Chapter. 20

편향된 모델은 쓸모없어: 클래스 불균형 문제

문제 정의 및 탐색 방법

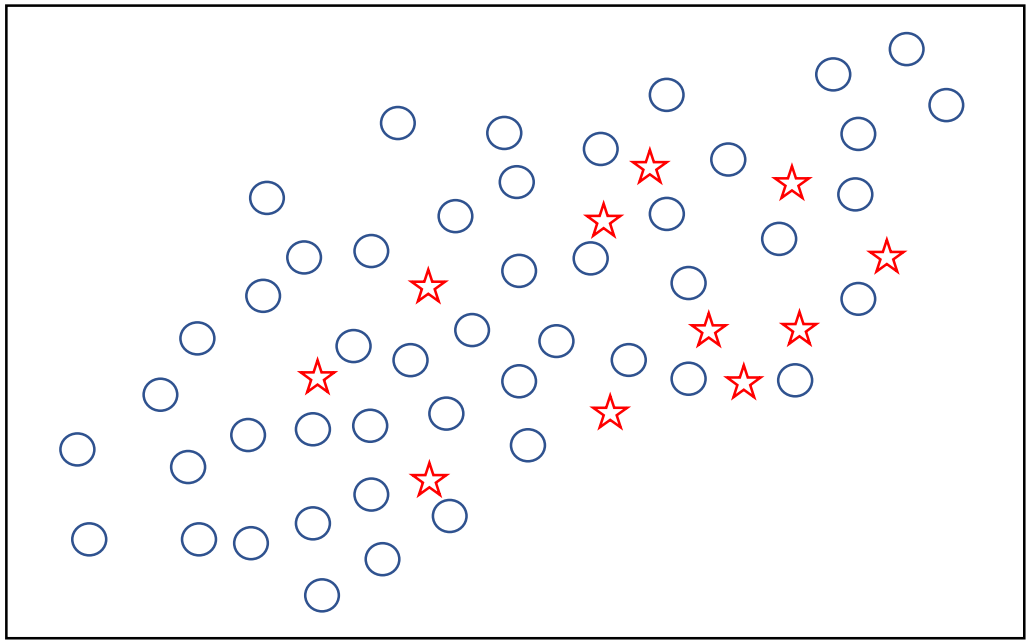
FAST CAMPUS ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 문제 정의

- 클래스 변수가 하나의 값에 치우친 데이터로 학습한 분류 모델이 치우친 클래스에 대해 편향되는 문제로, 이러한 모델은 대부분 샘플을 치우친 클래스 값으로만 분류하게 됨 (예시: 암환자 판별 문제)

