3. 데이터 탐색 및 전처리

1 데이터 탐색 및 전처리 개요

데이터 탐색 및 전처리의 중요성

데이터 탐색과 전처리는 지도 학습 모델의 성능을 결정하는 매우 중요한 단계입니다.

데이터 탐색은 기술 통계 분석과 시각화 등을 사용해 데이터를 이해하고 구체적인 분석 계획을 수립하는 단계입니다.

데이터 전처리는 모델링하기 적절한 형태로 데이터를 정제함으로써 해당 데이터로 학습했을 때 나올 수 있는 성능의 상한을 결정합니다.

기초 데이터 탐색

매우 간단한 기초 데이터 탐색을 통해 모델과 전처리 기법 등을 선택할 수 있습니다.

데이터 정합성 검토

- 데이터는 측정과 기록을 통해 수집되므로 측정 및 기록 오류 등으로 잘못된 값이 있을 수 있음
- 따라서 데이터에 잘못된 값이 있는지 코드북, 상태 공간, 최솟값과 최댓값 등을 활용해서 검토해야 함
- 논리적으로 이상한 값이 있다면 해당 값을 지우거나 원래 값을 추정해야 하지만, 완벽하게 원래 값을 추정할 수 없다면 삭제하는 것이 바람직함

데이터 크기

- 데이터 크기(모양) = (샘플 개수, 특징 개수)
- 샘플이 많을수록 과적합 가능성이 줄어들고 특징이 많을수록 과적합 가능성이 늘어남
- 통상적으로 샘플 개수가 특징 개수의 30배보다 적다면 특징이 상대적으로 많다고 할 수 있음
- 특징이 절대적으로 적다면 단순한 모델을 사용하는 것이 좋고, 특징이 절대적으로 많다면 데이터를 설명하는데 복잡한 모델이 필요함

특징의 유형

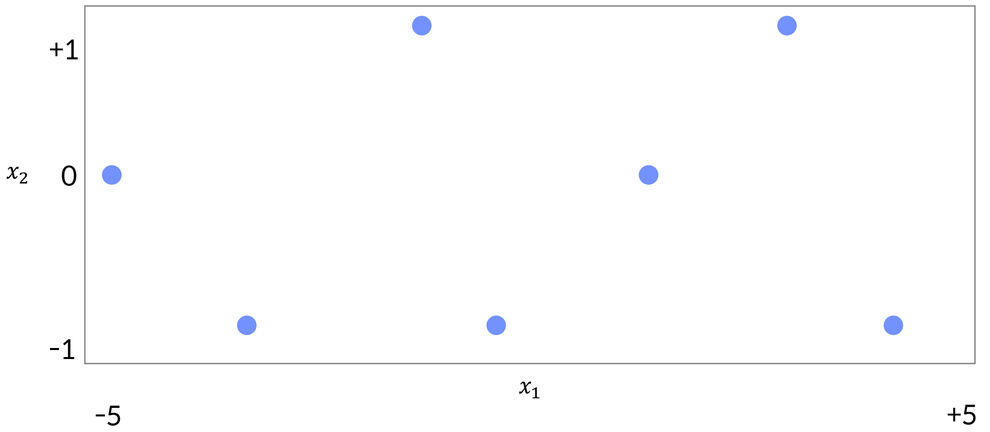
- 특징의 유형에 따라 특징이 차지하는 공간의 크기와 전처리하는 방법이 다름
- 일반적으로 연속형 변수가 범주형 변수보다 더 많은 공간을 차지함
- 범주형 특징은 더미화하지만, 연속형 특징은 절대로 더미화하지 않음

데이터 공간 (특징 공간)

데이터 공간 (특징 공간)이란 데이터가 차지하는 공간을 의미하며, 데이터 공간이 넓을수록 복잡한 모델이 필요합니다.

(예시)

- 특징 1 (x_1): -5 이상 5 미만의 연속형 변수
- 특징 2 (x_2): 상태 공간이 {-1, 0, 1}인 범주형 변수



특징 공간을 잘 이해해야 적절한 모델과 전처리 기법을 선택할 수 있음

3. 데이터 탐색 및 전처리

2 결측치 처리

결측치란?

결측치(missing value)란 누락된 값을 의미합니다.

ID	x_1	x_2	x_3
1	10	25	5
2	8	X	10
3	X	7	4
4	5	X	6
5	12	3	6

결측 발생

머신러닝 모델 대부분은 결측치가 포함된 데이터로 학습할 수 없으므로,

반드시 결측치를 제거하거나 원래 값을 추정해야 함

결측치의 종류

결측치는 None과 NaN으로 구분할 수 있으며, 처리 방법이 다릅니다.

None

- 값이 없는 것이 정상적인 결측치로, 엄밀한 의미에서 결측이라 보기 힘듦
- 예를 들어, 설문조사에서 직업을 물어보는 문항에 무직자는 응답할 수 없어 결측이 발생함
- 제거하거나 추정해야 하는 대상이 아니며, 새로운 값(예: “백수”, “무직자”)을 부여해야 하는 대상임

NaN

- 값이 있어야 하는데 없는 결측치로, 머신러닝 과제에서 관심을 갖는 결측치임
- 제거하거나 원래 값을 추정해야 함
- 이 강의에서 특별한 언급이 없으면 결측치는 NaN을 나타냄

결측치 제거

결측치 제거는 결측을 포함하는 행이나 열을 제거하는 것으로 가장 간단한 결측치 처리 방법입니다.

ID	x_1	x_2	x_3
1	10	25	5
2	8	X	10
3	X	7	4
4	5	X	6
5	12	3	6

결측 행 제거

ID	x_1	x_2	x_3
1	10	25	5
5	12	3	6

결측 열 제거

ID	x_3
1	5
2	10
3	4
4	6
5	6

결측치 제거의 장단점

결측치 제거는 매우 간단한 방법이지만, 결측치 제거에 따른 위험을 반드시 알아야 합니다.

장점

- 매우 간단함
- 결측치를 잘못 추정해서 발생하는 위험이 없음

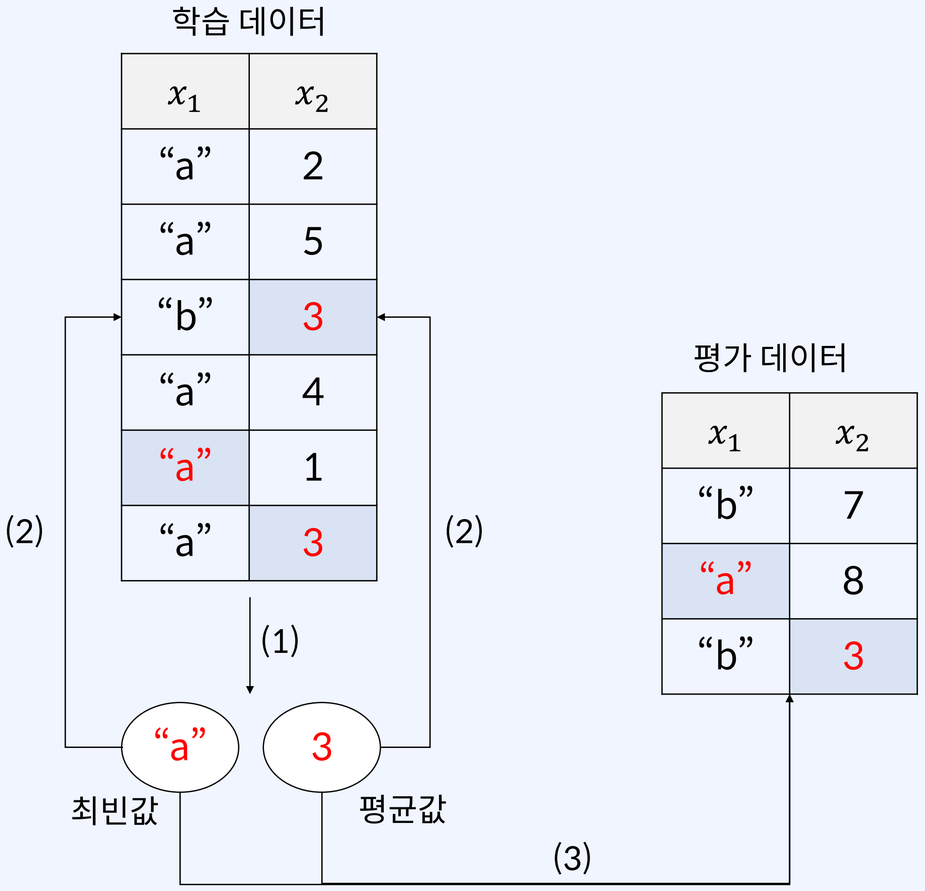
단점

- 결측을 포함하는 행이 많으면 지웠을 때 남는 샘플이 부족해져 과적합 위험이 높아짐
- 결측 행을 제거하고 만든 모델은 새로 입력된 데이터에 결측이 있으면 적절히 대처할 수 없음. 즉, 모델을 학습할 때 결측을 포함하는 행을 제거했으므로 해당 모델로 결측이 있는 행의 라벨을 예측할 수 없음
- 결측 열을 제거하면 소수의 결측 때문에 중요한 특징을 제거할 위험이 있음

