모델 선택, 다항 회귀, 그리고 교차 검증

CS109a 핵심 정리 노트

CS109a 수강생

October 26, 2025

▣ **강의명:** CS109A: 데이터 과학 입문

▣ **주차:** Lecture 06

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

▣ **목적:** Lecture 06의 핵심 개념 학습

1 개요

이 문서는 머신러닝 모델을 만들고 평가하는 핵심 과정인 '모델 선택'에 대해 다룹니다.

모델의 성능은 학습 데이터가 아닌, **본 적 없는 새로운 데이터**로 평가해야 합니다. 이 과정에서 모델이 학습 데이터의 노이즈까지 암기하는 **'과적합'**이 가장 큰 문제입니다. 우리는 선형 회귀를 확장한 **'상호작용 항'**과 **'다항 회귀'**를 통해 더 복잡한 모델을 만들 수 있지만, 이는 과적합의 위험을 높입니다. '모델 선택'은 이 복잡성과 일반화 성능 사이의 균형점을 찾는 과정입니다. **'교차 검증(Cross-Validation)'**은 단일 검증 세트의 함정을 피하고 모델의 일반화 성능을 신뢰성 있게 추정하는 표준적인 방법입니다.

2 핵심 용어 정리

모델 선택과 평가 과정을 이해하기 위해 필수적인 용어들을 정리했습니다.

Table 1: 모델 선택 및 평가 핵심 용어

용어	쉬운 설명	원어	비고 (예시)
과적합	모델이 학습 데이터를 '암기'해버려서, 새로운 데이터에 대한 예측 성능이 떨어지는 현상.	Overfitting	시험 족보만 외우고 응용 문제를 못 푸는 학생.
일반화 오차	모델이 '처음 보는' 데이터에서 발생하는 오차. 이 오차를 최소화하는 것이 최종 목표.	Generalization Error	실전 모의고사 성적.
모델 선택	여러 모델 후보(예: 다항식 차수) 중에서 일반화 오차가 가장 낮을 것으로 기대되는 모델을 고르는 과정.	Model Selection	1차, 2차, 3차 함수 중 2차 함수를 선택.
하이퍼파라미터	모델이 학습하기 전에 '사람'이 미리 정해야 하는 값.	Hyperparameter	다항 회귀의 '차수(M)', KNN의 'K값'.
검증 세트	하이퍼파라미터 튜닝(모델 선택)을 위해 사용하는 데이터.	Validation Set	여러 모델을 테스트해보는 연습 문제지.
테스트 세트	모델 선택이 끝난 후, '단 한 번' 최종 성능을 보고하기 위해 사용하는 데이터. 절대 모델 선택에 사용하면 안 됨.	Test Set	최종 학기말 고사.
상호작용 항	한 예측 변수의 효과가 다른 예측 변수의 수준에 따라 달라지는 효과(시너지 효과).	Interaction Term	TV 광고 효과(X_1)가 라디오 광고(X_2)와 함께할 때 더 커지는 현상 (X_1X_2).
다항 회귀	X, X^2, X^3 등 예측 변수의 거듭제곱을 새로운 예측 변수처럼 사용해 비선형 관계를 학습하는 기법.	Polynomial Regression	$Y = \beta_0 + \beta_1 X + \beta_2 X^2$
잔차	모델의 '예측값'과 '실제값'의 차이. 모델이 잘 맞는지 진단하는 핵심 도구.	Residual	$e_i = y_i - \hat{y}_i$
동분산성	예측 변수 X의 값에 관계없이 잔차의 분산이 일정한 것. (선형 회귀의 중요 가정)	Homoscedasticity	잔차 그래프가 깔때기 모양이 아님.
이분산성	X가 커질수록 잔차의 변동 폭도 커지거나 작아지는 현상. (동분산성 가정이 깨진 상태)	Heteroscedasticity	잔차 그래프가 깔때기(fanning) 모양.
K-겹 교차 검증	학습 데이터를 K개의 '조각'으로 나눈 뒤, K-1개로 학습하고 1개로 검증하는 과정을 K번 반복.	K-Fold Cross-Validation	데이터를 5조각(Fold)으로 나눔.

