Contents

Chapter 8: 다중 예측변수를 이용한 선형 회귀

학습 목표

Chapter 7에서 단일 예측변수를 사용한 선형 회귀 모델의 개념을 바탕으로, 이제 두 개 이상의 예측변수를 사용하는 **다중 선형 회귀**(multiple linear regression) 모델을 학습한다. 여러 설명변수들이 어떻게 상호작용하는지 고려함으로써, 예측변수와 반응변수 사이의 복잡한 관계를 발견할 수 있다. 여러 변수를 다룰 때의 한 가지 도전은 어떤 변수가 모델에 포함되어야 하는지 결정하기 어렵다는 점이다. 모델 구축은 광범위한 주제이며, 여기서는 **수정된** R^2(adjusted R^2) 값을 정의하고 활용하는 것으로 그 표면만 다룬다.

다중 회귀(multiple regression)는 단일 예측변수 회귀를 여전히 하나의 반응변수를 가지지만 많은 예측변수(x_1, x_2, x_3, ...)를 가지는 경우로 확장한다. 이 방법은 많은 변수들이 동시에 출력과 연결될 수 있는 시나리오에 의해 동기 부여된다.

우리는 P2P 대출 업체인 Lending Club의 대출 데이터를 고려할 것이다. 이 데이터셋은 Chapter 1에서 처음 접했다. 대출 데이터에는 대출 조건뿐만 아니라 차입자에 대한 정보도 포함되어 있다. 우리가 더 잘 이해하고자 하는 결과변수는 대출에 할당된 **이자율**(interest rate)이다.

데이터 소개: loans 데이터셋

데이터셋에는 10,000건의 대출에 대한 정보가 포함되어 있다.

표 8.1: loans 데이터셋의 처음 6행

interest_rate	verified_income	debt_to_income	credit_util	bankruptcy	term	credit_checks	issue_month
14.07	Verified	18.01	0.548	0	60	6	Mar-2018
12.61	Not Verified	5.04	0.150	1	36	1	Feb-2018
17.09	Source Verified	21.15	0.661	0	36	4	Feb-2018
6.72	Not Verified	10.16	0.197	0	36	0	Jan-2018
14.07	Verified	57.96	0.755	0	36	7	Mar-2018
6.72	Not Verified	6.46	0.093	0	36	6	Jan-2018

표 8.2: loans 데이터셋의 변수 설명

변수	설명
interest_rate	대출 이자율, 연간 백분율
verified_income	차입자의 소득 검증 상태: Verified, Source Verified, Not Verified
debt_to_income	부채 대 소득 비율
credit_util	신용 활용도 비율
bankruptcy	파산 기록 지시변수 (1: 있음, 0: 없음)
term	대출 기간 (개월)
issue_month	대출 발행 월
credit_checks	지난 12개월 신용 조회 횟수

8.1 지시변수와 범주형 예측변수

8.1.1 이진 범주형 변수

차입자의 파산 기록 여부로 이자율을 예측하는 선형 회귀 모델:

$$\widehat{\text{interest\_rate}} = 12.34 + 0.74 \times \text{bankruptcy}$$

표 8.3: 파산 여부에 따른 이자율 예측 모델

term	estimate	std.error	statistic	p.value
(Intercept)	12.34	0.05	231.49	<0.0001
bankruptcy1	0.74	0.15	4.82	<0.0001

예제 8.1: 파산 변수의 계수 해석

문제: 모델에서 파산 변수의 계수를 해석하라.

풀이:

기울기 0.74는 파산 기록이 있는 차입자가 없는 차입자보다 평균 0.74%포인트 더 높은 이자율을 받을 것으로 예측됨을 의미한다.