- 클래스 불균형 문제가 있는 모델은 정확도와 높고, 재현율이 매우 낮은 경향이 있음



		실제	
		○	☆
예측	○	9999	1
	☆	0	0

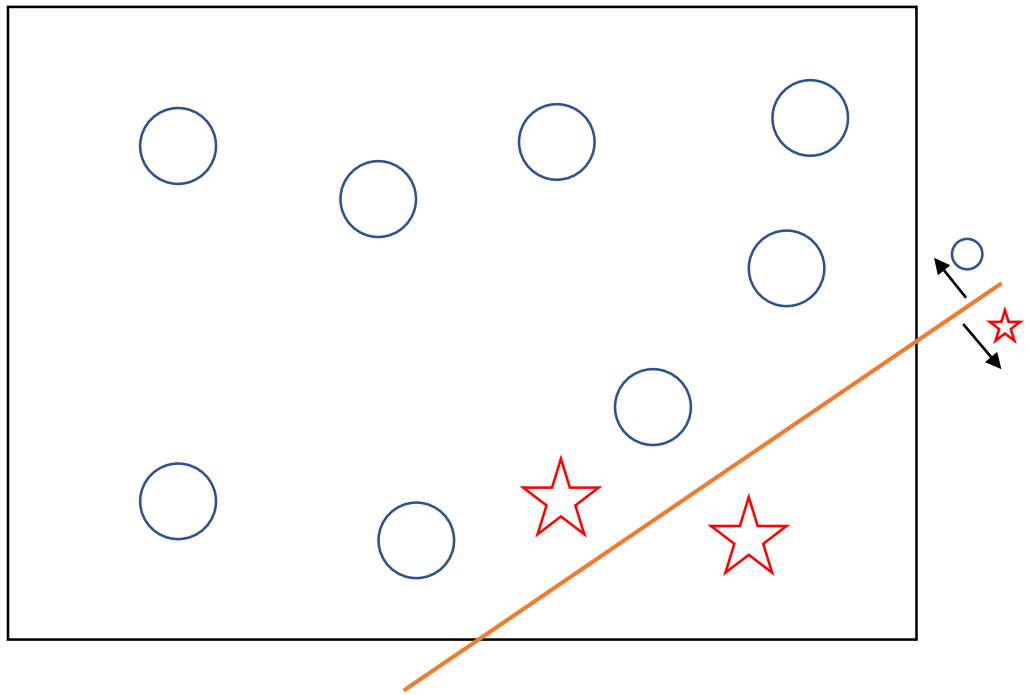
정확도: 99.99% 재현율: 0.00%

I 용어 정의

- 다수 클래스: 대부분 샘플이 속한 클래스 (예: 정상인)
- 소수 클래스: 대부분 샘플이 속하지 않은 클래스 (예: 암환자)
- 위양성 비용 (False positive; TP): 부정 클래스 샘플을 긍정 클래스 샘플로 분류해서 발생하는 비용
- 위음성 비용 (False negative; TN): 긍정 클래스 샘플을 부정 클래스 샘플로 분류해서 발생하는 비용
- 보통은 위음성 비용이 위양성 비용보다 훨씬 큼 (예: 정상인 → 암환자 vs 암환자 → 정상인)
- 절대 부족: 소수 클래스에 속한 샘플 개수가 절대적으로 부족한 상황