3 모델 평가와 과적합의 문제

3.1 학습 오차(Training Error)의 한계

모델을 평가할 때 MSE(평균 제곱 오차)나 R^2(결정 계수) 같은 지표를 사용합니다. 하지만 모델을 학습시킨 **학습 데이터(Training Data)**로 계산된 오차(학습 오차)는 모델의 실제 성능을 보장하지 않습니다.

학습 오차는 믿을 수 없다

네 개의 서로 다른 데이터셋이 동일한 MSE=1을 가질 수 있습니다. 하지만 그래프를 보면 어떤 모델은 비선형 관계를 놓치고, 어떤 모델은 수직선을 잘못 학습하고, 어떤 모델은 특이점에 과도하게 영향을 받습니다.
단순히 MSE가 낮다고 해서 좋은 모델이라고 말할 수 없습니다.

3.2 일반화 오차(Generalization Error)의 중요성

우리의 진짜 목표는 모델이 **본 적 없는 새로운 데이터(Unseen Data)**에서도 잘 작동하도록 하는 것입니다. 이때 '새로운 데이터'에서 발생하는 오차를 **일반화 오차(Generalization Error)**라고 부릅니다. 모델 선택의 목표는 이 일반화 오차를 최소화하는 모델을 찾는 것입니다.

3.3 과적합(Overfitting)이란 무엇인가?

과적합(Overfitting)은 모델이 학습 데이터의 패턴(경향성)뿐만 아니라, 데이터에 포함된 사소한 노이즈(Noise)나 특이점(Outlier)까지 모두 '암기'해버리는 현상을 말합니다.

□ 예제: title

과적합된 모델은 마치 시험 범위의 모든 예제와 답을 통째로 암기한 학생과 같습니다.

- **학습 데이터 (시험 족보)**: 100점을 받습니다. (낮은 학습 오차)
- **새로운 데이터 (응용 문제)**: 암기한 내용과 조금이라도 다르면 전혀 풀지 못합니다. (높은 일반화 오차)

모델은 데이터의 '개념(Trend)'을 배워야지, 데이터 자체를 '암기(Noise)'하면 안 됩니다.

과적합은 모델이 필요 이상으로 복잡할 때 발생합니다.

- 예측 변수(Feature)가 너무 많을 때
- 다항 회귀의 차수(Degree)가 너무 높을 때
- 상호작용 항이 너무 많을 때

3.4 모델 해석(Interpretation)의 함정

모델의 성능 지표(MSE)가 좋아 보여도, 반드시 모델의 계수(Coefficient)를 해석하여 상식에 맞는지 확인해야 합니다.

□ 예제: title

TV 광고 예산(X)과 매출(Y)의 관계를 모델링한 두 가지 경우입니다.

- **사례 1:** $Y = -0.05X + 6.2$
- **문제**: 기울기가 음수(-0.05)입니다. 이는 TV 광고 예산을 늘릴수록 매출이 줄어든다는 뜻입니다. (상식에 맞지 않습니다. 데이터에 오류가 있거나, 모델이 잘못되었을 수 있습니다.)
- **사례 2:** $Y = 0.02X - 0.5$
- **문제**: 절편이 음수(-0.5)입니다. 이는 광고 예산이 0일 때(X=0) 매출이 음수가 된다는 뜻입니다. (마찬가지로 상식에 맞지 않습니다.)

단순히 숫자에만 의존하지 말고, 모델이 현실을 잘 설명하는지 항상 비판적으로 검토해야 합니다.

4 선형 회귀의 확장: 비선형성 다루기

단순 선형 회귀는 강력하지만 현실의 복잡한 데이터를 설명하기엔 한계가 있습니다. 더 복잡한 모델을 만들기 전에, 선형 회귀의 기본 가정부터 확인해야 합니다.

4.1 선형 회귀의 4가지 핵심 가정

우리가 사용하는 MSE(평균 제곱 오차) 손실 함수는 다음 4가지 가정을 암묵적으로 전제합니다.