- 결측을 포함하지 않는 샘플이 충분히 많고 새로 입력된 데이터에는 결측이 없으리라 판단할 수 있을 때만 결측 행을 제거해야 함
- 결측이 있는 열 가운데 결측 비율이 너무 높아서 활용이 어렵거나 도메인 지식을 바탕으로 중요하지 않다고 판단된 열만 삭제해야 함

통계량 기반의 결측치 추정

통계량 기반의 결측치 추정은 특징의 대푯값으로 결측을 추정하는 방법입니다.



(1) 대표 통계량 계산

- 학습 데이터에서 특징별 대푯값을 계산
- 범주형 변수에 대해서는 최빈값을 주로 계산
- 연속형 변수에 대해서는 평균값을 주로 계산

(2) 학습 데이터 결측 추정

- (1)에서 추정한 대푯값으로 학습 데이터의 결측을 추정

(3) 평가 데이터 결측 추정

- (1)에서 추정한 대푯값으로 평가 데이터의 결측을 추정
- 평가 데이터에서 다시 대푯값을 계산하지 않음에 주의

통계량 기반의 결측치 추정의 문제점

통계량을 이용한 결측치 추정은 가장 널리 많이 사용되는 결측치 추정 방법이지만, 특징 간 상관관계가 큰 데이터에는 적용하기 부적절할 수 있습니다.

$x_1 + x_2 = 1$이라는 관계 존재

x_1	x_2
0	1
0	1
0	1
0	1
1	X
1	0
0	1
X	0

통계량 기반의 결측치 추정

x_1	x_2
0	1
0	1
0	1
0	1
1	1
1	0
0	1
0	0

개별 특징의 통계량을 사용하므로 특징 간 관계를 완전히 무시하는 문제가 있음

k-최근접 이웃 모델을 이용한 결측치 추정

k-최근접 이웃 모델을 이용한 결측치 추정 방법은 특징 간 강한 상관관계가 있는 데이터에 적용하기 적절합니다.

학습 데이터

ID	x_1	x_2	x_3
a	1	7	3
b	2	4	7
c	1	5	4
d	1	7	3
e	3	2	2

(1) 거리 계산

	a	b	c	d	e
a	0.00	5.05	1.73	0.00	2.74
c	1.73	3.87	0.00	2.74	4.42
d	0.00	6.12	2.74	0.00	6.24

(2) 이웃 탐색

	이웃	값
a	d	7
c	a	1
d	a	1

(3) 대체

(1) 거리 계산

- 결측이 있는 샘플과 다른 샘플 간 거리를 계산
- 결측이 있는 샘플 간 거리는 결측 유클리디안 거리를 주로 사용

(2) 이웃 탐색

- (1)에서 계산한 거리를 바탕으로 결측이 있는 샘플에 대해 k개의 이웃을 탐색

(3) 평가 데이터 결측 추정

- 결측이 있는 샘플의 이웃의 값을 바탕으로 결측을 대체

결측 유클리디안 거리

결측 유클리디안 거리는 결측이 있는 두 샘플 간 거리를 측정합니다.

$$\sqrt{\frac{d}{d_{nan}} \sum_{x_i \,\&\, y_i \neq \mathrm{NaN}} (x_i - y_i)^2}$$

- d: **차원**
- d_{nan}: x**와** y **둘 중 하나라도 결측인 요소 개수**

(예시)

- $x = (3, nan, nan, 6)$
- $y = (1, nan, 4, 5)$

$$\sqrt{\frac{4}{2}\left((3-1)^2 + (6-5)^2\right)}$$

k-최근접 이웃 모델을 이용한 결측치 추정의 장단점

k-최근접 이웃 모델을 이용한 결측치 추정은 특징 간 상관성이 높을 때만 사용해야 합니다.

장점

- 특징 간 상관성이 높을 때 유효함

단점

- 특징이 서로 독립적이라면 사용하기 부적절함
- 정교하게 결측을 추정하려면, 특징 간 관계를 자세히 분석해야 함
- k-최근접 이웃 모델의 하이퍼 파라미터를 설정해야 함

보간

시계열 데이터의 결측은 보간(interpolation)을 사용해 추정하는 것이 적절합니다.

결측값의 실제 값은 정확히 알 수 없지만,
이전 값과 이후 값을 보면 2번 위치에 있을 가능성이 가장 높음

현실적으로 결측치가 발생한 바로 이전 시점의 값으로 결측을 대체하는 것이 일반적임

3. 데이터 탐색 및 전처리

3 범주형 변수 처리

범주형 변수 처리가 필요한 이유

지도 학습 모델 대부분은 수치 연산을 수행하므로 범주형 변수는 적절한 숫자로 변환해야 합니다.

지도 학습 모델 대부분은 입력한 특징을 사용해 다양한 수치 연산을 수행함

(예: 신경망은 입력된 특징의 가중합을 계산하고, 가중합에 활성화 함수(activation function)를 적용함)

수치 연산을 적용한다는 것은 입력된 특징이 숫자라 간주한다고 해석할 수 있음

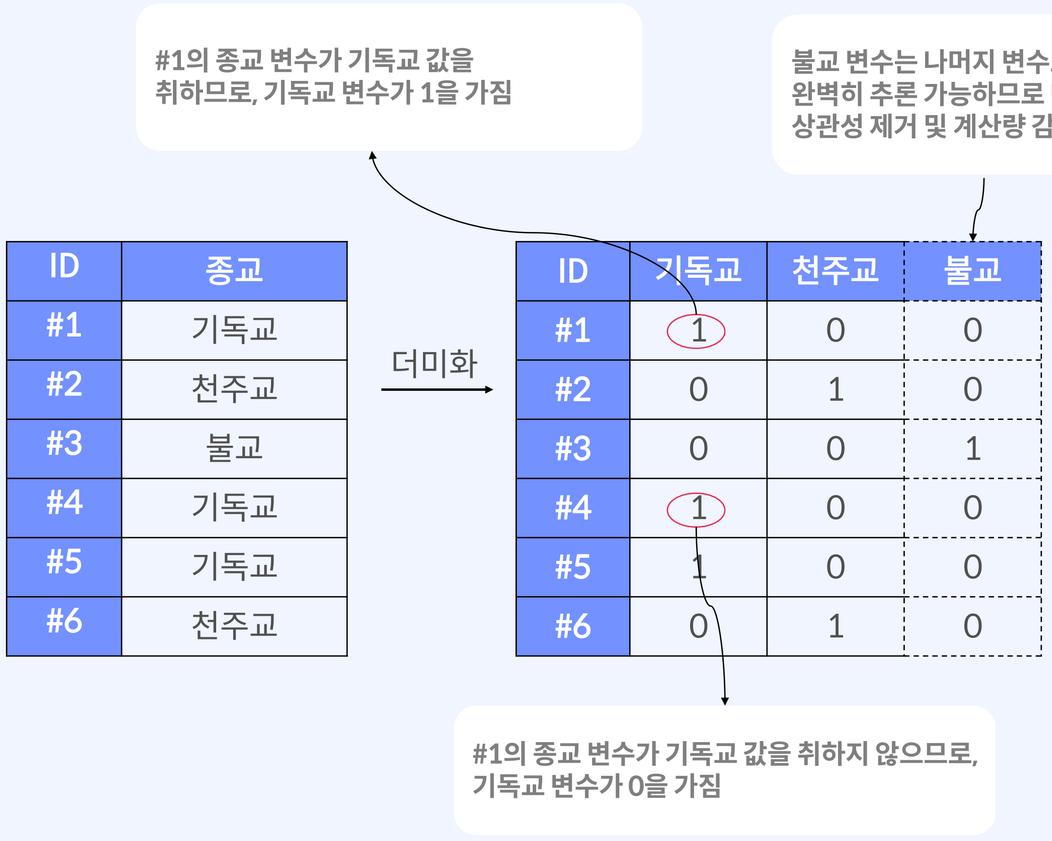
범주형 변수가 포함돼 있으면, 대다수의 지도 학습 모델이 학습되지 않거나 비정상적으로 학습됨

(문자로 표현된 범주형 변수: 학습되지 않음, 숫자로 표현된 범주형 변수: 비정상적으로 학습됨)

정상적으로 모델을 학습하고 활용하려면 범주형 변수를 반드시 적절한 숫자로 변환해야 하며, 임의로 값을 숫자로 바꾸면 안 됨

더미화

범주형 변수를 처리하는 가장 일반적인 방법인 더미화는 범주형 변수를 여러 개의 더미 변수(dummy variable)로 변환합니다.