- 파산 기록 없음 (bankruptcy = 0): 예측 이자율 = 12.34%
- 파산 기록 있음 (bankruptcy = 1): 예측 이자율 = 12.34 + 0.74 = 13.08%

```
import numpy as np
from scipy import stats

# 예측 계산
intercept = 12.34
slope = 0.74

pred_no_bankruptcy = intercept + slope * 0
pred_bankruptcy = intercept + slope * 1

print("=== 파산 여부별 예측 이자율 ===")
print(f"파산 기록 없음: {pred_no_bankruptcy:.2f}%")
print(f"파산 기록 있음: {pred_bankruptcy:.2f}%")
print(f"차이: {slope:.2f}%포인트")
```

새로운 시각: 지시변수의 이해

지시변수(indicator variable) 또는 **더미변수**(dummy variable)는 범주형 변수를 수치적으로 표현하는 방법이다. 이진 범주(예: 예/아니오, 남/여)의 경우 하나의 지시변수만 필요하며, 한 범주에 1을, 다른 범주에 0을 할당한다.

지시변수의 장점: 1. 범주형 데이터를 회귀 분석에 직접 사용 가능 2. 계수 해석이 직관적: 두 그룹 간 평균 차이를 나타냄 3. 통계적 검정 수행 가능

8.1.2 다수준 범주형 변수

3수준 범주형 변수 `verified_income`으로 이자율 예측:

표 8.4: 소득 검증 상태별 이자율 예측 모델

term	estimate	std.error	statistic	p.value
(Intercept)	11.10	0.08	137.2	<0.0001
verified_income(Source Verified)	1.42	0.11	12.8	<0.0001
verified_income(Verified)	3.25	0.13	25.1	<0.0001

예제 8.2: 다수준 범주형 변수의 회귀 방정식

문제: 회귀 모델 방정식을 작성하라.

풀이:

$$\widehat{\text{interest\_rate}} = 11.10 + 1.42 \times \text{verified\_income}_{\text{Source Verified}} + 3.25 \times \text{verified\_income}_{\text{Verified}}$$

기준 수준(reference level): Not Verified (테이블에 나타나지 않음)

각 수준별 예측: - Not Verified: 11.10% - Source Verified: 11.10 + 1.42 = 12.52% - Verified: 11.10 + 3.25 = 14.35%

```
# 각 수준별 예측 이자율
intercept = 11.10
coef_source = 1.42
coef_verified = 3.25

levels = {
    'Not Verified': intercept,
    'Source Verified': intercept + coef_source,
    'Verified': intercept + coef_verified
}

print("=== 소득 검증 수준별 예측 이자율 ===")
for level, rate in levels.items():
    print(f"{level}: {rate:.2f}%")
```

새로운 시각: 기준 수준의 선택

k개의 수준을 가진 범주형 변수는 k-1개의 지시변수로 표현된다. 누락된 수준이 **기준 수준**이 되며, 다른 수준의 계수는 이 기준과의 차이를 나타낸다.

기준 수준 선택 시 고려사항: - 대조군이나 기본 조건을 기준으로 설정 - 가장 큰 표본 크기를 가진 수준을 기준으로 설정 - 해석의 용이성을 고려

8.2 모델에서의 다중 예측변수

여러 변수를 동시에 고려하는 **다중 회귀**(multiple regression) 모델:

$$\widehat{\text{interest\_rate}} = b_0 + b_1 \times x_1 + b_2 \times x_2 + \cdots + b_k \times x_k$$

표 8.5: 전체 회귀 모델 출력

term	estimate	std.error	statistic	p.value
(Intercept)	1.89	0.21	9.01	<0.0001
verified_income(Source Verified)	1.00	0.10	10.06	<0.0001
verified_income(Verified)	2.56	0.12	21.87	<0.0001
debt_to_income	0.02	0.00	7.43	<0.0001
credit_util	4.90	0.16	30.25	<0.0001
bankruptcy1	0.39	0.13	2.96	0.0031
term	0.15	0.00	38.89	<0.0001
credit_checks	0.23	0.02	12.52	<0.0001
issue_month(Jan-2018)	0.05	0.11	0.42	0.6736
issue_month(Mar-2018)	-0.04	0.11	-0.39	0.696

예제 8.3: 전체 모델 방정식

적합된 모델:

$$\begin{aligned}\widehat{\text{interest\_rate}} = {} & 1.89 + 1.00 \times \text{verified\_income}_{\text{Source}} + 2.56 \times \text{verified\_income}_{\text{Verified}} \\ & + 0.02 \times \text{debt\_to\_income} + 4.90 \times \text{credit\_util} \\ & + 0.39 \times \text{bankruptcy} + 0.15 \times \text{term} + 0.23 \times \text{credit\_checks} \\ & + 0.05 \times \text{issue\_month}_{\text{Jan}} - 0.04 \times \text{issue\_month}_{\text{Mar}}\end{aligned}$$

Guided Practice 8.6: credit_checks 계수 해석

문제: credit_checks 변수의 계수를 해석하라.

