선형 회귀의 모든 것
단순 회귀부터 다중 회귀, 모델 해석까지

CS1090A Introduction to Data Science
(강사: Pavlos Protopapas, Kevin Rader, Chris Gumb)

October 26, 2025

▣ **강의명:** CS109A: 데이터 과학 입문

▣ **주차:** Lecture 05

▣ **교수명:** Pavlos Protopapas, Kevin Rader, Chris Gumb

▣ **목적:** Lecture 05의 핵심 개념 학습

Contents

Part I

선형 회귀의 기초

1 개요 (Overview)

선형 회귀(Linear Regression)는 데이터 과학과 기계 학습에서 가장 기본이 되는 핵심 모델입니다.

이 문서는 선형 회귀의 기초부터 실제 적용까지 초심자도 완벽히 이해할 수 있도록 구성되었습니다.

□ 핵심 요약

이 문서의 핵심 요약:

- **선형 회귀란?** 하나 이상의 입력 변수(예측 변수)와 하나의 연속적인 출력 변수(반응 변수) 사이의 **선형(직선) 관계**를 모델링하는 기법입니다.
- **왜 중요한가?** 복잡한 모델(신경망 등)을 이해하는 기초가 되며, "어떤 변수가 결과에 얼마나 영향을 미치는지" **해석**하기 용이합니다.
- **학습 흐름:**
  1. **단순 선형 회귀 (SLR):** 하나의 입력(X)으로 하나의 출력(Y)을 예측합니다. ($Y = \beta_0 + \beta_1 X$)
  2. **다중 선형 회귀 (MLR):** 여러 개의 입력($X_1, X_2, ...$)으로 하나의 출력(Y)을 예측합니다. ($Y = \beta_0 + \beta_1 X_1 + ...$)
  3. **모델 학습:** '최적의 선'을 찾기 위해 **평균 제곱 오차(MSE)**라는 손실 함수를 최소화합니다.
  4. **모델 해석 및 함정:** 계수(coefficient)의 의미, 스케일링, 다중공선성, 범주형 변수 처리 방법을 배웁니다.

2 핵심 용어 정리

선형 회귀를 이해하기 위해 다음 용어들을 먼저 숙지해야 합니다.

Table 1: 선형 회귀 핵심 용어

용어	원어	쉬운 설명	비고
반응 변수	Response Variable	우리가 예측하려는 **결과값** (Y)	종속 변수(Dependent Variable)라고도 함
예측 변수	Predictor Variable	결과를 예측하는 데 사용하는 **입력값** (X)	특성(Feature), 독립 변수라고도 함
계수	Coefficient	예측 변수가 결과에 미치는 **영향력** ($\beta_1, \beta_2 ...$)	기울기(Slope)라고도 함
절편	Intercept	모든 예측 변수가 0일 때의 **기본값** (β_0)	y 절편, 편향(Bias)이라고도 함
모델	Model	입력(X)을 출력(Y)으로 변환하는 수학 공식	$\hat{Y} = \beta_0 + \beta_1 X$
잔차	Residual	**실제값**(Y)과 모델의 **예측값**($\hat{Y}$)의 차이	$r = Y - \hat{Y}$, 즉 '모델이 틀린 정도'
손실 함수	Loss Function	모델이 얼마나 '못'하는지 측정하는 함수	이 함수의 값을 최소화하는 것이 학습의 목표
평균 제곱 오차	MSE	잔차들을 제곱하여 평균 낸 값	**Mean Squared Error**. 대표적인 손실 함수
학습/피팅	Training / Fitting	데이터로부터 최적의 계수(β)를 찾는 과정	손실 함수(MSE)를 최소화하는 과정
정규 방정식	Normal Equation	미분을 통해 MSE를 최소화하는 β를 한 번에 찾는 공식	$\hat{\beta} = (X^T X)^{-1} X^T Y$

3 왜 선형 회귀를 사용할까요?

세상에는 KNN, 신경망 등 복잡하고 강력한 모델이 많습니다. 왜 단순해 보이는 선형 회귀부터 배울까요?

