더 좁은 구간)
  - "특정 X 값에 대한 평균 Y값 ( $\hat{Y}$ ) 이 존재할 범위"에 대한 구간입니다.
  - 즉, "우리의 회귀선 자체가 얼마나 정확한가"를 보여줍니다.
  - 데이터가 많아질수록 0에 가깝게 좁아질 수 있습니다.
- **예측 구간 (Prediction Interval):** (더 넓은 구간)
  - "특정 X 값에 대한 새로운 개별 Y값 (single new observation) 이 존재할 범위"에 대한 구간입니다.
  - 이는 회귀선의 불확실성뿐만 아니라, 데이터 고유의 노이즈(줄일 수 없는 오차,  $\epsilon$ )까지 포함합니다.
  - 데이터가 무한히 많아져도, 이 고유의 노이즈 때문에 일정 수준 이하로 좁아지지 않습니다.

## 2 분류 (Classification)란 무엇인가?

### 2.1 회귀 vs. 분류 (Regression vs. Classification)

지금까지 우리가 다룬 문제는 대부분 회귀(Regression)였습니다.

- 회귀 (Regression): 예측하려는 값(Y)이 연속적인 숫자(quantitative)입니다.
  - 예: 내일의 기온(25.5도), 주택 가격(\$500,000), 광고비 대비 매출액(\$18.5)
- 분류 (Classification): 예측하려는 값(Y)이 범주형(qualitative, categorical)입니다.
  - 예: 내일 날씨(맑음, 흐림, 비), 환자의 심장병 유무(Yes, No), 학생의 전공(CS, Stats, Other)

#### □ 예제: title

- 회귀 질문: "이 학생의 최고 심박수는 몇입니까?" (예측: 150 bpm)
- 분류 질문: "이 학생은 심장병이 있습니까, 없습니까?" (예측: Yes)

### 2.2 왜 선형 회귀를 분류 문제에 쓰면 안 되는가?

Y값이 범주형일 때, 선형 회귀( $Y = \beta_0 + \beta_1 X$ )를 그냥 사용하면 두 가지 심각한 문제가 발생합니다.

#### 2.2.1 문제 1: 다중 클래스의 잘못된 순서 (False Ordering)

Y값이 3개 이상의 범주(multi-class)를 가질 때를 생각해봅시다. (예: 전공) 우리가 이 범주를 숫자로 강제 인코딩했다고 가정합니다. ( $Y = 1$  if CS,  $Y = 2$  if Statistics,  $Y = 3$  if Otherwise)

#### 주의사항

선형 회귀는 이 숫자들 사이에 수학적 관계가 있다고 가정합니다.

- 모델은 'CS(1)에서 Stats(2)로의 변화' (+1)와 'Stats(2)에서 Other(3)로의 변화' (+1)를 동일한 크기의 변화로 취급합니다.
- 이는 완전히 무의미한 가정입니다. 만약 'CS=3, Stats=1'로 순서를 바꾸면 모델의 결과가 완전히 달라집니다.

범주형 변수에는 자연스러운 순서나 등간격이 없습니다. (이러한 변수를 *nominal*하다고 합니다.)

#### 2.2.2 문제 2: 확률 범위를 벗어남 (Probability Bounds Violation)

Y값이 2개의 범주(binary)만 가질 때(예: 심장병 Yes=1, No=0)는 순서 문제는 없지만, 더 심각한 문제가 발생합니다.

이때 선형 회귀는  $P(Y = 1)$  (즉, 심장병에 걸릴 '확률')을 예측하도록 학습될 수 있습니다. 하지만 확률은 반드시 0과 1 사이의 값이어야 합니다.

#### 주의사항

선형 회귀의 예측값( $\hat{Y}$ )은 직선이므로, 범위의 제한이 없습니다.

- X가 매우 작으면(예: MaxHR이 매우 낮음), 모델이  $P(Y = 1) = 1.1$  (110%) 과 같이 1보다 큰 값을 예측할 수 있습니다.