1. **선형성(Linearity)**: 예측 변수와 반응 변수 간에 직선적인 관계가 있다.
2. **독립성(Independence)**: 각 데이터의 오차(잔차)는 서로 독립적이다. (MSE는 단순히 오차 제곱을 '더하기' 때문에 이 가정이 필요합니다.)
3. **등분산성(Homoscedasticity)**: 모든 데이터 포인트에서 오차의 분산이 동일하다. (MSE는 모든 오차에 '가중치'를 두지 않기 때문에 이 가정이 필요합니다.)
4. **잔차의 정규성(Normality of Residuals)**: 잔차가 정규분포를 따른다. (오차를 '제곱'하는 방식은 정규분포 가정과 통계적으로 연결됩니다.)

이 외에도 '예측 변수 X는 오차가 없다(Fixed X)', '예측 변수 간 상관관계가 높지 않다(No Multicollinearity)' 등의 가정이 있습니다.

4.2 진단 도구: 잔차 분석(Residual Analysis)

위의 가정이 맞는지 확인하는 가장 좋은 방법은 **잔차 분석**입니다. 잔차($e = Y - \hat{Y}$)를 X축에, 예측 변수(X)나 예측값($\hat{Y}$)을 Y축에 그려봅니다.

- **좋은 모델 (가정 만족)**: 잔차가 0을 기준으로 특별한 패턴 없이 무작위로 흩어져 있습니다. (White Noise처럼 보입니다.) 잔차의 히스토그램은 종 모양(정규분포)을 띕니다.
- **나쁜 모델 (선형성 위반)**: 잔차가 U자형, S자형 등 뚜렷한 패턴을 보입니다. 이는 모델이 데이터의 비선형적 경향을 놓치고 있다는 신호입니다.
- **나쁜 모델 (등분산성 위반)**: X가 커질수록 잔차의 변동 폭이 커지거나(깔때기 모양, Fanning) 작아집니다. 이는 이분산성(Heteroscedasticity)을 의미합니다.

잔차 분석의 핵심

"잔차 플롯에서 어떤 패턴이든 보인다면, 심지어 용이나 성(Dragons flying over castles)이 상상되더라도, 모델의 기본 가정이 위반되었을 가능성이 높습니다."

4.3 확장 1: 상호작용 항(Interaction Effect)

현실에서는 한 변수의 효과가 다른 변수에 따라 달라지는 **시너지 효과(Synergy Effect)**가 흔합니다. 이를 모델링하는 것이 **상호작용 항**입니다.

□ 예제: title

모델 A: 상호작용 항 없음 $balance = \beta_0 + \beta_1 \times income + \beta_2 \times student$

- 비학생(student=0): $balance = \beta_0 + \beta_1 \times income$
- 학생(student=1): $balance = (\beta_0 + \beta_2) + \beta_1 \times income$

해석: 학생과 비학생은 기본 잔액(절편)만 다를 뿐, 소득(income)이 1단위 증가할 때 잔액이 β_1 만큼 증가하는 **기울기는 동일**합니다. (두 개의 평행선)

모델 B: 상호작용 항 추가 $balance = \beta_0 + \beta_1 \times income + \beta_2 \times student + \beta_3 \times (income \times student)$

- 비학생(student=0): $balance = \beta_0 + \beta_1 \times income$
- 학생(student=1): $balance = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \times income$

해석: 학생과 비학생은 절편도 다르고(β_0 vs $\beta_0 + \beta_2$), 소득이 증가할 때의 **기울기도 다릅니다** (β_1 vs $\beta_1 + \beta_3$). 만약 $\beta_3 > 0$이라면, 학생은 소득이 증가할 때 비학생보다 대출 잔액이 더 가파르게 증가(더 많은 소비)한다는 의미입니다.

4.4 확장 2: 다항 회귀 (Polynomial Regression)

데이터가 명백한 곡선 형태일 때, **다항 회귀**를 사용합니다. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_M X^M$

다항 회귀의 "속임수": 왜 이것도 선형 회귀인가?

다항 회귀는 X와 Y의 관계는 비선형이지만, 통계적으로는 **다중 선형 회귀의 특수한 경우**입니다.

속임수(Trick): X^2를 $\tilde{X}_2$, X^3를 $\tilde{X}_3$라는 완전히 새로운 예측 변수로 취급합니다. $Y = \beta_0 + \beta_1 X_1 + \beta_2 \tilde{X}_2 + \cdots + \beta_M \tilde{X}_M$

이렇게 변환하고 나면, 이 모델은 각 계수($\beta_0, \beta_1, \beta_2, \ldots$)에 대해 선형입니다. 따라서 다중 선형 회귀를 푸는 것과 동일한 방식($\hat{\beta} = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T y$)으로 β 값들을 찾을 수 있습니다.