1. **모든 모델의 기초:** 복잡한 딥러닝 모델도 결국은 선형 변환과 비선형 함수의 조합입니다. 선형 회귀의 원리(모델 정의 → 손실 함수 → 최적화)를 완벽히 이해하면, 다른 모든 기계 학습 모델을 이해하는 튼튼한 기반이 됩니다.
2. **뛰어난 해석력 (Interpretability):** KNN 같은 모델은 "왜" 그런 예측이 나왔는지 설명하기 어렵습니다 (Non-parametric). 하지만 선형 회귀는 "어떤 변수가 결과에 얼마나 영향을 주는지" 명확하게 숫자로 보여줍니다.

□ 예제:

예시: KNN vs 선형 회귀

- **KNN (K-최근접 이웃):** "TV 광고 예산이 1억일 때 예상 매출은?" → "과거 1억과 비슷했던 3개 지점의 평균 매출이 10억이니, 10억일 겁니다." "TV 광고 예산을 2배로 늘리면 매출은?" → "음... 다시 계산해봐야 합니다." (직관적이지 않음)
- **선형 회귀:** "TV 광고 예산이 1억일 때 예상 매출은?" → "학습된 공식 $= 5 + 0.05 \times (TV \quad)$에 따라 10억입니다." "TV 광고 예산을 2배로 늘리면 매출은?" → "계수(기울기)가 0.05이므로, 광고비가 1억 증가할 때마다 매출이 5억씩 증가하는 경향이 있습니다." (직관적 해석 가능)

4 단순 선형 회귀 (Simple Linear Regression, SLR)

가장 간단한 형태로, **하나의 예측 변수**(X)가 **하나의 반응 변수**(Y)에 미치는 영향을 모델링합니다.

4.1 모델 정의: "최적의 직선 찾기"

우리는 X와 Y 사이에 직선 관계가 있다고 **가정**합니다. 모든 데이터 포인트를 완벽하게 지나는 직선은 없으므로, 약간의 오차(ϵ)를 포함합니다.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- Y: 반응 변수 (예: 매출)
- X: 예측 변수 (예: TV 광고비)
- β_0: **절편**. X가 0일 때의 Y 값 (광고비가 0일 때의 기본 매출)
- β_1: **기울기 (계수)**. X가 1단위 증가할 때 Y의 평균적인 변화량 (광고비 1원 증가 시 매출 변화량)
- ϵ: **오차(Error)**. 모델이 설명하지 못하는 무작위성 (다른 요인들)

우리의 목표는 데이터를 가장 잘 설명하는 β_0와 β_1를 찾는 것입니다. 이 예측된 모델을 $\hat{Y}$ (Y-hat)

이라고 부릅니다.

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

4.2 최적의 선 찾기: 손실 함수 (Loss Function)

수많은 직선 중에 "최적의 선"은 무엇일까요? 바로 **"실제 데이터와 가장 가까운 선"**입니다. 이 "가까운 정도"를 측정하는 것이 **손실 함수**입니다.

1. 잔차 (Residuals)

- **정의:** 실제 값(Y_i)과 모델의 예측 값($\hat{Y}_i$)의 차이입니다.
- **수식:** $r_i = Y_i - \hat{Y}_i$
- **비유:** 예측 선에서 실제 데이터 점까지의 "수직 거리"입니다. 이 거리가 짧을수록 좋은 모델입니다.

2. 평균 제곱 오차 (Mean Squared Error, MSE)

- **정의:** 모든 데이터의 잔차(r_i)를 **제곱**하여 더한 뒤, 데이터 개수(n)로 나눈 값입니다.
- **수식:** $L(\beta_0, \beta_1) = \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - (\beta_0 + \beta_1 X_i))^2$

주의사항

Q: 왜 잔차를 그냥 더하지 않고 제곱하나요?
A: 만약 제곱하지 않고 그냥 더하면, 예측보다 위에 있는 점(잔차 > 0)과 아래에 있는 점(잔차 < 0)이 서로 상쇄되어, 실제로는 오차가 큼에도 불구하고 총합이 0에 가까워질 수 있습니다.
제곱을 하는 이유:
1. 모든 잔차를 양수로 만듭니다. (상쇄 방지)
2. 오차가 큰 값(Outlier)에 더 큰 페널티를 부여합니다. (10의 제곱 $= 100$, 2의 제곱 $= 4$)
3. 수학적으로 미분하기 쉬워져 최적화에 유리합니다.

4.3 최적화: 손실 최소화하기

기계 학습의 핵심 3단계를 기억하세요.