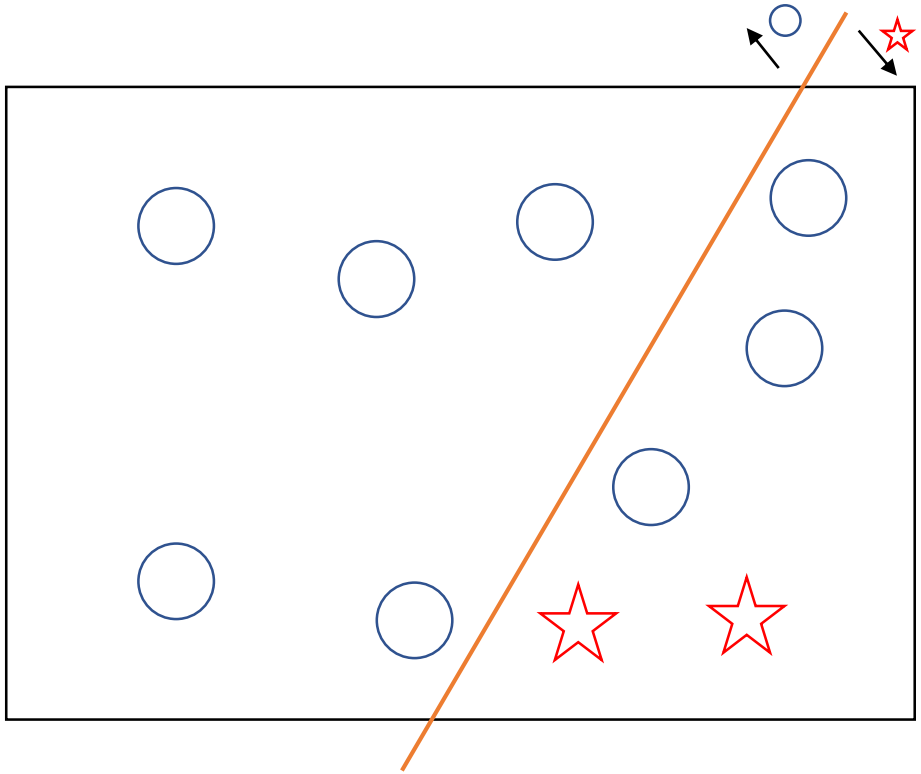
I 발생 원인

- 대부분의 분류 모형의 학습 목적식은 정확도를 최대화하는 것이므로, 대부분 샘플을 다수 클래스라고 분류하도록 학습됨



✓ 정확도: 9 / 10
✓ 재현율: 1 / 2

>



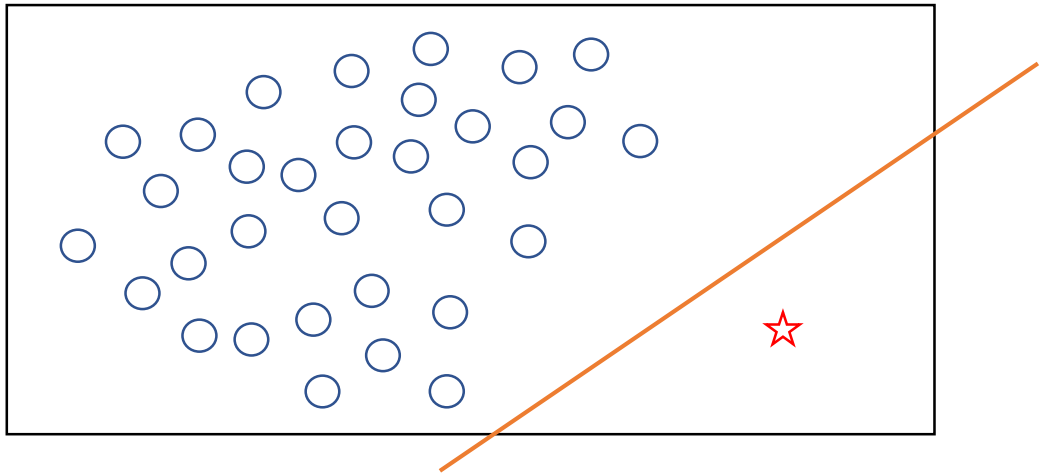
✓ 정확도: 7 / 10
✓ 재현율: 2 / 2

I 탐색 방법 (1) 클래스 불균형 비율

- 클래스 불균형 비율이 9 이상이면 편향된 모델이 학습될 가능성이 있음

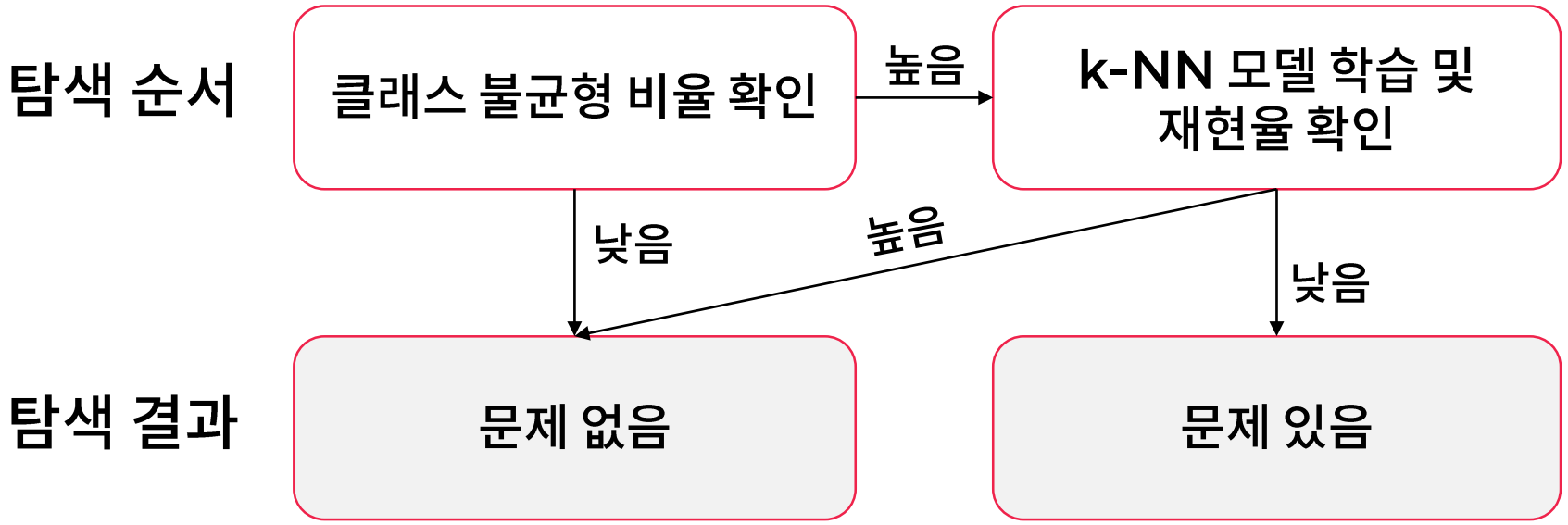
$$\text{클래스 불균형 비율} = \frac{\text{다수 클래스에 속한 샘플 수}}{\text{소수 클래스에 속한 샘플 수}}$$

- 다만, 클래스 불균형 비율이 높다고 해서 반드시 편향된 모델을 학습하는 것은 아님



I 탐색 방법 (2) k-최근접 이웃을 활용하는 방법

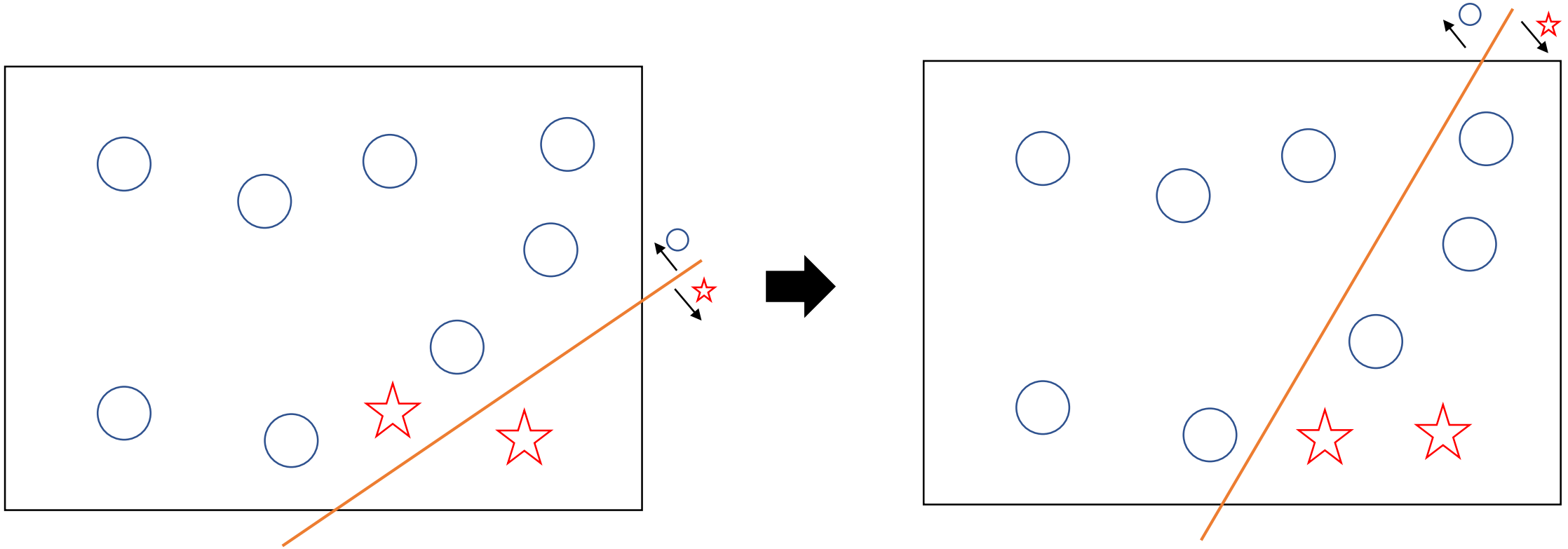
- k – 최근접 이웃은 이웃의 클래스 정보를 바탕으로 분류를 하기에 클래스 불균형에 매우 민감하므로, 클래스 불균형 문제를 진단하는데 적절함