`sklearn`에서는 `PolynomialFeatures`로 $X, X^2, X^3 \ldots$ 항들을 만든 후, `LinearRegression`을 `fit`시키면 됩니다.

4.5 다항 회귀의 함정: 스케일링과 항의 개수

다항 회귀 사용 시 주의사항

1. **특성 스케일링(Feature Scaling) 필수**: X의 범위가 100만 되어도 X^{10}은 천문학적인 숫자가 됩니다. 이렇게 값의 범위 차이가 극심하면 컴퓨터가 β 값을 계산할 때 수치적으로 불안정해집니다. 다항 회귀 사용 전, `StandardScaler` 등을 사용해 X의 범위를 (평균=0, 표준편차=1)로 표준화하는 것이 좋습니다.
2. **`PolynomialFeatures`의 작동 방식**: `sklearn`의 `PolynomialFeatures`는 기본적으로 '1'(절편 항), '상호작용 항'($X_1 X_2$) 등을 모두 포함하여 생성합니다.
   - 이 도구가 '1'을 이미 만들었으므로, `LinearRegression`을 학습시킬 때 `fit_intercept=False` 옵션을 주어야 절편이 중복 계산되지 않습니다.

- 예측 변수가 여러 개일 때 불필요한 상호작용 항이 너무 많이 생겨 과적합을 유발할 수 있습니다.

다항 회귀의 차수 M을 몇으로 할지 정하는 것 자체가 **하이퍼파라미터 튜닝**이며, 모델 선택의 핵심 문제입니다.

- **M이 너무 낮으면 (예: 1차)**: 데이터의 곡선 트렌드를 못 잡아냄 (과소적합, Underfitting)
- **M이 너무 높으면 (예: 50차)**: 데이터의 모든 노이즈를 통과하는 구불구불한 선이 됨 (과적합, Overfitting)

5 최적의 모델 찾기: 모델 선택 (Model Selection)

모델 선택은 과소적합과 과적합 사이의 '최적점(Sweet Spot)'을 찾는 과정입니다.

5.1 데이터 3-분할: 학습, 검증, 테스트

이를 위해 데이터를 3가지 용도로 나눕니다.

데이터의 3가지 역할

- **학습 세트 (Training Set)**: 모델을 학습(훈련)시키는 데 사용. (모델의 β 계수들을 찾는 데 사용)
- **검증 세트 (Validation Set)**: 학습된 모델들 중 최적의 하이퍼파라미터(예: 다항식 차수 M)를 **선택**하는 데 사용.
- **테스트 세트 (Test Set)**: 모델 선택까지 모두 끝난 후, 우리가 선택한 최종 모델의 일반화 성능을 **보고**하기 위해 '단 한 번' 사용.

테스트 세트의 신성불가침 원칙

"테스트 세트를 사용해 모델을 선택하거나 튜닝하는 행위는 학계의 가장 큰 금기 중 하나입니다." (마치 모의고사 문제로 기말고사를 내는 것과 같습니다.) 테스트 세트는 모델 개발 과정에서 완전히 격리되어야 하며, 최종 성능 보고 시에만 사용해야 합니다.

5.2 모델 선택 방법론

예측 변수가 J 개 있을 때, 어떤 변수를 모델에 포함시킬지 고르는 방법입니다.