기계 학습의 핵심 3단계 프로세스

1. **모델 정의 (Define Model):** $\hat{Y} = \beta_0 + \beta_1 X$ (직선이라고 가정)
2. **손실 정의 (Define Loss):** $\text{MSE} = \frac{1}{n}\sum(Y_i - \hat{Y}_i)^2$ (틀린 정도를 측정)
3. **손실 최소화 (Minimize Loss):** MSE가 가장 작아지는 β_0와 β_1을 찾는다.

MSE는 β_0와 β_1에 대한 2차 함수(3D 그릇 모양)입니다. 이 그릇의 **가장 낮은 지점**을 찾는 것이 목표입니다.

□ **예제:**

비유: 산에서 가장 낮은 계곡 찾기

- **현재 위치:** (β_0, β_1) 값
- **고도:** MSE 값
- **목표:** 고도(MSE)가 가장 낮은 지점 찾기
- **방법: 기울기(경사)**가 0이 되는 지점을 찾습니다.

수학적으로 "기울기가 0"인 지점은 **미분(Derivative)**을 통해 찾습니다. 손실 함수 L을 β_0와 β_1 각각에 대해 **편미분(Partial Derivative)**하여 0이 되는 지점을 찾습니다.

$$\frac{\partial L}{\partial \beta_0} = 0 \quad \text{and} \quad \frac{\partial L}{\partial \beta_1} = 0$$

이 두 방정식을 연립하여 풀면, MSE를 최소화하는 $\hat{\beta}_0$와 $\hat{\beta}_1$의 공식을 유도할 수 있습니다. 이를 **정규 방정식 (Normal Equation)** 또는 **최소 제곱법 (Least Squares)**이라고 합니다.

단순 선형 회귀의 정규 방정식 (Closed-form Solution)

복잡한 미분 과정(연쇄 법칙 포함)을 거치면 다음과 같은 깔끔한 공식을 얻을 수 있습니다.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$$

- $\bar{X}$: X의 평균
- $\bar{Y}$: Y의 평균

중요: 이 공식은 컴퓨터가 `.fit()` 명령을 실행할 때 내부적으로 계산하는 값입니다. 이처럼 최적의 해를 한 번의 계산으로 찾을 수 있는 경우는 매우 드물며, 선형 회귀의 강력한 특징입니다.

Part II

다중 선형 회귀로의 확장

5 다중 선형 회귀 (Multi-Linear Regression, MLR)

현실에서는 하나의 요인만으로 결과를 예측하기 어렵습니다. (예: 매출은 TV 광고비뿐만 아니라 라디오, 신문 광고비, 소셜 미디어 등에도 영향을 받음)

다중 선형 회귀는 **여러 개의 예측 변수**($X_1, X_2, \ldots, X_p$)를 사용하여 Y를 예측합니다.

5.1 모델 정의: "최적의 초평면 찾기"

SLR이 2D 평면에서 '선'을 찾는 것이라면, MLR은 3D 공간에서 '평면'을, 그 이상의 p차원 공간에서 '**초평면(Hyperplane)**'을 찾는 것입니다.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

- β_0: **절편**. 모든 예측 변수($X_1, \ldots, X_p$)가 0일 때의 Y 값.
- β_j: j번째 예측 변수의 **계수**. **해석이 중요: "다른 모든 예측 변수가 고정되어 있다고 가정할 때,"** X_j가 1단위 증가할 때 Y의 평균 변화량.

5.2 행렬 표기법 (Matrix Notation)

변수가 많아지면 위 공식을 쓰기 번거롭습니다. **선형 대수(행렬)**를 사용하면 매우 깔끔하게 표현할 수 있습니다.

n개의 데이터와 p개의 예측 변수가 있다고 가정합시다.