- k값이 크면 클수록 더욱 민감하므로, 보통 5 ~ 11 정도의 k를 설정하여 문제를 진단함

I 문제 해결의 기본 아이디어

- 클래스 불균형 문제 해결의 기본 아이디어는 소수 클래스에 대한 결정 공간을 넓히는 것임



Chapter. 20

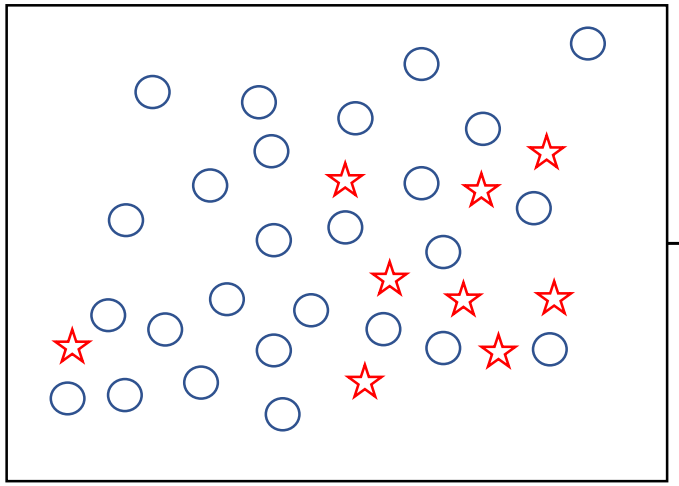
편향된 모델은 쓸모없어: 클래스 불균형 문제

재샘플링

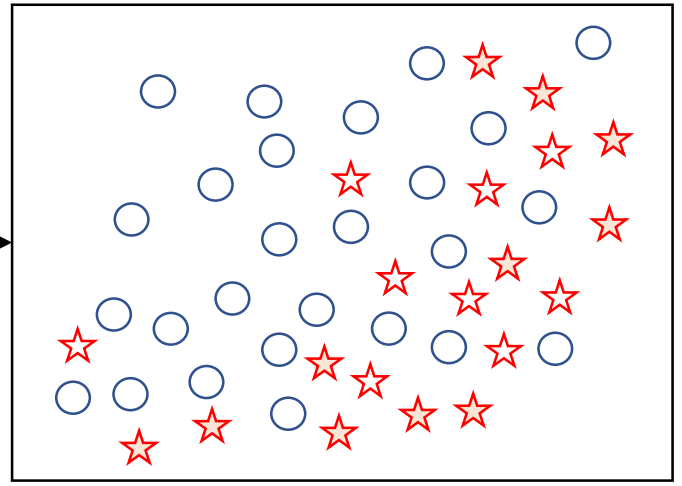
FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 분류: 오버샘플링과 언더샘플링