- $X$ 가 매우 크면(예: MaxHR이 매우 높음), 모델이  $P(Y = 1) = -0.1$  (-10%) 과 같이 0보다 작은 값을 예측할 수 있습니다.  
이는 수학적으로, 논리적으로 완전히 잘못된 예측입니다.

(참고: 강의 슬라이드 30페이지의 그림은 선형 회귀선이  $Y=0$ 과  $Y=1$  데이터를 벗어나 각각 0 미만, 1 초과를 예측하는 문제점을 시각적으로 보여줍니다.)

### 3 로지스틱 회귀 (Logistic Regression)

#### 3.1 핵심 아이디어: S-커브 (The S-Curve)

선형 회귀의 문제(0 1 범위를 벗어남)를 해결하기 위해, 우리는 예측값이 항상 0과 1 사이에 머무르도록 하는 새로운 함수가 필요합니다.

- (1) 한 줄 핵심 요약: 모든 입력을 0과 1 사이의 S자 곡선으로 '압축'시키는 **시그모이드(Sigmoid)** 함수를 사용해 확률을 모델링합니다.
- (2) 직관적 예시: 선형 회귀의 무한한 직선( $h$ )을 가져와서, 이 직선을 양쪽 끝에서 눌러 0이라는 '바닥'과 1이라는 '천장'에 닿도록 찌그러뜨린 모양을 상상하면 됩니다.
- (3) 기술적 설명:
  - 먼저, 선형 회귀와 똑같은 부분( $h$ )을 계산합니다:  $h = \beta_0 + \beta_1 X$
  - 이  $h$  값을 시그모이드(로지스틱) 함수에 통과시켜 확률  $p$ 를 얻습니다.

$$p = P(Y = 1) = \frac{1}{1 + e^{-h}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

이 함수는  $h$  값에 따라 항상 0과 1 사이의 값을 반환합니다.

- $h \rightarrow +\infty$  (아주 큰 양수) 이면,  $e^{-h} \rightarrow 0$ , 따라서  $p \rightarrow \frac{1}{1+0} = 1$
- $h \rightarrow -\infty$  (아주 큰 음수) 이면,  $e^{-h} \rightarrow \infty$ , 따라서  $p \rightarrow \frac{1}{1+\infty} = 0$
- $h = 0$  이면,  $e^0 = 1$ , 따라서  $p \rightarrow \frac{1}{1+1} = 0.5$

#### 로지스틱 회귀의 이름

이름은 '회귀(Regression)'이지만, 하는 일은 '**분류(Classification)**'입니다. 이는 모델이 확률이라는 '연속적인 숫자'를 예측한 뒤, 그 확률을 기준으로 '범주'를 결정하기 때문입니다.

#### 3.2 계수(Coefficient)의 해석: 오즈(Odds)와 로그-오즈(Log-Odds)

$p = \frac{1}{1+e^{-h}}$  공식은  $\beta_1$ 을 해석하기 매우 어렵습니다. "X가 1 증가할 때,  $p$ 는  $\frac{1}{1+e^{-(\beta_0 + \beta_1(X+1))}}$  ... 만큼 변한다"는 식은 직관적이지 않습니다.

대신, 위 공식을  $h$ 에 대해 정리하면(즉, 역함수를 구하면) 훨씬 강력한 해석이 가능해집니다.

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

이 방정식의 좌변을 이해하기 위해 두 가지 개념을 도입합니다.

- **오즈(Odds):** (성공 확률) / (실패 확률)

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

- $p = 0.5$  (확률 50%)  $\implies$  Odds = 1 (1:1)
- $p = 0.8$  (확률 80%)  $\implies$  Odds = 4 (실패보다 성공이 4배 높음)
- $p = 0.2$  (확률 20%)  $\implies$  Odds = 0.25 (성공보다 실패가 4배 높음)



- **로그-오즈 (Log-Odds) 또는 로짓(Logit):** 오즈에 자연로그(ln)를 취한 값.