- **Y (반응 변수 벡터):** $n \times 1$ 행렬 (결과값 n개)

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}$$

- **X (설계 행렬, Design Matrix):** $n \times (p+1)$ 행렬 (데이터 n개, 변수 p개 + **절편용 1**)

$$X = \begin{pmatrix} 1 & X_{11} & X_{12} & \ldots & X_{1p} \\ 1 & X_{21} & X_{22} & \ldots & X_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \ldots & X_{np} \end{pmatrix}$$

- β **(계수 벡터):** $(p+1) \times 1$ 행렬 (찾아야 할 파라미터 $p+1$ 개)

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

주의사항

Q: 왜 X 행렬에 '1'로 채워진 첫 번째 열이 있나요?
A: 수학적 트릭입니다. $Y = \beta_0 + \beta_1 X_1 + \dots$ 공식을 행렬 곱으로 표현하면 β_0 (절편)이 따로 떨어져 있어 불편합니다. X에 1을 추가하고 β에 β_0를 포함시키면, 행렬 곱셈 $X\beta$의 첫 번째 항이 $(1 \times \beta_0) + (X_1 \times \beta_1) + \dots$ 가 되어 절편을 자연스럽게 수식에 포함시킬 수 있습니다.

이제 다중 선형 회귀 모델은 단 세 개의 기호로 표현됩니다.

$$Y = X\beta + \epsilon$$

우리의 예측 모델은 $\hat{Y} = X\hat{\beta}$ 가 됩니다.

5.3 최적화: 다중 회귀의 정규 방정식

SLR과 마찬가지로, MSE를 최소화하는 $\hat{\beta}$ 벡터를 찾아야 합니다. 손실 함수 MSE를 행렬로 표현하면 다음과 같습니다.

$$L(\beta) = \text{MSE} = \frac{1}{n}||Y - X\beta||^2 = \frac{1}{n}(Y - X\beta)^T(Y - X\beta)$$

이 손실 함수를 β 벡터에 대해 미분하여 0으로 놓고 풀면 (선형 대수 연산 필요), $\hat{\beta}$를 구하는 강력한 공식을 얻습니다.

다중 선형 회귀의 정규 방정식 (The Normal Equation)

MSE를 최소화하는 계수 벡터 $\hat{\beta}$는 다음 공식으로 한 번에 계산됩니다.

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

- X^T: X의 **전치 행렬** (Transpose, 행과 열을 바꿈)
- $(...)^{-1}$: **역행렬** (Inverse, 행렬의 나눗셈)

이 공식이 바로 `scikit-learn`의 `reg.fit(X, y)` 명령이 내부적으로 수행하는 핵심 계산입니다.

Part III

모델 활용 및 해석

6 Python scikit-learn을 이용한 실습

이론적으로 유도된 정규 방정식을 직접 계산할 필요는 없습니다. Python의 scikit-learn 라이브러리가 이 모든 것을 대신해줍니다.

```python
# 1. 라이브러리임포트
from sklearn.linear_model import LinearRegression
import pandas as pd
import numpy as np

# 2. 데이터준비예시 (: 광고데이터 )
# df = pd.read_csv('Advertising.csv')
# X = df[['TV', 'Radio', 'Newspaper']].values # 예측변수행렬 ()
# y = df['Sales'].values                      # 반응변수벡터 ()

# --- 가상( 데이터생성 ) ---
X = np.array([[100, 20], [150, 30], [200, 25], [300, 40]])
y = np.array([11, 16, 18, 25])
# -------------------------

# 3. 모델객체생성인스턴스화 ()
reg = LinearRegression()

# 4. 모델학습피팅 ()
# 이 .fit() 한줄이 (X^T X)^-1 X^T Y 계산을수행합니다 !
reg.fit(X, y)

# 5. 결과확인
print(f"계수 (beta_1, beta_2...): {reg.coef_}")
print(f"절편 (beta_0): {reg.intercept_}")

# 6. 새로운데이터로예측
new_data = np.array([[250, 35]]) # TV=250, Radio일=35 때?
prediction = reg.predict(new_data)
print(f"예측된 매출: {prediction[0]}")
```

Listing 1: scikit-learn을 이용한 선형 회귀 학습

7 모델 파라미터 해석하기

모델을 만드는 것보다 중요한 것은 **결과를 해석**하는 것입니다.

- **단순 선형 회귀 (SLR)의 $\hat{\beta}_1$:** "X가 1단위 증가할 때, Y는 평균적으로 $\hat{\beta}_1$ 만큼 변화한다." (예: $\hat{\beta}_1 = 0.05 \rightarrow$ "TV 광고비를 1천원 더 쓰면, 매출은 평균 50유닛 증가한다.")
- **다중 선형 회귀 (MLR)의 $\hat{\beta}_j$: "다른 모든 변수(X_k)가 일정하다고 가정할 때,"** X_j 가 1단위 증가하면, Y는 평균적으로 $\hat{\beta}_j$ 만큼 변화한다." (예: $\hat{\beta}_{tv} = 0.04, \hat{\beta}_{radio} = 0.15 \rightarrow$ "라디오와 신문

광고비를 고정시킨 채, TV 광고비를 1천원 더 쓰면 매출은 평균 40유닛 증가한다.")