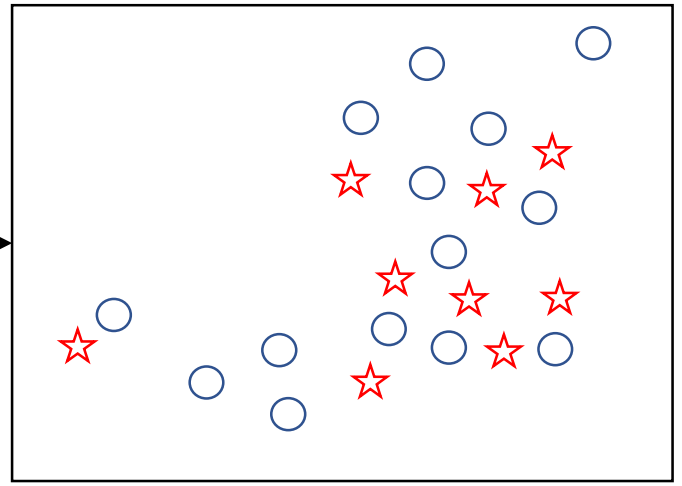


오버샘플링
소수 클래스 샘플 생성



원본 데이터가
작을 때 유용

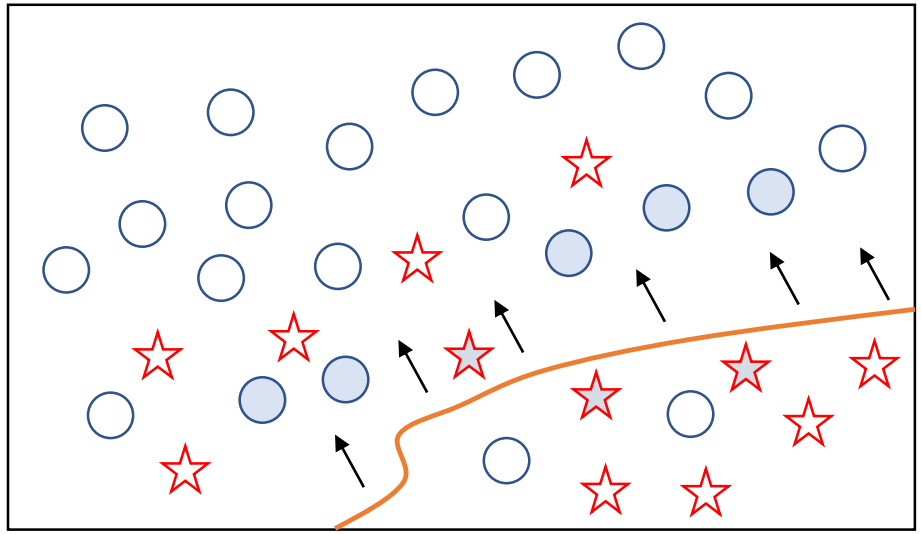
언더샘플링
다수 클래스 샘플 삭제



원본 데이터가
클 때 유용

I 어디에 만들고 어느 것을 지울까?

- 결정 경계에 가까운 다수 클래스 샘플을 제거하고, 결정 경계에 가까운 소수 클래스 샘플을 생성해야 함



다수 클래스 샘플

제거해야 하는 다수 클래스 샘플

소수 클래스 샘플

생성해야 하는 소수 클래스 샘플

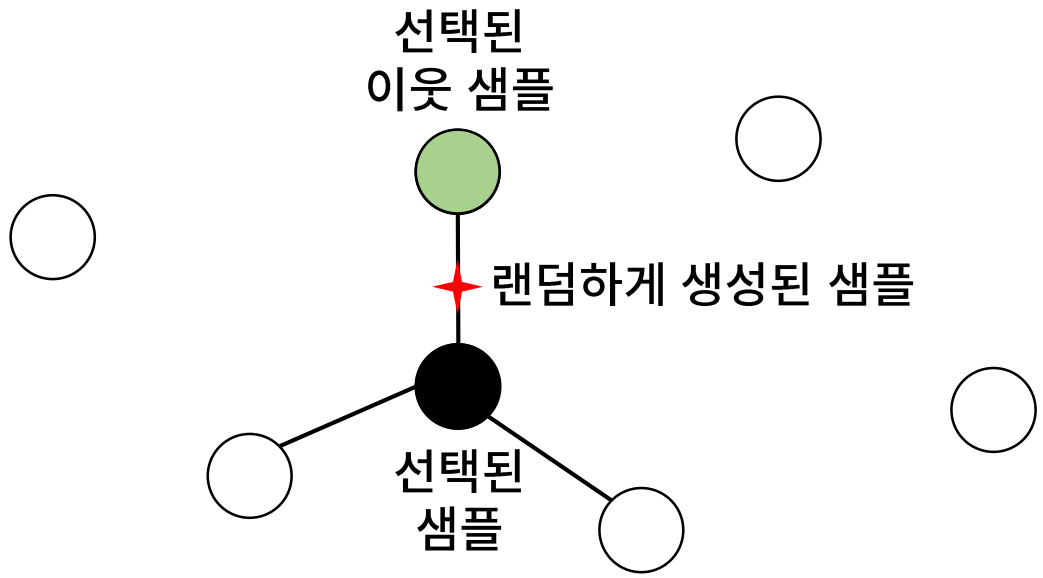
- 주의: 평가 데이터에 대해서는 절대로 재샘플링을 적용하면 안 됨

I 대표적인 오버샘플링 알고리즘: SMOTE

- SMOTE (Synthetic Minority Over-Sampling Technique)는 2002년에 제안된 기법으로, 대부분의 오버 샘플링 기법이 이 기법에 기반하고 있음

- 소수 클래스 샘플을 임의로 선택하고, 선택된 샘플의 이웃 가운데 하나의 샘플을 또 임의로 선택하여 그 중간에 샘플을 생성하는 과정을 반복하는 방법