풀이: 다른 모든 조건이 동일할 때, 지난 12개월 동안 신용 조회가 1회 추가될 때마다, 대출 이자율이 평균 **0.23%포인트** 더 높을 것으로 예상한다.

Guided Practice 8.7: 첫 번째 관측치의 잔차

문제: 표 8.1의 첫 번째 관측치의 잔차를 계산하라.

풀이:

첫 번째 관측치: Verified, debt_to_income=18.01, credit_util=0.548, bankruptcy=0, term=60, credit_checks=6, issue_month=Mar-2018

$$\hat{y}_1 = 1.89+1.00(0)+2.56(1)+0.02(18.01)+4.90(0.548)+0.39(0)+0.15(60)+0.23(6)+0.05(0)-0.04($$

$$\hat{y}_1 = 1.89 + 2.56 + 0.36 + 2.69 + 9.00 + 1.38 - 0.04 = 17.84$$

잔차: $e_1 = 14.07 - 17.84 = -3.77$

```
# 잔차 계산
y_obs = 14.07
y_pred = 1.89 + 2.56 + 0.02*18.01 + 4.90*0.548 + 0.15*60 + 0.23*6 - 0.04
residual = y_obs - y_pred

print(f"관측값: {y_obs}%")
print(f"예측값: {y_pred:.2f}%")
print(f"잔차: {residual:.2f}%포인트")
```

새로운 시각: 공선성과 계수 변화

단일 모델에서 bankruptcy 계수는 0.74였지만, 다중 모델에서는 0.39로 감소했다. 이는 **공선성**(collinearity) 때문이다.

단일 모델에서는 다른 변수를 통제하지 않아 파산의 효과에 다른 요인들의 영향이 혼합되어 있었다. 다중 모델에서는 다른 변수들을 통제하여 파산의 "순수한" 효과만 추정한다.

8.3 수정된 R-제곱

일반 R^2:

$$R^2 = 1 - \frac{Var(e_i)}{Var(y_i)}$$

수정된 R-제곱:

$$R^2_{adj} = 1 - \frac{s^2_{\text{residuals}}}{s^2_{\text{outcome}}} \times \frac{n-1}{n-k-1}$$

여기서 n은 관측치 수, k는 예측변수 수이다.

Guided Practice 8.9-8.10: R^2 계산

잔차 분산 = 18.53, 결과 분산 = 25.01, n = 10000, k = 9

$$R^2 = 1 - \frac{18.53}{25.01} = 0.2591$$

$$R^2_{adj} = 1 - \frac{18.53}{25.01} \times \frac{9999}{9990} = 0.2584$$

```
var_residuals = 18.53
var_outcome = 25.01
n, k = 10000, 9
```

```
r2 = 1 - var_residuals / var_outcome
r2_adj = 1 - (var_residuals / var_outcome) * ((n-1) / (n-k-1))

print(f"$R^2$ = {r2:.4f}")
print(f"$R^2$_adj = {r2_adj:.4f}")
```

새로운 시각: 왜 수정된 R^2가 필요한가?

일반 R^2는 예측변수를 추가하면 항상 증가하거나 동일하다. 이는 무의미한 변수를 추가해도 R^2가 감소하지 않음을 의미한다. 수정된 R^2는 예측변수 수에 대한 페널티를 부과하여 이 문제를 해결한다.

8.4 모델 선택

8.4.1 단계적 선택

후진 제거법(Backward elimination): 전체 모델에서 시작하여 변수를 하나씩 제거

전진 선택법(Forward selection): 빈 모델에서 시작하여 변수를 하나씩 추가

표 8.6: 전체 모델 (R^2_adj = 0.2597) 표 8.7: issue_month 제외 모델 (R^2_adj = 0.2598)

issue_month 제외 시 R^2_adj가 증가하므로 이 변수를 제거한다.