변수가 많을 때는 계수 값을 시각화하는 **특성 중요도 그래프(Feature Importance Plot)**를 사용합니다. 막대가 길수록(양/음 방향 모두) 해당 변수가 예측에 큰 영향을 미친다는 의미입니다.

8 모델 정확도를 위한 고려사항

모델을 그냥 만들고 끝내면 안 됩니다. 계수 값을 신뢰할 수 있는지, 모델이 안정적인지 확인해야 합니다.

8.1 스케일링 (Scaling)

주의사항

문제점: "단위"가 다르면 계수 비교가 불가능합니다.
'TV 광고비' (단위: 억 원)와 '라디오 광고비' (단위: 만 원)를 예측 변수로 사용했다고 가정해봅시다. TV 광고비가 1단위(1억) 변하는 것과 라디오 광고비가 1단위(1만원) 변하는 것은 크기 자체가 다릅니다.
이때 $\hat{\beta}_{tv} = 10$, $\hat{\beta}_{radio} = 0.1$ 이 나왔다고 해서 "TV 광고가 라디오보다 100배 중요하다"고 말할 수 없습니다. 변수의 **스케일(단위)**이 다르기 때문에 β 계수의 절대 크기를 직접 비교하는 것은 무의미합니다.

해결책: 스케일링 모든 예측 변수 X들을 학습 전에 비슷한 범위(스케일)로 변환합니다.

1. **표준화 (Standardization):** 데이터를 평균 0, 표준편차 1인 분포로 변환합니다. (Z-score)

$$X_{\text{scaled}} = \frac{X - \text{mean}(X)}{\text{std}(X)}$$

2. **정규화 (Normalization):** 데이터를 0과 1 사이의 범위로 변환합니다. (Min-Max Scaling)

$$X_{\text{scaled}} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

스케일링을 수행한 후 모델을 학습시키면, β 계수들은 **단위의 영향에서 벗어나** 변수의 순수한 중요도를 (근사적으로) 비교할 수 있게 됩니다.

8.2 다중공선성 (Collinearity)

주의사항

문제점: 예측 변수끼리 너무 친한 경우
다중공선성이란 **예측 변수들끼리 높은 상관관계**를 갖는 상황을 말합니다. (예: X_1='신용한도', X_2='신용 등급'. 두 변수는 거의 같은 정보를 담고 있음)
비유: 공로를 구분하기 힘든 두 가수 "두 명의 가수(예측 변수)가 정확히 똑같은 멜로디(정보)

를 부르며 노래(반응 변수)의 인기에 기여하고 있습니다. 이때 노래 인기의 공로가 누구에게 몇
결과:

1. 모델의 전체적인 예측 성능(MSE)은 괜찮을 수 있습니다.
2. 하지만 **개별 β 계수의 신뢰도가 박살납니다.**
3. β 값이 비상식적으로 커지거나, 부호가 반대로 나올 수 있습니다.
4. 데이터를 조금만 바꿔도 β 값이 크게 널뛰기합니다. (불안정)

(예: '신용 한도'를 제거했더니 '신용 등급'의 β 값이 1.1에서 3.9로 갑자기 뛰어오름)

해결책:

- **시각화:** 변수 간의 산점도 행렬(Scatter Matrix)을 그려 높은 상관관계를 확인합니다.
- **제거:** 상관관계가 매우 높은 변수 중 하나를 제거합니다.

8.3 범주형 예측 변수 (Qualitative Predictors)

'성별' (Male/Female), '학생 여부' (Yes/No), '인종' (Asian/Caucasian/...)처럼 숫자가 아닌 텍스트 데이터는 어떻게 처리할까요?

해결책 1: 더미 변수 (Dummy Variables) (2개의 레벨을 가질 때)

컴퓨터가 이해하도록 0과 1로 바꿔줍니다. (예: '성별' 변수 → 'is_female' 이라는 새 변수 생성)

$$x_{\text{is\_female}} = \begin{cases} 1 & \text{if person is female} \\ 0 & \text{if person is male} \end{cases}$$

이 변수를 모델에 포함시키면 ($Y = \beta_0 + \beta_1 x_{\text{is\_female}}$) 해석이 매우 흥미로워집니다.