이웃 수(k): 3



작동과정 상세

(1) 소수 클래스 샘플 x를 임의로 선택

(2) 샘플 x와 가까운 k개의 소수 클래스 이웃 샘플 $\{x_1^{nb}, x_2^{nb}, \cdots, x_k^{nb}\}$을 찾음

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

(4) 새로운 샘플 $x_{new} = x + (\hat{x}^{nb} - x) \times \delta$를 생성 (여기서 δ는 0과 1사이의 난수)

imblearn.over_sampling.SMOTE

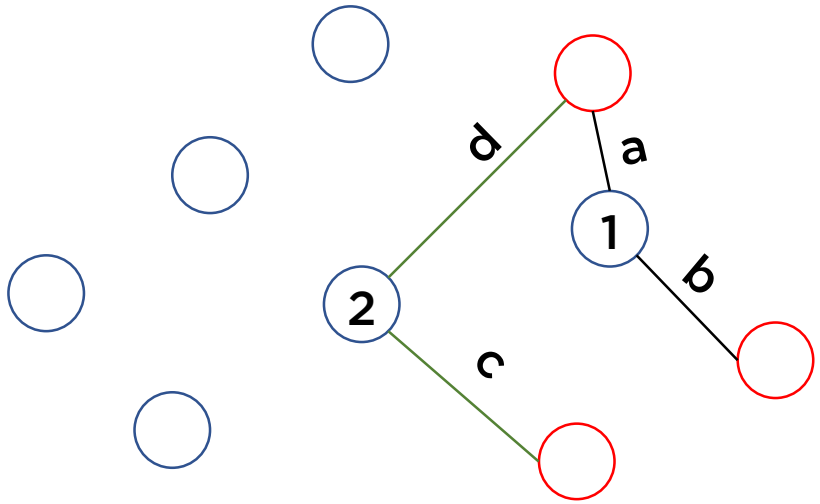
- 주요 입력
  - sampling_strategy: 입력하지 않으면 1:1 비율이 맞을 때까지 샘플을 생성하며, 사전 형태로 입력하여 클래스별로 생성하는 샘플 개수를 조절 가능
  - k_neighbors: SMOTE에서 고려하는 이웃 수 (보통 1, 3, 5 정도로 작게 설정)

- 주요 메서드
  - .fit_sample(X, Y): X와 Y에 대해 SMOTE를 적용한 결과를 ndarray 형태로 반환

I 대표적인 언더샘플링 알고리즘: NearMiss

- 가장 가까운 n개의 소수 클래스 샘플까지 평균 거리가 짧은 다수 클래스 샘플을 순서대로 제거하는 방법

$n = 2$



a + b < c + d이므로
1번 샘플을 2번 샘플보다 먼저 삭제

imblearn.under_sampling.NearMiss

- 주요 입력
  - sampling_strategy: 입력하지 않으면 1:1 비율이 맞을 때까지 샘플을 생성하며, 사전 형태로 입력하여 클래스별로 생성하는 샘플 개수를 조절 가능
  - n_neighbors: 평균 거리를 구하는 소수 클래스 샘플 수
  - version: NearMiss의 version으로, 2를 설정하면 모든 소수 클래스 샘플까지의 평균 거리를 사용

- 주요 메서드
  - .fit_sample(X, Y): X와 Y에 대해 NearMiss를 적용한 결과를 ndarray 형태로 반환

Chapter. 20

편향된 모델은 쓸모없어: 클래스 불균형 문제

비용 민감 모델

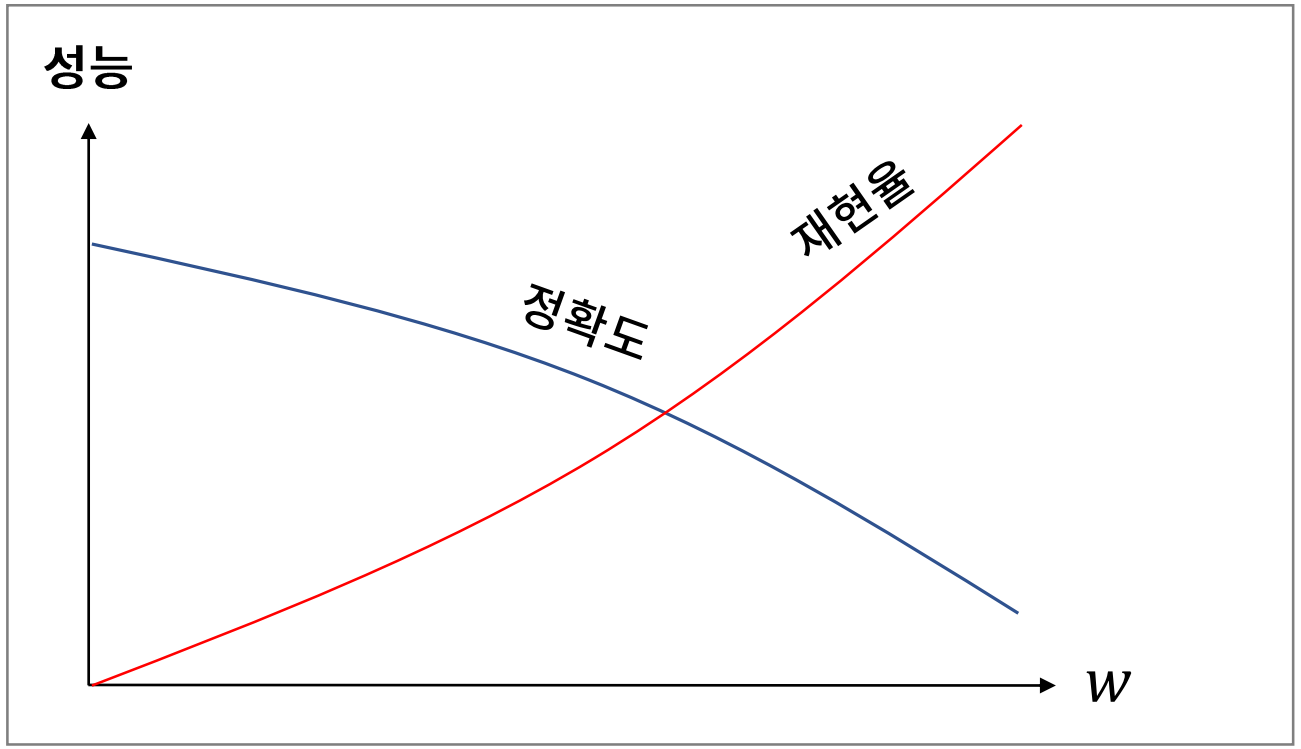
FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 정의