- **전체 탐색 (Exhaustive Search)**: J 개의 변수로 만들 수 있는 모든 조합(2^J 개)의 모델을 다 만들어보고, 검증 세트 성능이 가장 좋은 것을 고릅니다. 변수가 10개만 돼도 1024개, 20개면 100만 개가 넘어 현실적으로 불가능합니다.
- **탐욕적 알고리즘 (Greedy Algorithms)**: 매 순간 '지금 당장' 가장 좋아 보이는 선택을 하는 방식입니다.
  - **전진 선택법 (Forward Selection)**: 1. 아무 변수도 없는 모델(M_0)에서 시작. 2. J 개의 변수 중 1개를 추가했을 때 검증 오차가 가장 많이 줄어드는 변수를 추가 (M_1). 3. M_1 에 $J-1$ 개의 남은 변수 중 1개를 추가했을 때 검증 오차가 가장 많이 줄어드는 변수를 추가 (M_2). 4. ... 변수를 J 개 다 쓸 때까지 반복. 5. 만들어진 $M_0, M_1, \ldots, M_J$ 중에서 검증 오차가 가장 낮았던 모델을 최종 선택.
  - (이 외에 후진 제거법, 단계적 선택법 등이 있습니다.)
  - **장점**: 2^J 에 비해 $O(J^2)$ 수준으로 계산이 훨씬 빠릅니다.
  - **단점**: 최적의 조합을 놓칠 수 있습니다. (예: X_1 만 있을 때보다 X_2 만 있을 때가 더 나빠도, $X_1 + X_2$ 조합이 $X_1 + X_3$ 조합보다 훨씬 좋을 수 있습니다.)

5.3 하이퍼파라미터 튜닝과 검증 세트

다항 회귀의 차수 M을 찾는 것과 같은 하이퍼파라미터 튜닝이 모델 선택의 핵심입니다. $M = 1, 2, 3, \ldots, 10$까지의 모델을 모두 **학습 세트**로 학습시킨 뒤, 각 모델의 성능을 **검증 세트**로 평가합니다.

모델 복잡도(Degree)에 따른 오차 그래프

- **학습 오차 (Training MSE)**: 모델이 복잡해질수록(차수가 높아질수록) 학습 데이터를 더 잘 '암기'할 수 있으므로 **계속 감소**합니다.
- **검증 오차 (Validation MSE)**:
  - **과소적합 영역 (Underfitting)**: 차수가 낮으면 트렌드를 못 잡아 오차가 높습니다.
  - **최적점 (Best Model)**: 트렌드는 잘 잡고 노이즈는 무시하는 지점에서 오차가 가장 낮아집니다.
  - **과적합 영역 (Overfitting)**: 차수가 너무 높으면 노이즈까지 암기하기 시작해, '새로운' 검증 데이터에서는 오차가 다시 **증가**합니다.

우리는 이 검증 오차 그래프(U자형 커브)에서 **오차가 최소가 되는 지점(Minimum)**의 차수 M을 최적의 하이퍼파라미터로 선택합니다.

6 신뢰할 수 있는 평가: 교차 검증 (Cross-Validation)

6.1 단일 검증 세트의 문제점 (CV의 동기)

만약 데이터를 학습/검증 세트로 딱 한 번만 나눈다면, **검증 세트가 우연히** 특정 모델에 유리하게 뽑힐 수 있습니다.

□ 예제: title

데이터의 실제 트렌드는 3차 함수(노란색 선)에 가깝다고 가정해봅시다. 하지만 우리가 **우연히** 뽑은 검증 데이터(분홍색 점)가 1차 함수(녹색 선) 근처에 몰려있을 수 있습니다.
이 경우, 우리는 1차, 2차, 3차 모델을 검증 세트로 테스트한 후, "검증 오차가 가장 낮은 1차 모델이 최고다!"라고 잘못된 선택을 하게 됩니다.
이는 우리가 **학습 데이터에 과적합**되는 것을 피하려다, **검증 데이터에 과적합**되는 결과를 낳습니다.

6.2 K-겹 교차 검증 (K-Fold Cross-Validation) 절차

이 문제를 해결하기 위해, 데이터를 '여러 번' 다르게 쪼개서 검증하고 그 성능을 '평균'내는 것이 **교차 검증**입니다. (K-Fold CV가 표준입니다.)

전제: 테스트 세트는 미리 분리해두고 절대 사용하지 않습니다. **남은 학습+검증 데이터**를 가지고 다음을 수행합니다.