새로운 시각: 모델 선택의 한계

단계적 선택의 한계: 1. 후진 제거와 전진 선택이 다른 결과를 낼 수 있음 2. 국소 최적해에 빠질 수 있음 3. 모든 가능한 모델을 고려하지 않음

대안적 접근: - 전문가 지식 기반 변수 선택 - AIC/BIC 기준 - 교차 검증

8.5 장 요약

다중 선형 회귀는 여러 예측변수를 동시에 고려하여 결과를 예측한다. 각 계수는 다른 변수가 일정할 때 해당 예측변수의 효과를 나타낸다. 다중공선성은 해석을 복잡하게 만들 수 있으며, 수정된 R^2는 모델 비교에 유용한 도구이다.

주요 용어: - **수정된 R-제곱**(adjusted R-squared) - **후진 제거법**(backward elimination) - **공선성**(collinear) - **자유도**(degrees of freedom) - **전진 선택법**(forward selection) - **전체 모델**(full model) - **다중공선성**(multicollinearity) - **다중 회귀**(multiple regression) - **간명한**(parsimonious) - **기준 수준**(reference level) - **단계적 선택**(stepwise selection) —

8.6 연습문제

홀수 번호 연습문제의 상세 풀이

연습문제 8.1: 높은 상관관계, 좋은 것인가 나쁜 것인가?

문제: Frances는 선형 모델을 구축할 때 데이터셋의 모든 변수들이 서로 높은 상관관계를 가지는 것이 바람직하다고 주장한다. Annika는 각 예측변수가 결과와 높은 상관관계를 가지는 것은 바람직하지만, 예측변수들끼리 서로 높은 상관관계를 가지는 것은 바람직하지 않다고 주장한다. 누가 맞는가?

상세 풀이:

Annika가 맞다.

이유:

1. **예측변수-결과 상관관계**: 예측변수가 결과와 높은 상관관계를 가지면 예측력이 좋다. 이는 바람직하다.
2. **예측변수 간 상관관계 (다중공선성)**: 예측변수들이 서로 높은 상관관계를 가지면 **다중공선성**(multicollinearity 발생한다:
    - 계수 추정치 불안정
    - 표준 오차 증가
    - 개별 효과 분리 어려움

```
import numpy as np
from sklearn.linear_model import LinearRegression

# 다중공선성 시연
np.random.seed(42)
n = 100

# 독립적인 예측변수
x1_indep = np.random.normal(0, 1, n)
x2_indep = np.random.normal(0, 1, n)
y = 2 + 3*x1_indep + 2*x2_indep + np.random.normal(0, 1, n)

# 상관된 예측변수
x1_corr = np.random.normal(0, 1, n)
x2_corr = 0.95 * x1_corr + 0.05 * np.random.normal(0, 1, n)

print("독립 변수 간 상관:", np.corrcoef(x1_indep, x2_indep)[0,1])
print("상관 변수 간 상관:", np.corrcoef(x1_corr, x2_corr)[0,1])
```

연습문제 8.3: 육류 소비와 기대 수명

문제: (a) 육류 소비와 기대 수명 사이의 관계를 설명하라. (b) 왜 양의 상관관계가 있는가? (c) 소득별로 분리하면 관계가 더 강한가, 약한가?

상세 풀이:

(a) 관계 설명: 전체적으로 **양의 상관관계**가 있다. 육류 소비가 증가할수록 기대 수명도 증가하는 경향이 있다.

(b) 양의 상관관계 이유: 교란변수 때문이다. **소득 수준**이 둘 다에 영향을 미친다: - 부유한 국가: 더 많은 육류 소비 + 더 나은 의료 = 높은 기대 수명 - 가난한 국가: 적은 육류 소비 + 제한된 의료 = 낮은 기대 수명

이것은 **허위 상관**(spurious correlation)일 수 있다.

(c) 소득별 비교: 관계가 **더 약해진다**. 결합된 데이터에서 보이는 강한 관계의 대부분은 소득 그룹 간 차이 때문이다. 각 소득 그룹 내에서 관계는 덜 뚜렷하다.