- 더미 변수: 기존 변수가 특정 값을 갖는지를 나타내는 변수
- 더미 변수는 적절한 숫자일 뿐만 아니라, 범주형 변수가 갖고 있는 정보도 그대로 가짐
- 상태 공간이 큰 범주형 변수를 더미화하면 지나치게 많은 더미 변수가 추가돼 차원의 저주 문제로 이어질 수 있어, 상태 공간에 포함된 모든 값을 더미 변수로 변환하지 않고, 자주 출현한 값만 변환하기도 합니다

라벨을 활용한 치환

지도 학습 모델링에 한해 범주형 변수에 따른 라벨의 대푯값을 바탕으로 범주형 변수를 연속형 변수로 치환할 수 있습니다.

X	Y
A	100
B	200
A	300
C	400
B	500
C	600

X에 따른
Y의 평균 계산 →

X	Y
A	200
B	350
C	500

치환 →

X	Y
200	100
350	200
200	300
500	400
350	500
500	600

기존 변수가 가지는 정보가 일부 손실될 수 있고 상대적으로 구현하고 활용하기 어렵다는 단점이 있으나,
차원의 크기가 변하지 않으며 모델링에 적합한 변수로 변환할 수 있다는 장점이 있음

3. 데이터 탐색 및 전처리

4 분포 확인

스케일링: 개요

특징 간 스케일 차이가 크면 스케일링을 통해 특징의 스케일을 비슷하게 맞춰야 합니다.

최소-최대 스케일링

특징의 스케일을 [0, 1]로 조정하는 최소 – 최대 스케일링은 데이터의 분포를 가정하지 않는 모델에 적합합니다.

$$z_i = \frac{x_i - min(\boldsymbol{x})}{max(\boldsymbol{x}) - min(\boldsymbol{x})}$$

- $\boldsymbol{x}$: 스케일링할 벡터, $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)$
- $\boldsymbol{z}$: 스케일링된 벡터, $\boldsymbol{z} = (z_1, z_2, \cdots, z_d)$
- min: 최솟값 함수
- max: 최댓값 함수

데이터의 분포를 가정하지 않는 모델에 적합함

- 신경망
- k-최근접 이웃 등

표준화

표준화는 데이터의 분포를 가정하는 모델에 적합합니다.

$$z_i = \frac{x_i - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$$

- $\boldsymbol{x}$: 스케일링할 벡터, $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)$
- $\boldsymbol{z}$: 스케일링된 벡터, $\boldsymbol{z} = (z_1, z_2, \cdots, z_d)$
- μ: 평균 함수
- σ: 표준편차 함수

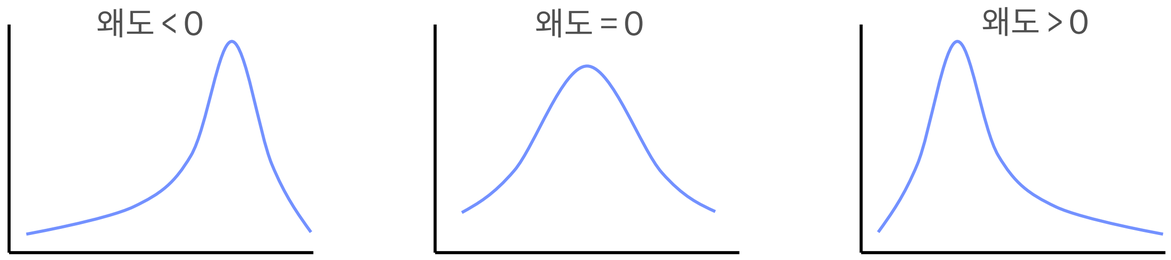
데이터의 분포를 가정하는 모델에 적합함

- 선형 회귀 모델
- Lasso
- Ridge 등

치우침 확인 및 제거

변수가 한 방향으로 치우쳐 있으면 치우치지 않은 방향에 있는 값이 이상치처럼 작용하거나 잔차가 정규 분포를 따르지 않을 위험이 있어,
치우침을 확인하고 제거해야 함

치우침 확인 방법: 왜도



- 그래프를 통해 치우침을 확인하는 것이 가장 좋으나, 현실적으로 힘듦
- 통상적으로 왜도의 절댓값이 1.5 이상이면 치우쳤다고 간주함

치우침 제거 방법: 변수 변환

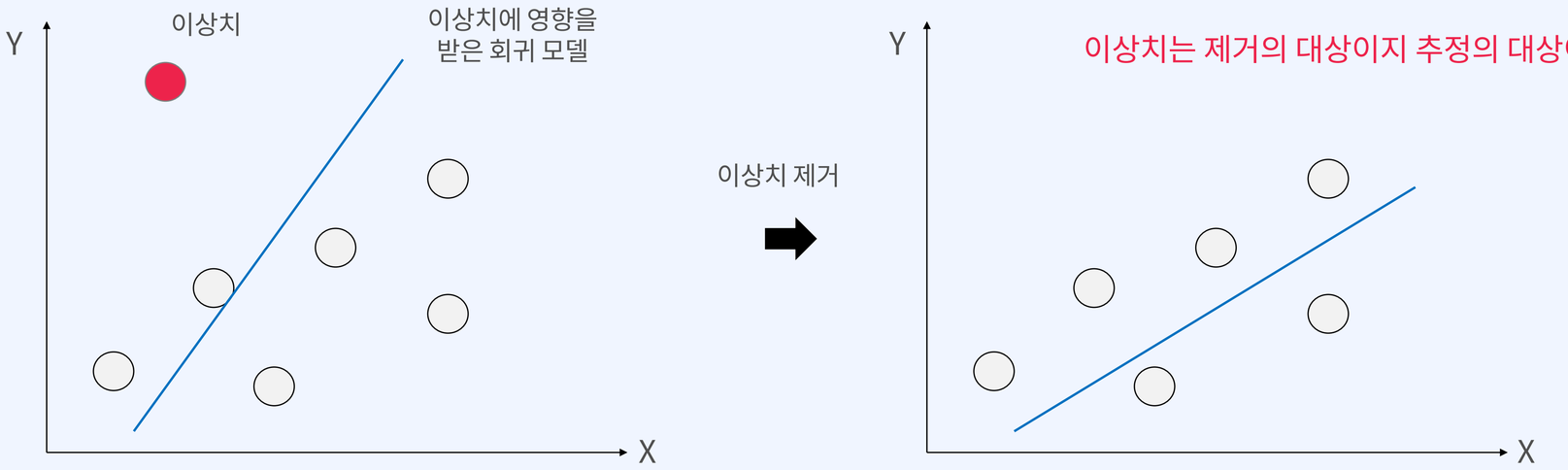
로그 변환 $z_i = \log(x_i - \min(\boldsymbol{x}) + 1)$

- 변수가 치우쳤다는 것은 치우친 방향의 값과 치우치지 않은 방향의 값 간 차이가 크다는 것을 뜻하므로,
  변수 치우침은 다양한 변수 변환 방법을 사용해 값 간 차이를 줄임으로써 해결할 수 있음
- 로그와 루트 모두 양수에 대해서만 정의되므로 최솟값을 빼고 1을 더했음

루트 변환 $z_i = \sqrt{(x_i - \min(\boldsymbol{x})) + 1}$

이상치란?

이상치(outlier)는 데이터 분포에서 많이 벗어난값으로 일반화된 모델을 생성하는 데 악영향을 끼칩니다.