1. 전체 학습 데이터를 K개의 균등한 '조각(Fold)'으로 나눕니다. (보통 K=5 또는 K=10 사용)
2. K번의 반복(Iteration)을 수행합니다.
3. **1번째 반복**:
   - 1번 조각(Fold 1)을 **검증 세트**로 사용합니다.
   - 나머지 K-1개 조각(Fold 2, 3, 4, 5)을 **학습 세트**로 사용해 모델을 학습합니다.
   - 학습된 모델로 Fold 1을 예측하여 검증 오차(MSE_1)를 계산합니다.
4. **2번째 반복**:
   - 2번 조각(Fold 2)을 **검증 세트**로 사용합니다.
   - 나머지 K-1개 조각(Fold 1, 3, 4, 5)을 **학습 세트**로 사용해 모델을 학습합니다.
   - 학습된 모델로 Fold 2를 예측하여 검증 오차(MSE_2)를 계산합니다.
5. **... K번째 반복**까지 동일하게 수행합니다.
6. **최종 CV 점수**: K개의 검증 오차($MSE_1, \ldots, MSE_K$)를 **평균**냅니다. $CV(Model) = \frac{1}{K}\sum_{i=1}^{K} MSE_i^{val}$

모델 선택(예: 다항식 차수 M 찾기)을 할 때, $M=1, M=2, M=3$ 등 각 후보에 대해 이 K-Fold CV 과정을 모두 수행하고, **CV 점수가 가장 낮은** M을 최종 모델로 선택합니다.

6.3 LOOCV (Leave-One-Out Cross-Validation)

K-Fold CV의 극단적인 형태로, $K = N$ (데이터 포인트 개수)인 경우입니다.

- 총 N번의 반복을 수행합니다.
- 매 반복마다 데이터 1개만 검증 세트로 쓰고, 나머지 N-1개로 학습합니다.
- 장점: 편향(Bias)이 매우 낮습니다.
- 단점: N번이나 모델을 학습시켜야 하므로 계산 비용이 매우 비쌉니다.

6.4 구현: `sklearn`과 `neg_mean_squared_error`

`sklearn.model_selection.cross_validate` 함수를 사용해 교차 검증을 쉽게 수행할 수 있습니다.

Scoring: 왜 'Negative' MSE를 쓰는가?

`sklearn`의 교차 검증 및 튜닝 도구는 점수(Score)를 **최대화(Maximize)**하도록 설계되어 있습니다. (예: 정확도(Accuracy)는 높을수록 좋음)
하지만 MSE는 **최소화(Minimize)**해야 하는 '오차(Error)' 지표입니다. 따라서 MSE를 최소화하는 것은 **-MSE를 최대화**하는 것과 같습니다.
`cross_validate`의 `scoring` 매개변수에 'mse'가 아닌 **'neg_mean_squared_error'**를 전달해야 올바르게 작동합니다.

```
from sklearn.model_selection import cross_validate
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline

# 예: 차3 다항회귀모델파이프라인생성
model = make_pipeline(
    PolynomialFeatures(degree=3, include_bias=False),
    LinearRegression(fit_intercept=True)
    # 가PolynomialFeatures 을1 만들지않게 (include_bias=False)하고
    # 이LinearRegression 절편을찾게 (fit_intercept=True) 할수있음
    # 또는( 반대로설정 )
)

# 데이터 X, 와y 겹5-(cv=5) 교차검증수행
# 점수지표로  'neg_mean_squared_error' 사용
cv_results = cross_validate(
    model,
    X,
    y,
    cv=5,
    scoring="neg_mean_squared_error",
    return_train_score=True # 학습스코어도반환
)

# cv_results['test_score'] 에개의 5 음수() MSE 값이들어있음
# 이값들의평균을내고부호를바꾸면최종        CV 점수(MSE)가 됨
```

```
final_cv_mse = -cv_results['test_score'].mean()
```

Listing 1: sklearn을 사용한 교차 검증 (의사 코드)

7 빠르게 훑어보기 (1페이지 요약)

모델링 핵심 요약 카드

1. 목표: 일반화 (Generalization)

모델의 진짜 성능은 '처음 보는' 데이터(Unseen Data)에서 나옵니다. 이때의 오차(일반화 오차)를 최소화하는 것이 목표입니다. 학습 데이터 오차(Training Error)는 중요하지 않습니다.

2. 적: 과적합 (Overfitting)

모델이 너무 복잡해져서(예: 너무 높은 차수, 너무 많은 변수) 학습 데이터의 '노이즈'까지 암기하는 현상입니다. **증상**: 학습 오차는 매우 낮지만, 검증 오차(일반화 오차)는 매우 높습니다.