이는 **심슨의 역설**(Simpson's Paradox)의 예이다.

```
import numpy as np
import matplotlib.pyplot as plt

# 시뮬레이션 데이터
np.random.seed(42)
income_groups = ['Low', 'Lower-middle', 'Upper-middle', 'High']

all_meat, all_life = [], []
for i, group in enumerate(income_groups):
    base_meat = 20 + i * 25
    base_life = 55 + i * 10
    meat = np.random.normal(base_meat, 15, 40)
    life = base_life + 0.05 * (meat - base_meat) + np.random.normal(0, 3, 40)
    all_meat.extend(meat)
    all_life.extend(life)

# 전체 vs 그룹 내 상관
r_total = np.corrcoef(all_meat, all_life)[0,1]
print(f"전체 상관: {r_total:.3f}")
print("그룹 내 상관은 훨씬 약함 → 소득이 교란변수")
```

연습문제 8.5: 5K 훈련

문제: days_since_start, days_till_race, mood, tiredness, time 중 모든 변수를 모델에 포함해야 하는가?

상세 풀이:

아니오. `days_since_start`와 `days_till_race`는 **완전 공선성**이 있다:

$$\text{days\_since\_start} + \text{days\_till\_race} = 30$$

따라서: - 하나를 알면 다른 하나가 완전히 결정됨 - 두 변수 모두 포함하면 계수를 추정할 수 없음 - 소프트웨어가 오류 발생 또는 자동 제거

해결책: 둘 중 하나만 모델에 포함해야 한다.

```python
import numpy as np

days_since_start = np.arange(1, 31)
days_till_race = 30 - days_since_start

print(f"상관계수: {np.corrcoef(days_since_start, days_till_race)[0,1]:.4f}")
print("→ -1.0 = 완전한 음의 선형 관계 (완전 공선성)")
```

연습문제 8.7: 아기 체중과 흡연

문제: (a) 회귀 방정식을 작성하라. (b) 기울기를 해석하고 예측값을 계산하라.

term	estimate	std.error	statistic	p.value
(Intercept)	7.270	0.0435	167.22	<0.0001
habitsmoker	-0.593	0.1275	-4.65	<0.0001

상세 풀이:

(a) 회귀 방정식:

$$\widehat{\text{weight}} = 7.270 - 0.593 \times \text{habit}_{\text{smoker}}$$

(b) 기울기 해석: 흡연자 산모에게서 태어난 아기는 비흡연자 산모의 아기보다 평균 **0.593파운드(약 269g) 더 가벼움**.

예측: - 비흡연자: 7.270 파운드 (약 3,297g) - 흡연자: 7.270 - 0.593 = 6.677 파운드 (약 3,028g)

```python
intercept, slope = 7.270, -0.593

pred_nonsmoker = intercept
pred_smoker = intercept + slope

print(f"비흡연자 산모: {pred_nonsmoker:.3f} 파운드")
print(f"흡연자 산모: {pred_smoker:.3f} 파운드")
print(f"차이: {abs(slope):.3f} 파운드 ({abs(slope)*453.6:.0f}g)")
```

연습문제 8.9: 영화 수익률

문제: (a) 가장 높은 ROI를 가진 장르는? (b) 제작 예산을 추가해야 하는가?

term	estimate
(Intercept)	-156.04
release_year	0.08
genreAdventure	0.30

term	estimate
genreComedy	0.57
genreDrama	0.37
genreHorror	8.61

상세 풀이:

(a) 가장 높은 ROI 장르: 기준: Action (테이블에 없음)

장르	계수 (Action 대비)
Action	0
Adventure	+0.30
Comedy	+0.57
Drama	+0.37
Horror	**+8.61**

Horror가 가장 높은 ROI를 가진다.