- **Male ($x = 0$):** $Y = \beta_0 + \beta_1(0) = \beta_0 \to \beta_0$ (절편)는 **남성의 평균 Y 값 (기준선)**이 됩니다.
- **Female ($x = 1$):** $Y = \beta_0 + \beta_1(1) = \beta_0 + \beta_1 \to \beta_1$은 **여성과 남성의 평균 Y 값 차이**가 됩니다.

해결책 2: 원-핫 인코딩 (One-Hot Encoding) (3개 이상의 레벨을 가질 때)

(예: '인종' 변수 → 'Asian', 'Caucasian', 'African American')

k 개의 레벨이 있다면, $k-1$ 개의 더미 변수를 만듭니다. (하나를 기준선으로 삼음)

$$x_{\text{is\_Asian}} = \begin{cases} 1 & \text{if Asian} \\ 0 & \text{else} \end{cases} \qquad x_{\text{is\_Caucasian}} = \begin{cases} 1 & \text{if Caucasian} \\ 0 & \text{else} \end{cases}$$

모델: $Y = \beta_0 + \beta_1 x_{\text{is\_Asian}} + \beta_2 x_{\text{is\_Caucasian}}$

- **African American (기준선, $x_1 = 0, x_2 = 0$):** $Y = \beta_0$
- **Asian ($x_1 = 1, x_2 = 0$):** $Y = \beta_0 + \beta_1$
- **Caucasian ($x_1 = 0, x_2 = 1$):** $Y = \beta_0 + \beta_2$

→ β_0는 기준선(African American)의 평균 Y가 되고, β_1과 β_2는 각각 기준선과의 차이를 나타냅니다.

Part IV

학습 점검

9 핵심 학습 체크리스트

이 문서를 다 읽은 후, 다음 질문에 답할 수 있는지 확인하세요.

선형 회귀가 KNN과 같은 다른 모델에 비해 갖는 장점(해석력)은 무엇인가?

'모델 학습(Training)'의 3단계 프로세스(모델 정의, 손실 정의, 손실 최소화)를 설명할 수 있는가?

손실 함수로 MSE를 사용할 때, 왜 잔차를 그냥 더하지 않고 '제곱'하는가?

단순 선형 회귀(SLR)의 β_1 계수의 의미를 정확히 설명할 수 있는가?

다중 선형 회귀(MLR)의 β_j 계수의 의미를 "다른 변수를 고정할 때"라는 조건과 함께 설명할 수 있는가?

`scikit-learn`의 `.fit()` 메소드가 내부적으로 어떤 수학적 계산(정규 방정식)을 수행하는지 아는가?

왜 변수 스케일링(Scaling)이 필요한가? (단위가 다른 변수 간 계수 비교 문제)

다중공선성(Collinearity)이 무엇이며, 왜 모델 '해석'에 문제를 일으키는지 설명할 수 있는가?

'성별'과 같은 범주형 데이터를 모델에 포함시키기 위한 '더미 변수' 기법을 설명할 수 있는가?

10 초심자 FAQ

주의사항

Q: 왜 손실 함수로 잔차의 '절대값'이 아닌 '제곱'(MSE)을 주로 쓰나요? A: 절대값(MAE, Mean Absolute Error)도 좋은 손실 함수입니다. 하지만 MSE를 더 선호하는 두 가지 이유가 있습니다. 1) MSE는 수학적으로 미분이 부드럽게 가능하여 최적화(가장 낮은 지점 찾기)에 유리합니다. 2) MSE는 오차가 큰 값(Outlier)에 제곱으로 페널티를 주므로, 모델이 큰 실수를 하지 않도록 유도하는 경향이 있습니다.

Q: `reg.fit(X, y)` 명령은 마법 상자인가요? 정확히 뭘 하는 거죠? A: 마법이 아닙니다! `.fit()`은 이 문서에서 배운 **정규 방정식** $\hat{\beta} = (X^T X)^{-1} X^T Y$ 공식을 데이터 X와 y에 대해 정확히 계산하여, MSE를 최소화하는 $\hat{\beta}$ 벡터(즉, `reg.coef_`와 `reg.intercept_`)를 찾아내는 과정입니다.