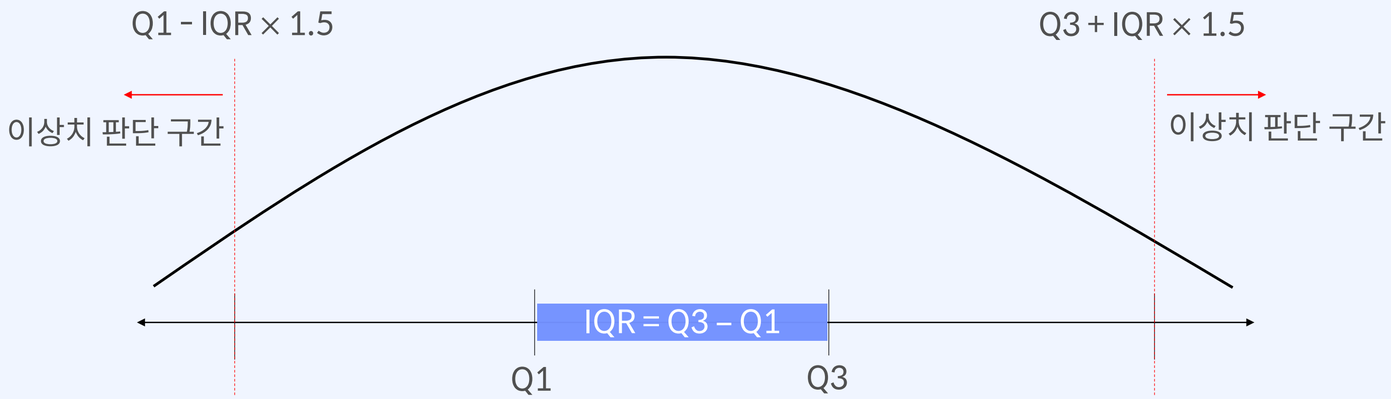
이상치는 제거의 대상이지 추정의 대상이 아님

결측은 원래 있어야 하는 값이 없는 것이므로 그 값을 추정하는 것이 논리적으로 가능하지만,

이상치는 다른 값들과는 다를 뿐이지 잘못된 값이 아니므로 원래 값을 추정하는 것이 논리적으로 이상함

IQR 규칙

IQR 규칙은 각 특징의 값이 [Q1-IQR * 1.5, Q3 + IQR * 1.5]를 벗어나면 이상치라 간주합니다.



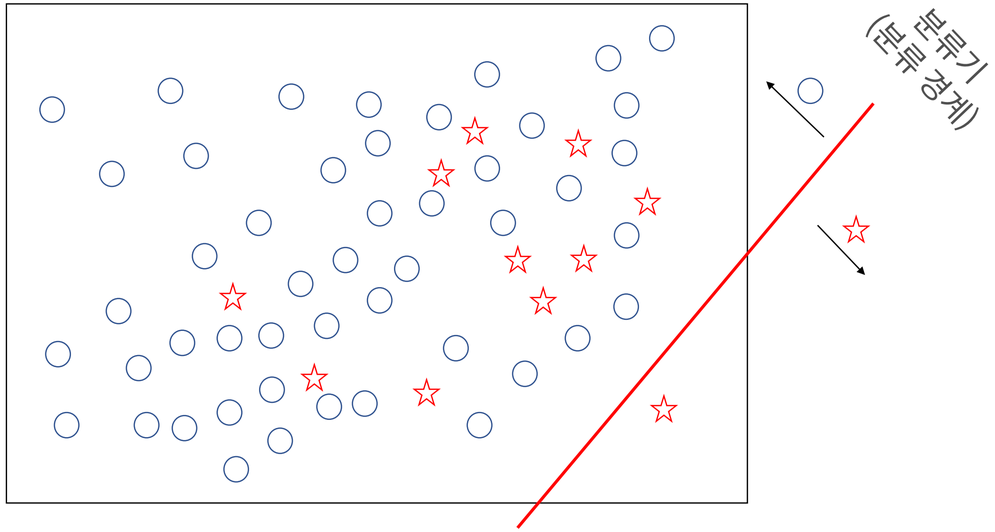
직관적이고 사용이 간편하지만, 단일 변수로 판단하기 어려운 이상치인 지역 이상치(local outlier)를 탐지하기 어렵다는 문제가 있음

3. 데이터 탐색 및 전처리

5 클래스 불균형 문제

클래스 불균형 문제

클래스 불균형 문제란 클래스 변수가 하나의 값에 치우친 데이터로 학습한 분류 모델이 치우친 클래스에 대해 편향되는 문제를 의미합니다.



- 편향된 모델은 빈도가 높은 클래스 값의 결정 공간이 비정상적으로 커서 대부분의 샘플을 하나의 클래스로 분류함
- 원(○)으로 분류되는 결정 공간이 별(☆)로 분류되는 공간보다 훨씬 넓음
- 실제 원에 속하는 샘플은 정 분류될 가능성이 크지만 별에 속하는 샘플은 오 분류될 가능성이 큼
- 클래스 불균형 문제가 있는 모델은 정확도가 높지만, 재현율과 f1 점수가 매우 낮은 경향이 있음

클래스 불균형 문제의 핵심은 클래스 변수가 불균형한 것 자체가 아니라, 한 클래스에 모델이 편향되게 하는 것임

클래스 불균형 문제의 주요 용어

다수 클래스	클래스 변수가 주로 갖는 값 (이전 그림에서 ◯ 클래스)
소수 클래스	클래스 변수가 주로 갖는 값이 아닌 값 (이전 그림에서 ☆ 클래스)
긍정 클래스	분류 시에 관심을 두는 클래스 (주로 소수 클래스인 경우가 많음)
부정 클래스	분류 시에 관심을 두지 않는 클래스 (주로 다수 클래스인 경우가 많음)

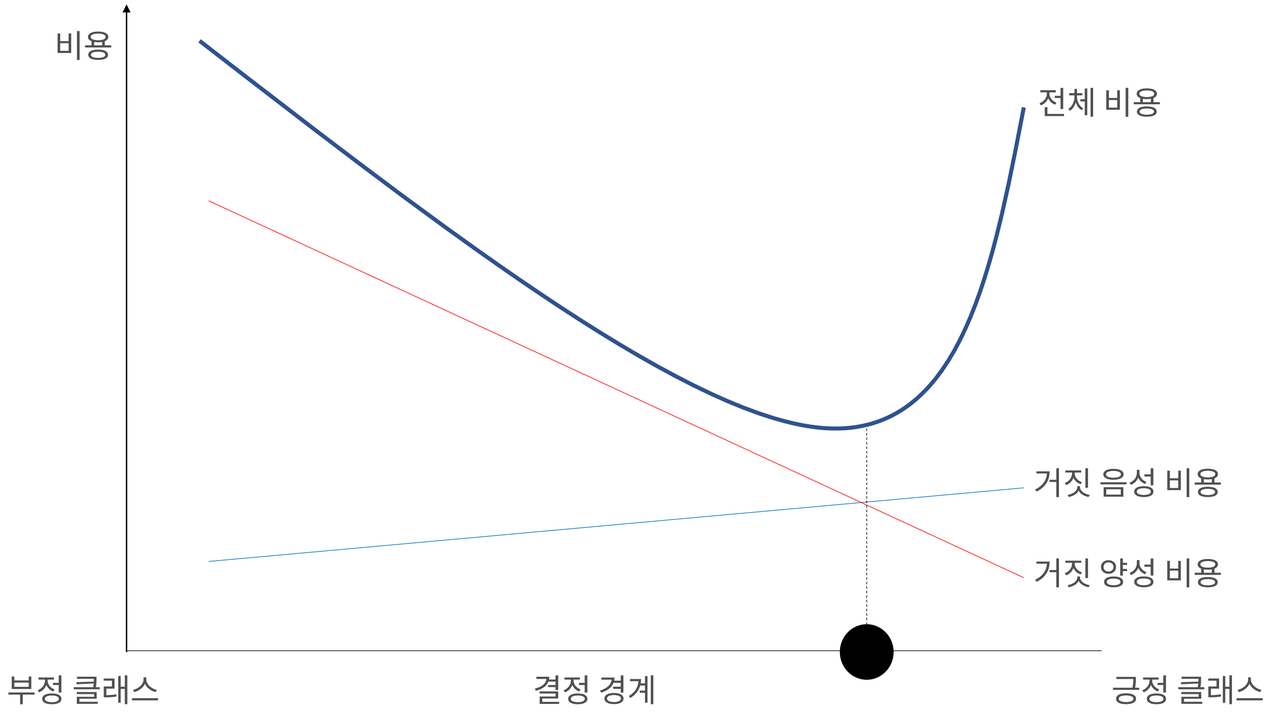
클래스 불균형 문제가 있는 모델은 대부분의 샘플을 다수 클래스로 분류해서 소수 클래스를 제대로 분류하지 못하는데,
소수 클래스가 긍정 클래스인 경우가 많아 문제가 됨

예시

- 암 환자를 판별하는 분류 문제에서 우리의 관심 대상은 암 환자며, 암 환자는 암 환자가 아닌 사람보다 훨씬 적음
- 제품의 불량을 판별하는 문제에서 우리의 관심 대상은 불량품이며, 불량품은 양품보다 훨씬 적음

클래스 불균형 문제의 주요 원인

클래스 불균형 문제의 근본적인 이유는 분류 모델의 손실 함수에서 거짓 양성 비용과 거짓 음성 비용을 구분하지 않기 때문입니다.