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

#### □ 핵심 요약

**로지스틱 회귀의 핵심 해석:** 로지스틱 회귀는 로그-오즈(Log-Odds)를 선형 회귀로 모델링하는 것입니다!

”X가 1단위 증가할 때, 로그-오즈가  $\beta_1$  만큼 더하기(additive)로 변한다.”

이것은 ”X가 1단위 증가할 때, 오즈(Odds)가  $e^{\beta_1}$  만큼 곱하기(multiplicative)로 변한다.”는 의미와 같습니다.

#### □ 예제: title

심장병 예측 모델에서  $X = \text{MaxHR}$  (최대 심박수)에 대한 계수가  $\hat{\beta}_1 = -0.0434$  라고 가정합니다.

- **로그-오즈 해석 (어려움):** 최대 심박수가 1 증가할 때마다, 심장병에 걸릴 로그-오즈가 -0.0434만큼 감소합니다.
- **오즈 해석 (쉬움):**  $e^{\beta_1} = e^{-0.0434} \approx 0.957$   
이는 최대 심박수가 1 증가할 때마다, 심장병에 걸릴 오즈가 약 **0.957배**가 된다는 의미입니다. (즉, 약 4.3%씩 감소합니다.)
- 만약  $\hat{\beta}_1 = 0$  이었다면?  $e^0 = 1$  이므로, 오즈가 1배 (변화 없음)가 됩니다. 즉, X와 Y는 관계가 없습니다.
- 만약  $\hat{\beta}_1 = 0.7$  이었다면?  $e^{0.7} \approx 2.01$  이므로, 오즈가 약 2배 증가합니다.

### 3.3 모델 추정: 최대가능도추정 (MLE)

최적의 S-커브 (즉, 최적의  $\beta_0, \beta_1$ )는 어떻게 찾을까요?

- **선형 회귀의 경우:** 손실 함수인 **MSE**(평균 제곱 오차)를 최소화하는  $\beta$  값을 찾았습니다. (이는 Y가 정규분포를 따른다고 가정한 것과 같습니다.)
- **로지스틱 회귀의 경우:** Y가 0 또는 1이므로, **베르누이 분포(Bernoulli Distribution)** (동전 던지기)를 따른다고 가정합니다.

이때 사용하는 학습 원리가 **최대가능도추정 (MLE, Maximum Likelihood Estimation)**입니다.

- **가능도 (Likelihood):** ”현재 우리가 가정한 S-커브(모델)가, 지금 우리가 가진 데이터(Y=0 또는 1)를 만들어 냈을 총 확률”입니다.
- 모델이 예측한 확률이  $p_i$  일 때:
  - 실제 값이  $y_i = 1$  (성공)이면: 이 관측치의 가능도는  $p_i$
  - 실제 값이  $y_i = 0$  (실패)이면: 이 관측치의 가능도는  $1 - p_i$
- 이를 하나의 수식으로 표현하면  $L_i = p_i^{y_i} \cdot (1 - p_i)^{1-y_i}$  입니다.
- **전체 가능도 (Total Likelihood):** 모든 데이터가 독립이라고 가정하므로, 모든 관측치의 가능도를 곱합니다.

$$L(\beta) = \prod_{i=1}^n L_i = \prod_{i=1}^n p_i^{y_i} \cdot (1 - p_i)^{1-y_i}$$

- **MLE의 목표:** 이  $L(\beta)$  값을 **최대**로 만드는  $\beta$  (즉,  $\beta_0, \beta_1$ )를 찾는 것입니다.