(b) 예산 추가 여부: - 추가 전: R^2_adj = 10.71% - 추가 후: R^2_adj = 10.84% - 개선: 0.13%포인트

추가하지 않는 것이 좋다. 개선이 너무 작아 실질적 의미가 없다.

```
genres = ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror']
coefs = [0, 0.30, 0.57, 0.37, 8.61]

for g, c in zip(genres, coefs):
    print(f"{g}: +{c:.2f}")
print(f"\n최고 ROI: Horror (+8.61)")
```

연습문제 8.11: 아기 체중 다중 예측

문제: (a) 방정식 작성 (b) weeks, habit 해석 (c) 계수 차이 이유 (d) 잔차 계산

term	estimate
(Intercept)	-3.82
weeks	0.26
mage	0.02
sexmale	0.37
visits	0.02
habitsmoker	-0.43

상세 풀이:

(a) 방정식:

$$\widehat{\text{weight}} = -3.82 + 0.26 \times \text{weeks} + 0.02 \times \text{mage} + 0.37 \times \text{sex}_{\text{male}} + 0.02 \times \text{visits} - 0.43 \times \text{habit}_{\text{smoker}}$$

(b) 해석: - **weeks**: 다른 조건 동일 시, 임신 기간 1주 증가당 체중 0.26파운드 증가 - **habitsmoker**: 다른 조건 동일 시, 흡연자 산모의 아기가 0.43파운드 더 가벼움

(c) 계수 차이 이유: 단순 모델(-0.593) vs 다중 모델(-0.43) - 단순 모델: 다른 변수 미통제 → 교란변수 효과 포함 - 다중 모델: 다른 변수 통제 → "순수한" 흡연 효과

(d) 잔차 (첫 번째 관측치: weight=6.96, weeks=37, mage=34, sex=male, visits=14, habit=nonsmoker):

$$\hat{y} = -3.82 + 0.26(37) + 0.02(34) + 0.37(1) + 0.02(14) - 0.43(0) = 7.13$$
$$e = 6.96 - 7.13 = -0.17 \text{ 파운드}$$

```
y_pred = -3.82 + 0.26*37 + 0.02*34 + 0.37*1 + 0.02*14 - 0.43*0
residual = 6.96 - y_pred
print(f"예측값: {y_pred:.2f}")
print(f"잔차: {residual:.2f}")
```

연습문제 8.13: 후진 제거법

문제: 전체 모델 R^2_adj = 0.326. 각 변수 제외 시: - mature 제외: 0.321 - weeks 제외: 0.061 - visits 제외: 0.326 - gained 제외: 0.327 - sex 제외: 0.301

어떤 변수를 제거해야 하는가?

상세 풀이:

제외 변수	R^2_adj	변화
mature	0.321	-0.005
weeks	0.061	-0.265
visits	0.326	0.000
gained	**0.327**	**+0.001**
sex	0.301	-0.025

gained를 먼저 제거해야 한다. - gained 제외 시 R^2_adj가 가장 높음 (0.327) - 이 변수가 예측력에 기여하지 않음

```
baseline = 0.326
results = {'mature': 0.321, 'weeks': 0.061, 'visits': 0.326,
           'gained': 0.327, 'sex': 0.301}

for var, r2 in sorted(results.items(), key=lambda x: x[1], reverse=True):
    change = r2 - baseline
    print(f"{var}: {r2:.3f} ({change:+.3f})")
```

연습문제 8.15: 전진 선택법

문제: 단일 변수 모델의 R^2_adj: - mature: 0.002 - weeks: 0.300 - visits: 0.034 - gained: 0.021 - sex: 0.018 - habit: 0.021

어떤 변수를 먼저 추가해야 하는가?

상세 풀이:

변수	R^2_adj
weeks	**0.300**
visits	0.034
gained	0.021
habit	0.021
sex	0.018
mature	0.002

weeks를 먼저 추가해야 한다. - weeks가 가장 높은 R^2_adj (0.300) - 단독으로 체중 변동의 30% 설명 - 생물학적으로 타당: 임신 기간이 출생 체중의 주요 결정 요인

```
vars_r2 = {'mature': 0.002, 'weeks': 0.300, 'visits': 0.034,
           'gained': 0.021, 'sex': 0.018, 'habit': 0.021}

for var, r2 in sorted(vars_r2.items(), key=lambda x: x[1], reverse=True):
    print(f"{var}: {r2:.3f}")
print("\n→ weeks 먼저 추가 ($R^2$_adj = 0.300)")
```