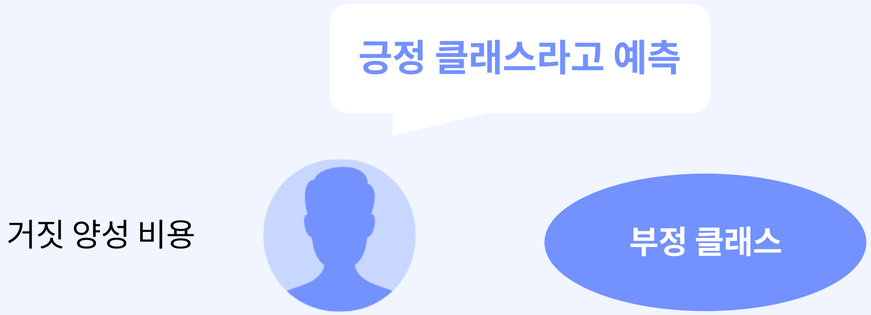


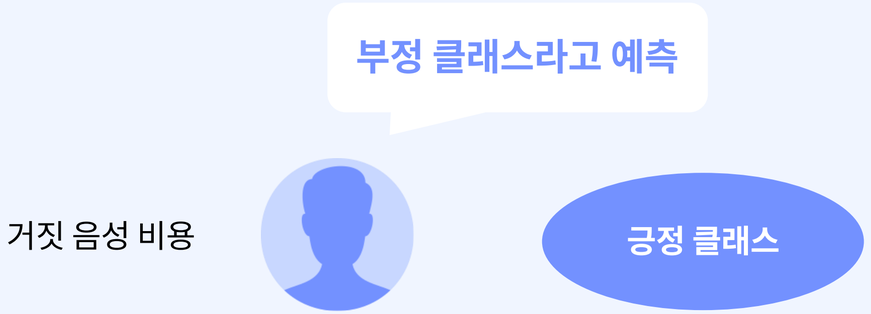
- 거짓 양성 비용: 부정 클래스를 긍정 클래스로 오 분류했을 때 발생하는 비용
- 거짓 음성 비용: 긍정 클래스를 부정 클래스로 오 분류했을 때 발생하는 비용
- 결정 경계가 긍정 클래스에 가까워지면 거짓 음성 비용은 증가하나 거짓 양성 비용은 줄어듦

거짓 음성 비용이 증가하는 양에 비해, 거짓 양성 비용이 감소하는 양이 훨씬 크므로 결정 경계가 긍정 클래스에 가까운 곳으로 학습됨

거짓 음성 비용과 거짓 양성 비용

현실적으로 거짓 음성 비용이 거짓 양성 비용보다 훨씬 크지만, 대부분의 분류 모델은 이를 고려하지 않습니다.





거짓 음성 비용 >> 거짓 양성 비용

- (예시) 암 환자를 암 환자가 아니라고 오 분류 하면 이 환자는 아무런 조치를 취하지 않다가 병세가 악화돼 죽음에 이르겠지만, 암 환자가 아닌 사람을 암 환자로 오 분류하면 이 사람은 기껏해야 추가로 검진을 받을 것임
- 거짓 음성 비용: 암 환자의 죽음
- 거짓 양성 비용: 추가 검진 비용

탐색 방법

클래스 불균형 문제는 데이터에 특정 클래스의 값이 많아 발생하므로 클래스 변수의 분포를 확인하면 클래스 불균형 문제가 있을지 알 수 있습니다.

클래스 불균형 비율

$$\frac{N(y = neg)}{N(y = pos)}$$

- $N(y = neg)$: 클래스 변수가 부정 클래스인 개수
- $N(y = pos)$: 클래스 변수가 긍정 클래스인 개수

통상적으로 클래스 불균형 비율이 9 이상이면 편향된 모델이 학습될 가능성이 크다고 봄

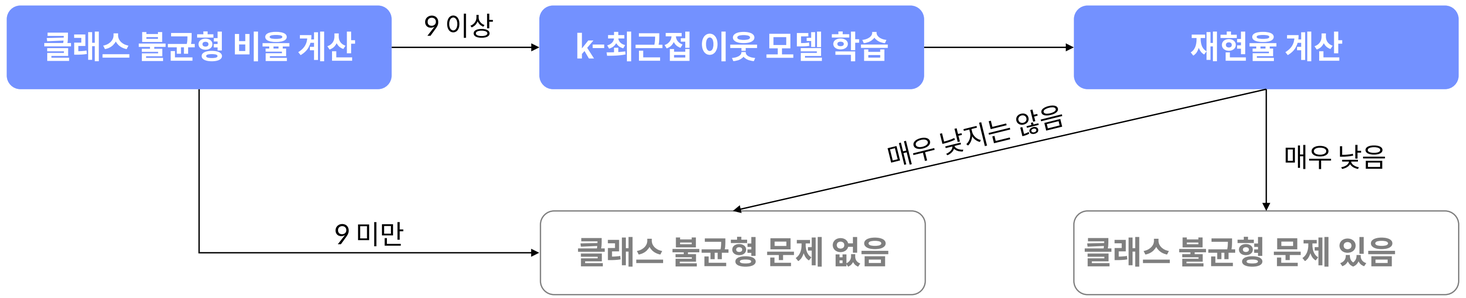
↓

그러나 클래스의 분포가 불균형하다고 항상 클래스 불균형 문제가 발생하는 것은 아님

↓

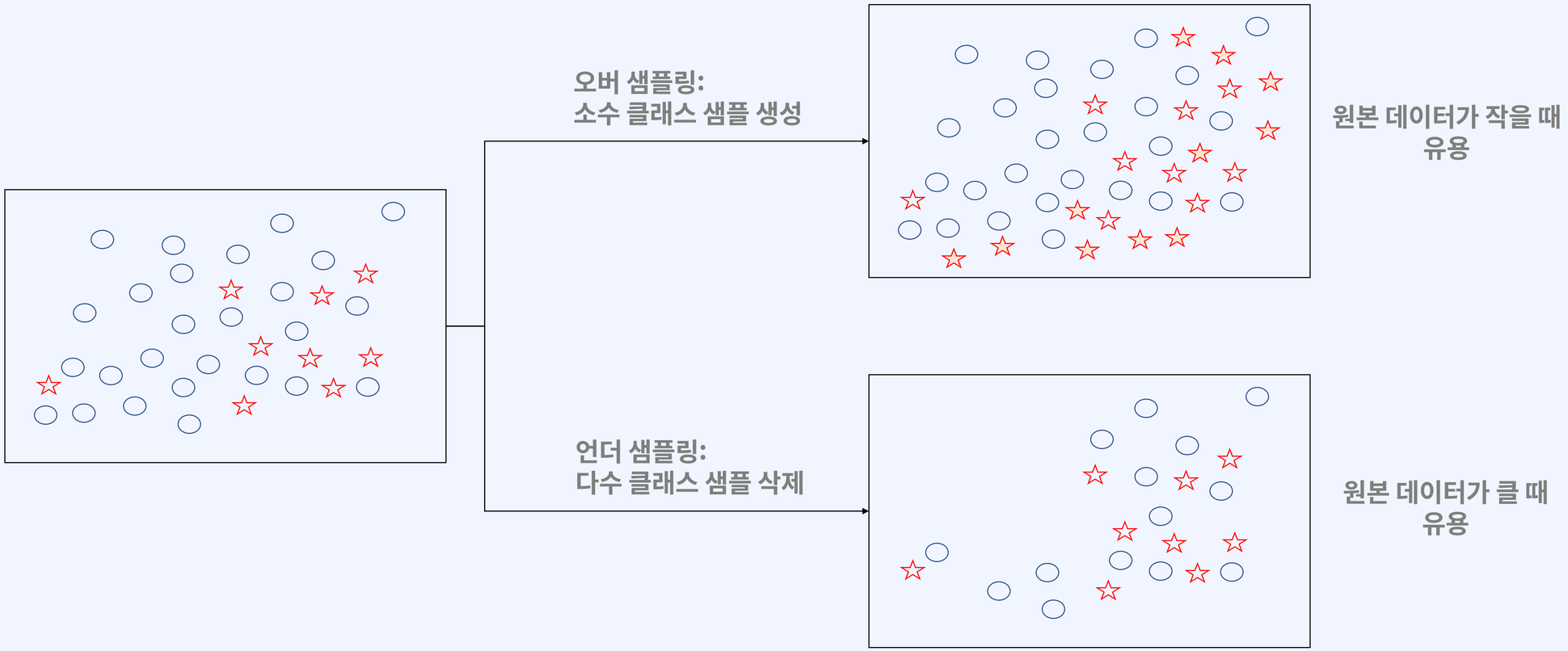
클래스 불균형 비율이 높으면, 클래스 불균형에 민감한 모델을 사용해서 다시 판단해야 함