#### 손실 함수: 이진 교차 엔트로피 (BCE)

곱셈( $\prod$ )은 미분하기 매우 어렵습니다. 따라서 계산을 쉽게 하기 위해 양변에 로그(log)를 취합니다. 이를 **로그 가능도(Log-Likelihood)**라고 합니다. (로그를 취해도 최대가 되는 지점은 변하지 않습니다.)

$$l(\beta) = \log L(\beta) = \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

컴퓨터는 보통 '최소화' 문제를 풉니다. 따라서 위 값에 마이너스(-)를 붙인 **음의 로그 가능도(Negative Log-Likelihood)**를 최소화합니다.

$$\text{Loss} = -l(\beta) = - \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

이 손실 함수를 **이진 교차 엔트로피(Binary Cross-Entropy, BCE)**라고 부릅니다. 선형 회귀가 MSE를 최소화하듯, 로지스틱 회귀는 BCE를 최소화합니다.

## 4 다중 로지스틱 회귀와 결정 경계

### 4.1 다중 로지스틱 회귀 (Multiple Logistic Regression)

선형 회귀를 다중 선형 회귀로 확장했듯이, 로지스틱 회귀도 여러 개의 예측 변수( $X$ )를 사용하도록 쉽게 확장할 수 있습니다.

단순히 로그-오즈에 대한 선형 방정식을 확장하면 됩니다.

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

- 해석:  $\beta_j$ 의 해석은 "다른 모든 변수( $X_k$ )가 일정하다고 가정할 때"라는 조건이 추가됩니다.
- 즉,  $e^{\beta_j}$ 는 다른 변수들이 고정된 상태에서  $X_j$ 가 1단위 증가할 때 오즈(Odds)의 곱셈 변화량입니다.
- 다중 공선성, 과적합 등 다중 선형 회귀에서 발생했던 문제들이 여기서도 동일하게 발생하며, 정규화 (Ridge, Lasso) 등이 필요할 수 있습니다.

### 4.2 결정 경계 (Decision Boundaries)

로지스틱 회귀는 확률( $p$ )을 반환합니다. 이를 '분류' (Yes/No)로 바꾸려면 **결정 임계값(Threshold)**이 필요합니다.

- 기본 임계값:  $p \geq 0.5$  이면  $Y = 1$  (Yes)로 분류,  $p < 0.5$  이면  $Y = 0$  (No)로 분류합니다.
- 결정 경계 (Decision Boundary): 모델이 Yes와 No를 구분하는 경계선, 즉  $p = 0.5$ 가 되는 지점입니다.

$p = 0.5$ 는 Odds =  $\frac{0.5}{1-0.5} = 1$ 을 의미하고, Log-Odds =  $\ln(1) = 0$ 을 의미합니다. 따라서 결정 경계는 로지스틱 회귀의 선형 방정식 부분이 0이 되는 지점입니다.

$$\text{결정 경계: } \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

### 4.3 결정 경계의 형태: 선형과 비선형

- 선형 경계 (Linear Boundary): 기본적인 다중 로지스틱 회귀 모델( $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$ )은  $X_1$ 과  $X_2$ 에 대한 1차 방정식이므로, 결정 경계는 항상 직선 (또는 3D에서는 평면)이 됩니다.

(참고: 강의 슬라이드 83페이지는 *MaxHR*과 *Chol* 변수만 사용했을 때, 두 클래스를 나누는 경계가 직선으로 나타나는 것을 보여줍니다.)

- 비선형 경계 (Non-linear Boundary): 하지만 데이터가 직선으로 잘 나뉘지 않는 경우가 많습니다.

(참고: 강의 슬라이드 84페이지는 두 클래스가 곡선 형태로 섞여있어 직선 경계로는 잘 나눌 수 없는 예시를 보여줍니다.)

**해결책:** 선형 회귀에서 다항 회귀를 사용했듯이, 로지스틱 회귀에도 변수들을 변형하여 추가합니다.

1. 상호작용 항 추가:  $X_3 = X_1 \cdot X_2$
2. 다항 항 추가:  $X_4 = X_1^2, X_5 = X_2^2$

이렇게 변형된 변수들로 모델을 만들면,

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 (X_1 \cdot X_2) + \beta_4 X_1^2 + \beta_5 X_2^2$$

결정 경계 ( $= 0$ )는  $X_1, X_2$ 에 대한 2차 방정식이 되므로, 곡선, 원, 타원 형태의 비선형 경계를 만들어낼 수 있습니다.