- 학습 목적식에서 위음성 비용 (긍정 클래스를 부정 클래스로 오분류할 때 발생하는 비용)과 위양성 비용 (부정 클래스를 긍정 클래스로 오분류할 때 발생하는 비용)를 다르게 설정하는 모델로, 보통 **위음성 비용을 위양성 비용보다 크게 설정**

- 즉, 위음성 비용 = $w \times$ 위양성 비용 $(w > 1)$로 설정한 모델을 비용 민감 모델이라 함



| 확률 모델

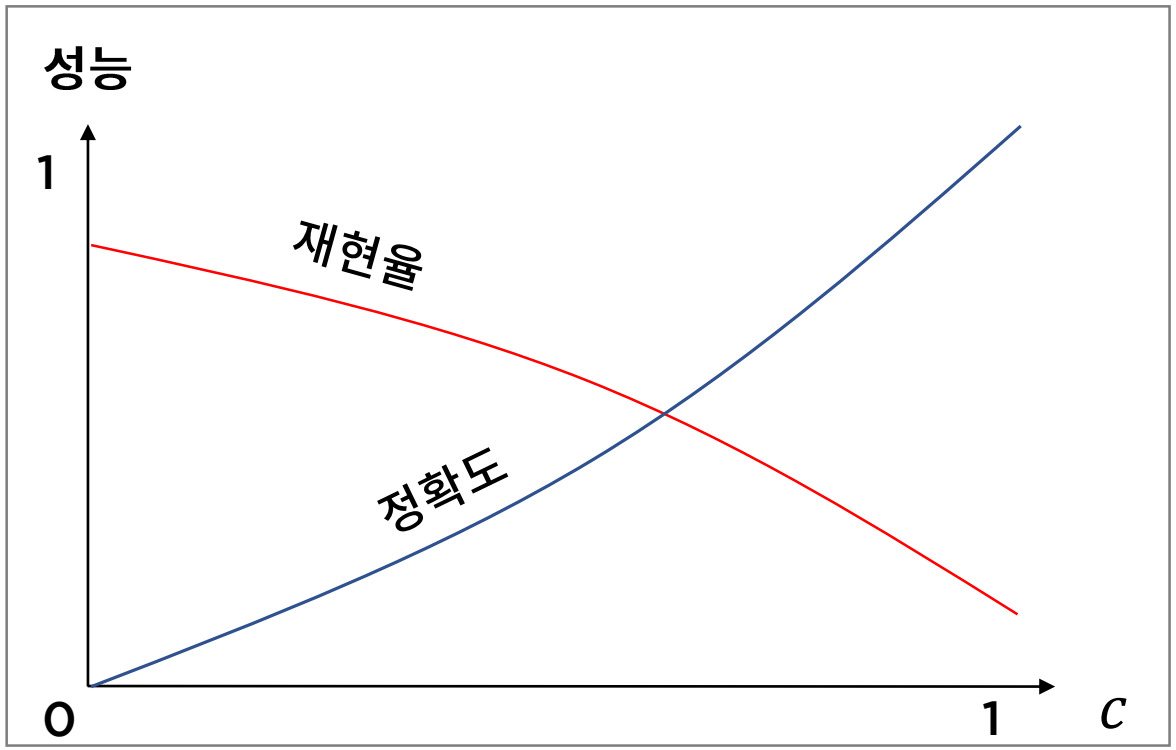
- 로지스틱 회귀, 나이브 베이즈 등의 확률 모델들은 **cut – off value**, c를 조정하는 방식으로 비용 민감 모델을 구현할 수 있음