↓



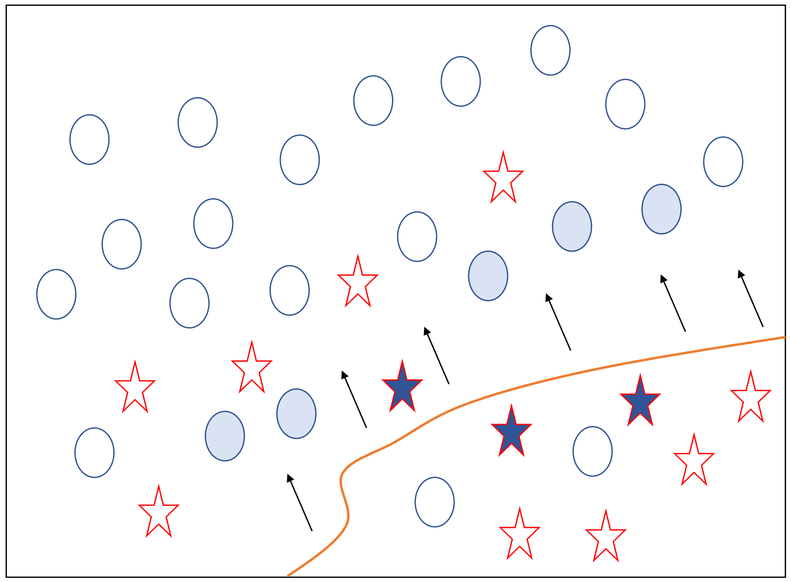
해법 1. 재샘플링

재샘플링을 통해 클래스 비율을 균형하게 맞춤으로써 클래스 불균형 문제를 해소할 수 있습니다.



해법 1. 재샘플링
기본 아이디어

클래스 불균형 문제는 결정 경계가 지나치게 긍정 클래스에 가까워 발생하므로, 결정 경계를 부정 클래스에 옮길 수 있도록 재샘플링을 해야 합니다.



다수 클래스 샘플

제거해야 하는 다수 클래스 샘플

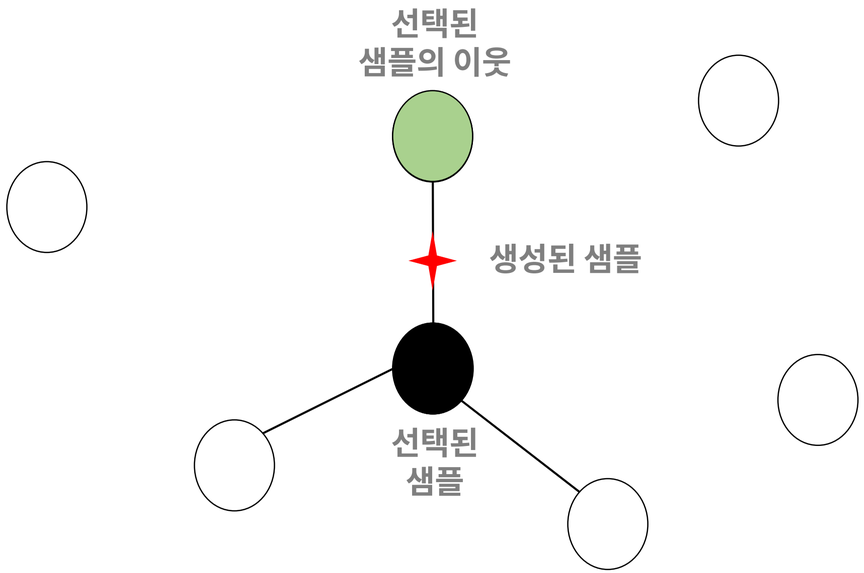
소수 클래스 샘플

생성해야 하는 소수 클래스 샘플

재샘플링 알고리즘 대부분은 결정 경계에 가까운 소수 클래스 샘플을 생성하거나 다수 클래스 샘플을 제거하는 방식으로 결정 경계를 부정 클래스 방향으로 밀도록 고안됨

해법 1. 재샘플링
(1) SMOTE

대표적인 오버샘플링 알고리즘인 SMOTE [1]는 한 소수 클래스 샘플과 그 이웃 사이에 임의로 샘플을 생성합니다.



(1) 소수 클래스 샘플 $\boldsymbol{x}$를 임의로 선택

(2) 샘플 $\boldsymbol{x}$와 가까운 k개의 소수 클래스 이웃 샘플 $\{\boldsymbol{x}_1^{nb}, \boldsymbol{x}_2^{nb}, \cdots, \boldsymbol{x}_k^{nb}\}$을 찾음

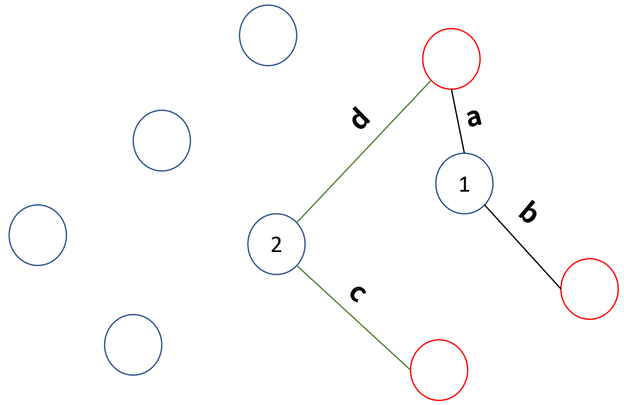
(3) k개의 이웃 샘플 이웃 가운데 임의로 하나를 선택하며, 이를 $\hat{\boldsymbol{x}}^{nb}$라 함

(4) 새로운 샘플 $\boldsymbol{x}_{new} = \boldsymbol{x} + \left(\hat{\boldsymbol{x}}^{nb} - \boldsymbol{x}\right) \times \delta$를 생성 ($\delta$는 0과 1사이의 난수)

[1] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.

해법 1. 재샘플링
(2) NearMiss

대표적인 언더샘플링 알고리즘 NearMiss는 n개의 소수 클래스 샘플까지의 평균 거리가 짧은 순서대로 다수 클래스 샘플을 제거합니다.



a + b < c + d이므로
1번 샘플을 2번 샘플보다 먼저 삭제

해법 2. 비용 민감 모델

클래스 불균형 문제는 거짓 부정 비용에 더 큰 가중치를 부여한 손실 함수를 사용함으로써 해결할 수 있습니다.