#### 모델은 여전히 '선형'입니다

비선형 경계를 만들었음에도 불구하고, 이 모델은 여전히 '선형 모델'로 분류됩니다.

왜냐하면 모델은 계수( $\beta$ )에 대해 선형이기 때문입니다. ( $X_1^2$ 를 그냥  $Z_4$ 라는 새로운 변수로 보면, 모델은  $\beta_0 + \beta_1 Z_1 + \dots + \beta_5 Z_5$  형태의 선형 결합입니다.)

우리는 입력 변수( $\mathbf{X}$ )를 비선형으로 변환(feature engineering)하여, 선형 모델로 비선형 경계를 찾도록 한 것입니다.

## 5 실습 코드 예제 (Python)

강의에서 사용된 statsmodels 및 scikit-learn 코드 예제입니다.

### 5.1 순열 검정 (Permutation Test) 예제 (Numpy)

```

1 import numpy as np
2 import sklearn.linear_model
3
4 nsims = 1000
5 X = homes[['sqft']]
6 y = homes[['price']]
7
8 indices = np.arange(0, len(homes))
9 beta1_permute = []
10
11 for i in np.arange(0, nsims):
12     # 1. 귀무가설을시뮬레이션하기위해의 y 인덱스를섞습니다 .
13     np.random.shuffle(indices)
14     y_permute = y.iloc[indices]
15
16     # 2. 섞인 y 원본로 X 모델을적합합니다 .
17     permute_ols = sk.linear_model.LinearRegression().fit(X, y_permute)
18
19     # 3. 귀무가설하의기울기 (beta1)를 저장합니다.
20     beta1_permute.append(permute_ols.coef_[0][0])
21
22 # 의 beta1_permute 분포귀무 ( 분포)와
23 # 원본데이터의기울기 (beta1_observed)를 비교하여 p-를 value 계산합니다.

```

Listing 1: Numpy를 이용한 순열 검정 구현

### 5.2 상호작용 모델 (Statsmodels)

```

1 import statsmodels.formula.api as smf
2
3 # price ~ sqft + type + sqft:type 모델을적합합니다 .
4 # 'type은' 범주형변수로자동인식됩니다 .
5 interaction_ols = smf.ols(formula="price ~ sqft * type",
6                           data=homes).fit()
7
8 # 결과요약출력
9 print(interaction_ols.summary())
10
11 # coef
12 # -----
13 # Intercept                170.5182
14 # type[T.multifamily]      142.0626
15 # type[T.singlefamily]    -708.8103

```

```

16 # sqft                                0.6659 <-- Condo기준()의 sqft 기울기
17 # sqft:type[T.multifamily]            -0.2863 <-- Condo 대비의 multifamily 기울기차이
18 # sqft:type[T.singlefamily]          0.4769 <-- Condo 대비의 singlefamily 기울기차
    이 ''
19 # ...

```

Listing 2: Statsmodels를 이용한 상호작용 모델 적합

### 5.3 단순 로지스틱 회귀 (Scikit-learn)

```

1 from sklearn.linear_model import LogisticRegression
2
3 # X 예측 (변수, 2D 배열이어야함)
4 X_hr = df_heart[['MaxHR']]
5 # Y 반응 (변수, 1D 배열)
6 y_ahd = df_heart['AHD']
7
8 # penalty='none' : 정규화(Ridge/Lasso)를 사용하지않음
9 logreg = LogisticRegression(penalty='none')
10 logreg.fit(X_hr, y_ahd)
11
12 # beta_1 기울기()
13 print('Estimated beta1: \n', logreg.coef_)
14 # [[-0.04341112]]
15
16 # beta_0 절편()
17 print('Estimated beta0: \n', logreg.intercept_)
18 # [6.3249492]
19
20 # 모델: log(odds) = 6.325 - 0.0434 * MaxHR