3. 확장: 비선형 모델링

- **상호작용 항 (X_1X_2)**: 한 변수의 효과가 다른 변수에 따라 달라지는 '시너지' 효과를 모델링합니다.
- **다항 회귀 (X, X^2, X^3)**: 데이터의 곡선 트렌드를 잡습니다. (주의: 스케일링 필수!)

4. 전략: 모델 선택 (Model Selection)

과소적합(너무 단순)과 과적합(너무 복잡) 사이의 균형점을 찾는 과정입니다.

- **데이터 3-분할**: **학습**(훈련), **검증**(모델 선택/튜닝), **테스트**(최종 보고).
- **방법**: 하이퍼파라미터(예: 차수 M)를 바꿔가며 **검증 세트** 오차(Validation MSE)가 U자 곡선을 그릴 때, 가장 낮은 지점을 선택합니다.

5. 무기: K-겹 교차 검증 (K-Fold CV)

단일 검증 세트는 '우연'에 의해 잘못된 모델을 선택할 수 있습니다. (검증 세트에 과적합) **해결**: 데이터를 K조각으로 나눠, K번의 (학습/검증)을 반복하고 그 오차를 **평균**냅니다. 이 CV 점수를 기준으로 하이퍼파라미터를 선택하는 것이 가장 신뢰할 수 있습니다.

8 초심자 FAQ

Q: 다항 회귀가 왜 '선형' 회귀인가요? 너무 헷갈립니다.

A: Y와 X의 관계(그래프)는 곡선(비선형)이 맞습니다. 하지만 모델을 수식으로 볼 때, $Y = \beta_0 + \beta_1 X + \beta_2 X^2$에서 우리가 찾아야 할 값은 $\beta_0, \beta_1, \beta_2$입니다. 이 **계수(Coefficient) β에 대해서는 덧셈으로만 연결**되어 있으므로 '계수에 대해 선형(Linear in parameters)'이라고 부릅니다. X^2을 $\tilde{X}$라는 새로운 변수로 보면 $Y = \beta_0 + \beta_1 X + \beta_2 \tilde{X}$가 되어 다중 '선형' 회귀와 형태가 똑같아집니다.

Q: K-겹 교차 검증에서 K는 몇으로 정해야 하나요?

A: 정답은 없지만, 관례적으로 **K=5 또는 K=10**을 가장 많이 사용합니다. K가 너무 작으면(예: K=2) 검증 데이터의 변동성이 커서 불안정하고, K가 너무 크면(예: K=N, 즉 LOOCV) 계산 시간이 매우 오래 걸립니다. 5 또는 10이 계산 비용과 추정치의 안정성 사이의 적절한 타협점으로 알려져 있습니다.

Q: 검증 세트와 테스트 세트가 뭐가 다른 건가요? 둘 다 '평가'하는 것 아닌가요?

A: 역할이 완전히 다릅니다.

- **검증 세트 (Validation Set)**: '모델을 고르기 위한' 평가 세트입니다. 마치 여러 벌의 옷(모델 후보)을 입어보고(테스트) 가장 잘 어울리는 옷(최적 모델)을 **'선택'**하는 과정입니다.
- **테스트 세트 (Test Set)**: '선택이 끝난 후' 최종적으로 한 번만 평가하는 세트입니다. 가장 잘 고른 옷을 입고 나가서 사람들(새로운 데이터)에게 **'평가 보고'**를 받는 과정입니다.

검증 세트로는 여러 모델을 반복적으로 테스트하지만, 테스트 세트로는 최종 선택된 단 하나의 모델만 테스트해야 합니다.

Q: 특성 스케일링(StandardScaler)은 언제나 필요한가요?

A: 항상 필수는 아니지만, 사용하는 것이 훨씬 안전합니다. 단순 선형 회귀($Y = \beta_0 + \beta_1 X$)에서는 스케일링이 결과에 영향을 주지 않습니다. 하지만 **다항 회귀**($X^2, X^3 ...$), **정규화 회귀**(Ridge, Lasso), **KNN, SVM, 신경망** 등 대부분의 고급 머신러닝 모델은 특성 간의 스케일 차이에 매우 민감합니다. 따라서 모델링 전 스케일링을 적용하는 것을 습관화하는 것이 좋습니다.