예시 1. 서포트 벡터 머신 손실 함수

일반 모델

$$\|\boldsymbol{w}\| + C\sum_i \xi_i$$

- ξ_i: 샘플 i에 대한 오분류 비용
- C: 오차 패널티

비용 민감 모델

$$\|\boldsymbol{w}\| + C\left(C_1\sum_{\{i|y_i=1\}}\xi_i + C_2\sum_{\{i|y_i=-1\}}\xi_i\right)$$

- C: 오차 페널티
- C_1: 거짓 음성 비용에 대한 오차 페널티
- C_2: 거짓 양성 비용에 대한 오차 페널티

예시 2. k-최근접 이웃의 예측 과정

일반 모델

$$\hat{y} = \begin{cases} 1, & \text{if } n_1 > n_0 \\ 0, & otherwise \end{cases}$$

- n_1: 이웃 가운데 라벨이 1인 샘플 수
- n_0: 이웃 가운데 라벨이 0인 샘플 수

비용 민감 모델

$$\hat{y} = \begin{cases} 1, & \text{if } w_1 n_1 > w_0 n_0 \\ 0, & otherwise \end{cases}$$

- w_1: n_1에 대한 가중치 (보통 클래스 불균형 비율로 설정)
- w_0: n_0에 대한 가중치 (보통 1로 설정)

예시 3. 확률 모델의 예측 과정

일반 모델

$$\hat{y} = \begin{cases} 1, & \text{if } \Pr(y|\boldsymbol{x}) > 0.5 \\ 0, & otherwise \end{cases}$$

비용 민감 모델

$$\hat{y} = \begin{cases} 1, & \text{if } \Pr(y|\boldsymbol{x}) > \theta \\ 0, & otherwise \end{cases}$$

- θ: 라벨을 1로 예측하기 위한 임계치로 비용 민감 모델에선 0.5 미만으로 설정

3. 데이터 탐색 및 전처리

6 특징 공학

특징 공학의 중요성

지도 학습 모델의 성능에 가장 큰 영향을 끼치는 요인은 단연 학습 데이터로, 특징 공간은 모델 성능의 상한을 결정합니다.

지도 학습 모델의 성능에 가장 큰 영향을 끼치는 요인은 단연 학습 데이터임

데이터를 표현하는 공간인 특징 공간이 모델 성능의 상한을 결정한다고 알려져 있음

다시 말해, 특징 공간이 고정되면 모델링 방법에 관계없이 얻을 수 있는 모델의 최대 성능은 결정됨

원 데이터의 공간을 모델링 목적에 부합하는 특징 공간으로 변환하는 일련의 작업인 특징 공학은
전체 모델링 과정에서 가장 중요한 단계라고 해도 과언이 아님

특징 공학의 분류

원 데이터의 특징을 바탕으로 새로운 특징을 생성하거나, 불필요한 특징을 제거하고 필요한 특징만 선택하거나, 기존 특징 집합을 완전히 새로운 특징 집합으로 변환하는 모든 작업이 특징 공학에 해당함

구분	설명	관련 방법
특징 생성	새로운 특징을 생성하여 원 특징 집합에 추가	• 도메인 지식을 활용 • 단항 연산자를 활용 • 다항 연산자를 활용
특징 추출	기존 특징 공간을 완전히 새로운 특징 공간으로 변환	• 주성분 분석 • 통계량 추출
특징 선택	중요도가 높지 않거나 다른 특징과 중복된 특징을 제거	• 필터링 방법 • 래퍼 방법 • 사후 방법

특징 생성

특징 생성은 새로운 특징을 생성하여 원 특징 집합에 추가하는 작업으로, 도메인 지식, 단항 및 다항 연산자를 활용하는 방법이 있습니다.

도메인 지식을 활용

(정의) 도메인 지식을 활용하는 방법은 말 그대로 도메인 지식을 활용하여 데이터를 더 잘 표현할 수 있는 새로운 특징을 만드는 것

(예시) 쇼핑몰 등에서 다음 달 고객 등급을 예측하는 문제를 해결하는 데 사용하는 데이터에 "최근 방문 날짜"라는 특징이 있음

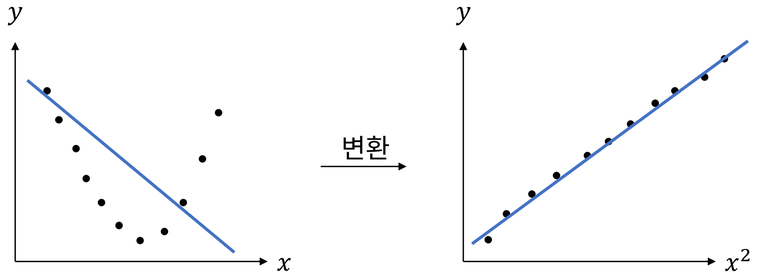
- 이 특징은 숫자가 아닌 날짜이므로 모델링에 직접 사용하기 어려우며, 날짜 자체보다 현재 날짜를 기준으로 방문한 지 얼마나 됐는지가 고객 등급을 예측하는 데 더 중요함
- 방문한지 아주 오래됐다면 얼마나 오래됐는지는 무의미할 수 있으며 하루 이틀 차이는 크게 의미가 없음
- 따라서 방문한 지 얼마나 됐는지를 구간화(예: 10일 미만, 10일 이상 30일 미만 등)하는 것이 더 적절함

단항 연산자를 활용

(정의) 단항 연산자를 사용한 특징 생성은 제곱, 루트, 로그 등 특정한 함수를 한 특징에 적용하여 새로운 특징을 만드는 것

(활용) 모델 특성에 맞춰 기존 특징을 변환하는 데 주로 사용함

(예시) 비선형 특징을 추가하여 선형 관계로 변환



특징 생성 (계속)

특징 생성은 새로운 특징을 생성하여 원 특징 집합에 추가하는 작업으로, 도메인 지식, 단항 및 다항 연산자를 활용하는 방법이 있습니다.

다항 연산자를 활용

(정의) 사용한 특징 생성은 둘 이상의 특징에 대해 함수를 적용하여 새로운 특징을 생성하는 방법

(예시) 두 특징 x_1과 x_2를 곱하여 새로운 특징 x_1x_2를 만들 수 있음

(효과 예시) 100개의 특징이 있고 이 가운데 하나라도 0이라면 긍정 클래스이고 그렇지 않다면 부정 클래스인 문제를 결정 나무를 학습하여 해결하는 상황

- 이 상황에서는 결정 나무가 과적합 될 가능성이 큼
- 결정 나무는 각 특징이 0인지를 최대 99번이나 물어봐야 정확히 분류할 수 있음
- 그렇지만 100개 특징의 곱을 하나의 새로운 특징으로 사용한다면 매우 간단하게 문제를 해결할 수 있음

머신러닝 자동화 관점에서의 특징 생성

특징 생성은 도메인 지식이 굉장히 깊이 관여하는 분야로, 완전한 자동화가 어려움

특징 생성은 도메인 지식이 굉장히 깊이 관여하는 분야임

특징과 라벨 간 관계, 특징 간 관계 등은 데이터를 통해 모두 확인하고 그에 따른 특징을 생성하는 것은 사실 불가능함

현실적으로는 도메인 지식을 바탕으로 적절한 가설을 수립하고 검정하는 방식으로 특징을 생성함. 그러나 이러한 접근은 자동화가 어려움

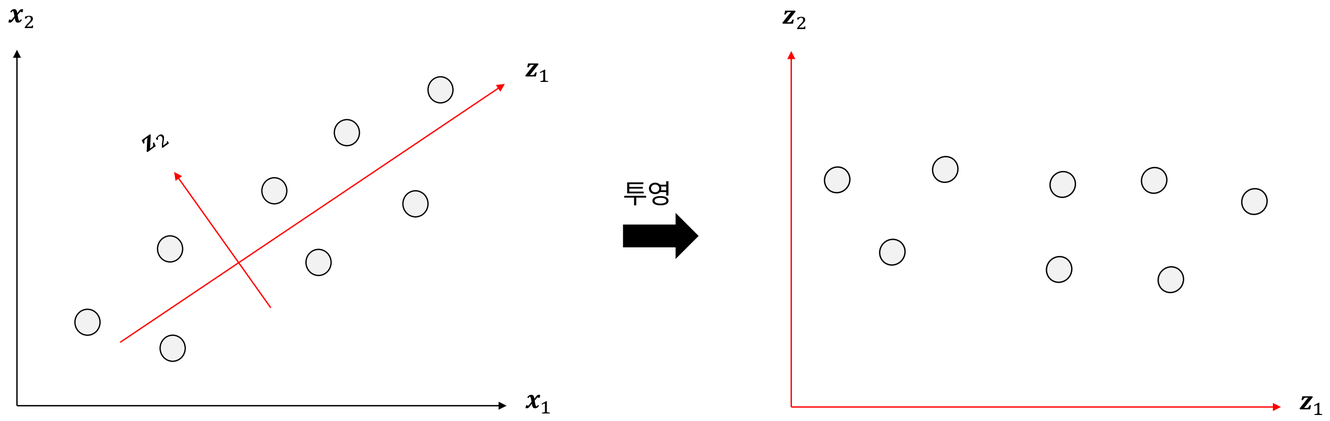
머신러닝 자동화 시스템 대부분은 특징 생성을 자동화 범위에 포함하지 않거나
미리 정의한 다양한 단항 및 다항 연산자를 모든 특징에 적용하여 후보 특징을 생성하는 데 그치고 있음

특징 추출

특징 추출은 원 특징 집합을 새로운 특징 집합으로 변환하는 방법으로, 주성분 분석 등이 여기에 속합니다.