```

Listing 3: Scikit-learn을 이용한 단순 로지스틱 회귀

### 5.4 비선형 결정을 위한 다항 로지스틱 회귀 (Scikit-learn)

```

1 # 1. 비선형특성생성
2 df_heart['Interaction'] = df_heart.MaxHR * df_heart.Chol
3 df_heart['MaxHR_sq'] = df_heart.MaxHR**2
4 df_heart['Chol_sq'] = df_heart.Chol**2
5
6 # 사용할변수리스트
7 features = ['MaxHR', 'Chol', 'Interaction', 'MaxHR_sq', 'Chol_sq']
8
9 data_x = df_heart[features]
10 data_y = df_heart['AHD']
11
12 # 2. 모델적합
13 logreg_poly = LogisticRegression(penalty='none', fit_intercept=True, max_iter
    =1000)
14 logreg_poly.fit(data_x, data_y)

```

```
15
16 # 3. 계수확인
17 print('Estimated betas: \n', logreg_poly.coef_)
18 print('Estimated beta0: \n', logreg_poly.intercept_)
19
20 # 이모델의결정경계 (log_odds = 0)는  $X_1$ , 에  $X_2$  대한차식이 2 되어
21 # 원형또는타원형의곡선경계를생성합니다 .
```

Listing 4: 다항 및 상호작용 항을 사용한 로지스틱 회귀

## 빠르게 훑어보기 (1-Page Summary)

### 분류(Classification)와 로지스틱 회귀 핵심 요약

#### 1. 문제 정의: 회귀 vs. 분류

- **회귀 (Regression):** 숫자 예측 (예: 가격, 온도). *Tool:* 선형 회귀
- **분류 (Classification):** 범주 예측 (예: Yes/No, 스팸/정상). *Tool:* 로지스틱 회귀

#### 2. 왜 선형 회귀는 분류에 실패하는가?

- **이유 1 (다중 클래스):** 1=CS, 2=Stats 처럼 강제 인코딩 시, 무의미한 순서와 간격을 가정하게 됨.
- **이유 2 (이진 클래스):** 예측값이 확률의 범위 [0, 1]을 벗어남 (예: 110% 또는 -10%).

#### 3. 해결책: 로지스틱 회귀와 시그모이드

- 선형 회귀의 결과 ( $h = \beta_0 + \beta_1 X$ )를 시그모이드(Sigmoid) 함수에 넣어 0 1 사이의 확률 값( $p$ )으로 '압축' 시킴.

$$p = P(Y = 1) = \frac{1}{1 + e^{-h}}$$

#### 4. 핵심 해석: 로그-오즈 (Log-Odds)

- 모델의 공식을 변형하면 로그-오즈(Logit)가  $X$ 에 대한 선형 함수임을 알 수 있음.

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

- $\beta_1$ 의 해석:  $X$ 가 1 증가할 때, 오즈(Odds)가  $e^{\beta_1}$  배가 된다.
- $e^{\beta_1} > 1$  (긍정적 관계),  $e^{\beta_1} = 1$  (관계 없음),  $e^{\beta_1} < 1$  (부정적 관계)

#### 5. 학습 원리: MLE와 BCE

- 가정:  $Y$ 는 베르누이 분포를 따름 (동전 던지기).
- 목표: 관측된 데이터가 나타날 가능성도(Likelihood)를 최대로 만드는  $\beta$ 를 찾음 (MLE).
- 손실 함수: 이진 교차 엔트로피(BCE) (음의 로그 가능성도)를 최소화함.

#### 6. 결정 경계 (Decision Boundary)

- 모델이 0과 1을 나누는 경계선. (즉,  $p = 0.5$  또는 Log-Odds = 0 이 되는 지점)
- 기본 모델:  $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0 \implies$  직선 경계.
- 다항/상호작용 모델:  $\beta_0 + \dots + \beta_4 X_1^2 + \beta_5 X_2^2 = 0 \implies$  곡선 경계.