Q: 스케일링을 하면 모델의 예측 성능(MSE)이 좋아지나요? A: 단순 선형 회귀나 다중 선형 회귀에서는 스케일링이 예측 성능 자체에 영향을 주지 않습니다. (어차피 정규 방정식으로 최적의 해를 찾기 때문입니다.) 하지만 계수를 **해석**하고 **비교**하기 위해 스케일링이 필요합니다. (참고: 경사 하강법(Gradient Descent)을 사용하는 모델이나, 정규화(Ridge/Lasso)가 포함된 모델에서는 스케일링이 성능과 수렴 속도에 큰 영향을 줍니다.)

Q: 다중공선성이 높으면 모델이 "틀린" 건가요? A: "틀렸다"기보다는 "불안정하다"고 표현하는 것이 맞습니다. 모델의 **예측 성능 자체는 여전히 높을 수 있습니다.** (어차피 변수들이 비슷한 정보를 담고 있으므로) 하지만 "각 변수가 얼마나 중요한가"를 나타내는 β **계수 값을 신뢰할 수 없게** 됩니다. 따라서 '예측'만이 목표라면 큰 문제가 아닐 수 있지만, '해석'이 목표라면 반드시 해결해야 합니다.

Q: 왜 k개의 범주(예: 3개 인종)에 k개가 아닌 $k-1$개(2개)의 더미 변수를 쓰나요? A: k개를 모두 사용하면 완벽한 다중공선성(Dummy Variable Trap)이 발생합니다. 예를 들어 x_{Asian}, $x_{\text{Caucasian}}$, $x_{\text{AfricanAmerican}}$ 3개를 모두 만들면, $x_{\text{Asian}} + x_{\text{Caucasian}} + x_{\text{AfricanAmerican}} = 1$ 이라는 완벽한 선형 관계가 생깁니다. 이는 X 행렬의 역행렬 $(X^T X)^{-1}$을 계산할 수 없게 만듭니다. 따라서 하나를 기준선(Baseline)으로 제외하여 이 문제를 피합니다.

11 빠르게 훑어보기 (1-Page Summary)

기계 학습 3단계 프로세스

모든 지도 학습은 이 3단계를 따릅니다.

1. **모델 정의:** 데이터의 관계를 어떤 함수(예: 직선)로 가정할지 선택합니다.
2. **손실 함수 정의:** 모델의 예측이 실제와 얼마나 다른지(오차) 측정하는 기준(예: MSE)을 정합니다.
3. **손실 최소화:** 손실이 최소가 되는 모델의 파라미터(예: β)를 수학적 방법(예: 정규 방정식, 경사 하강법)으로 찾습니다.

단순 선형 회귀 (SLR): Y

- **목표:** 2D 평면에서 데이터를 가장 잘 표현하는 **직선**을 찾는다.
- **해석:** β_1은 X가 1단위 증가할 때 Y의 평균 변화량이다.
- **해법:** $\hat{\beta}_1 = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sum(X_i - \bar{X})^2}$, $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$

다중 선형 회귀 (MLR): Y

- **목표:** $p+1$ 차원 공간에서 데이터를 가장 잘 표현하는 **초평면(Hyperplane)**을 찾는다.
- **해석:** β_j는 **다른 모든 변수가 고정**되었을 때 X_j가 1단위 증가할 때 Y의 평균 변화량이다.
- **해법 (정규 방정식):** $\hat{\beta} = (X^T X)^{-1} X^T Y$ (이것이 `.fit()`의 핵심!)

모델 해석의 3대 함정

1. **스케일링 문제 (Apple vs Orange):** 단위(스케일)가 다른 변수들의 β 계수 크기는 직접 비교할 수 없다. → **해결: 표준화(Standardization) 후 비교**
2. **다중공선성 문제 (Clones):** 서로 상관관계가 높은 변수들은 β 계수 값을 불안정하게 만든다. → **해결: 상관관계 높은 변수 중 하나를 제거**
3. **범주형 변수 문제 (Text):** 'Male'/'Female' 같은 텍스트는 0/1로 변환(더미 변수)해야 한다. → **해결: k개 레벨에 $k-1$개 더미 변수 사용**