주성분 분석

(정의) 주성분 분석은 원 데이터 공간에서 데이터의 분포를 잘 설명하는 축인 주성분을 찾아, 주성분을 축으로 하는 새로운 공간으로 데이터를 투영하는 방법

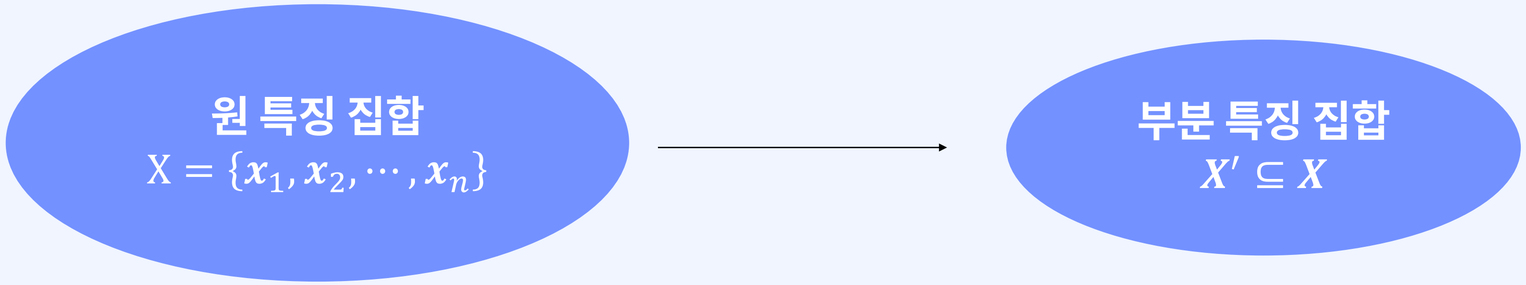


(활용) 주성분은 원 데이터를 잘 설명하는 축이므로 모든 축을 사용하지 않더라도 원 데이터를 충분히 설명할 수 있으므로, 특징의 개수를 줄이는 데(차원 축소)도 사용할 수 있음

특징 추출은 전체 특징 공간을 새로운 공간으로 변환시키므로 선택지가 적어 머신러닝 자동화에서 사용하기 부적절합니다.

특징 선택

특징 선택은 원 특징 집합에서 특징 간 중복이 적고 예측 성능을 최대로 하는 최적의 부분 집합을 구성하는데 필요한 특징을 선택하는 것입니다.



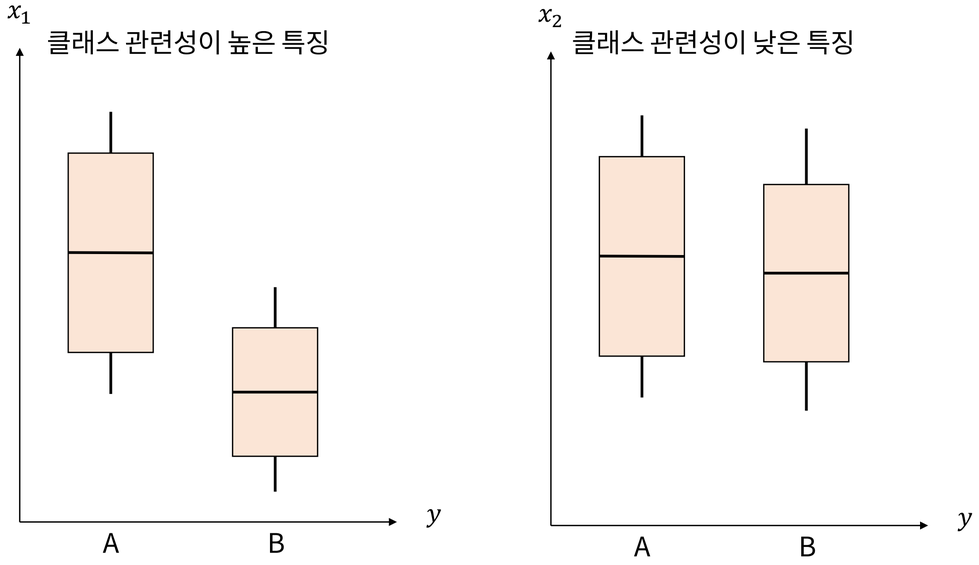
특징 선택 방법의 구분

방법	설명
필터링	통계 검정 기법을 바탕으로 각각의 특징과 라벨 간 관계를 점수로 나타내어, 점수가 높은 특징을 선택하는 방법
래퍼	모델의 성능을 최대화하는 특징 집합을 찾는 방법
사후	학습한 모델에서 각 특징의 중요도를 측정하고, 이 중요도를 바탕으로 특징을 선택하는 방법

특징 선택
(1) 필터링

필터링 방법(filtering method)은 클래스 관련성(class relevance)이 높은 특징을 선택하는 방법입니다.

클래스 관련성: 한 특징이 클래스를 얼마나 잘 설명하는지를 나타내는 척도로 주로 통계적 검정 기법을 사용해 측정함



- 라벨 y는 범주형 변수로 A 혹은 B를 가지며, x_1과 x_2는 모두 연속형 특징임
- 특징 x_1은 y가 A일 때 값이 크고 B일 때 값이 작음
- 특징 x_2는 y가 A든 B든 값 차이가 크지 않음
- 따라서 x_1은 y를 예측하는 데 도움이 되지만 x_2는 그렇지 않음

특징 선택
(1) 필터링 (계속)

클래스 관련성을 측정하는 척도는 주로 통계적 검정에 사용하는 척도이며, 라벨과 특징의 유형에 따라 사용할 수 있는 척도가 결정됩니다.

특징	라벨	척도
범주형	범주형 (분류)	카이제곱 통계량, 상호 정보량
범주형	연속형 (회귀)	상호 정보량, F – 통계량
연속형	범주형 (분류)	상호 정보량, F – 통계량
연속형	연속형 (회귀)	R^2, 상호 정보량

특징과 라벨의 유형에 따라 척도를 선택하면 되며, 모든 척도가 값이 클수록 클래스 관련성이 높다고 할 수 있음

특징 선택
(2) 래퍼 방법

래퍼 방법(wrapper method)은 모델의 예측 정확도 측면에서 가장 좋은 성능을 보이는 특징 집합을 구성하는 방법입니다.

실제 모델의 성능을 최대화할 수 있도록 특징 집합을 구성한다는 장점이 있지만,
사용하고자 하는 모델에 종속적이며 하나의 특징 집합을 평가하려면 시간이 오래 걸린다는 단점이 있음

특징 선택
(3) 사후 방법

사후 방법(post-hoc method)은 학습한 모델에서 측정한 특징의 중요도(feature importance)를 바탕으로 특징을 선택하는 방법입니다.

방법	설명
모델에서 측정	결정 나무와 결정 나무 기반의 앙상블 모델은 각 특징이 모델에 몇 번이나 등장했는지 혹은 불순도를 얼마나 낮췄는지를 기준으로 특징 중요도를 계산할 수 있음
모델 학습 후 측정	순열 중요도(permutation importance)는 정상적인 특징 벡터를 입력했을 때의 예측 성능과 각 특징의 순서를 임의로 섞은 특징 벡터를 입력했을 때의 예측 성능의 차이를 바탕으로 특징의 중요도를 계산할 수 있음

머신러닝 자동화에서 특징 선택

머신러닝 자동화 관점에서는 필터링 방법이 가장 적합한 특징 선택 방법이라고 할 수 있습니다.

래퍼 방법과 사후 방법은 특징을 평가하려면 모델을 학습해야 하므로 시간이 오래 걸리고 모델에 의존적임

머신러닝 자동화에서는 여러 모델과 하이퍼 파라미터를 비교하므로, 여기에 특징 선택까지 포함한다면 탐색 범위가 비현실적으로 넓어짐

필터링 방법은 시간이 오래 걸리지도 않고 모델에 독립적이지 않으며, 하이퍼 파라미터가 있긴 하나 충분히 대응할만한 수준임

머신러닝 자동화에서는 단항 및 다항 연산 기반의 특징 생성과 필터링 기반의 특징 선택을 주로 수행합니다.