- 정확한 확률 추정은 불가능하지만 그 개념을 도입할 수 있는 모델(k-최근접 이웃, 신경망, 의사결정나무, 앙상블 모델 등)에도 역시 적용이 가능함

$$\Pr(y|\boldsymbol{x}) \geq c \Rightarrow \text{“Pos”}$$

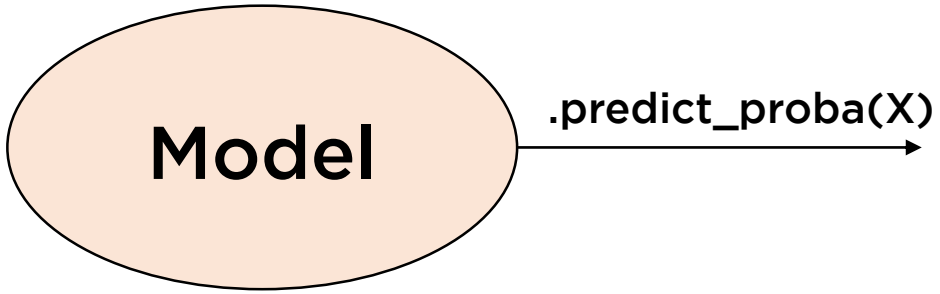
$$\Pr(y|\boldsymbol{x}) < c \Rightarrow \text{“Neg”}$$

확률 모델



I 관련 문법: .predict_proba

- sklearn의 확률 모델이 갖는 메서드(fit 이후)로서, X를 입력으로 받아 각 클래스에 속할 확률을 출력



Index	Pr(Neg|x)	Pr(Pos|x)
1	Pr(Neg|x_1)	Pr(Pos|x_1)
2	Pr(Neg|x_2)	Pr(Pos|x_2)
3	Pr(Neg|x_3)	Pr(Pos|x_3)
4	Pr(Neg|x_4)	Pr(Pos|x_4)

Model.classes_ = [Neg, Pos]

Tip. Numpy와 Pandas 잘 쓰는 기본 원칙: 가능하면 배열 단위 연산을 하라

- 유니버설 함수, 브로드캐스팅, 마스크 연산을 최대한 활용하자
- (예시) 확률 모델의 cut_off_value를 0.3으로 설정하기

Index	Pr(Neg|x)	Pr(Pos|x)
1	0.80	0.20
2	0.65	0.35
3	0.60	0.40
4	0.55	0.45

X

X.iloc[:, -1] >= 0.3

Index	Value
1	False
2	True
3	True
4	True

R

2 * R - 1

Index	Value
1	-1
2	1
3	1
4	1

I 비확률 모델 (1) 서포트 벡터 머신

Minimize	$\|\boldsymbol{w}\| + C\sum_i \xi_i$
Subject to	$y_i(w\boldsymbol{x}_i - b) \geq 1 - \xi_i$, **for all** i

일반 서포트 벡터 머신

- ξ_i: 샘플 i에 대한 오분류 비용
- C: 오차 패널티

⇒ 거짓 양성 비용과 거짓 음성 비용을 구별하지 않음

Minimize	$\|\boldsymbol{w}\| + C\left(C_1\sum_{\{i\|y_i=1\}}\xi_i + C_2\sum_{\{i\|y_i=-1\}}\xi_i\right)$
Subject to	$y_i(w\boldsymbol{x}_i - b) \geq 1 - \xi_i$, **for all** i

비용 민감 서포트 벡터 머신

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

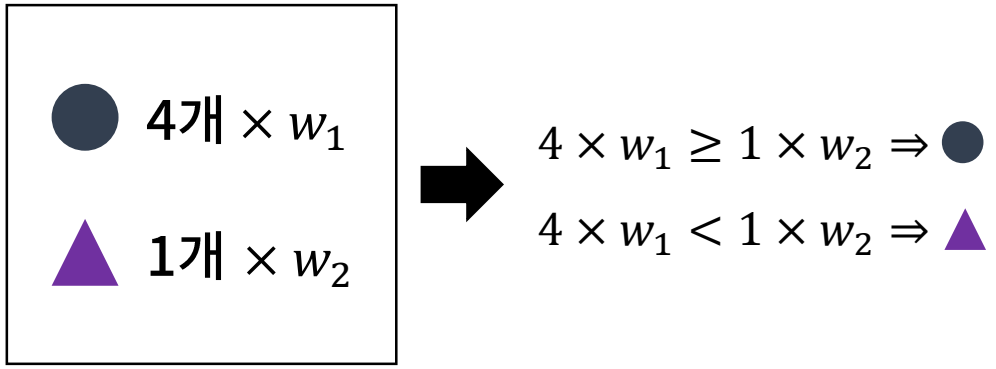
⇒ 거짓 양성 비용과 거짓 음성 비용을 구별함
⇒ 보통은 $C_1 > C_2$로 설정함

I 비확률 모델 (2) 의사결정나무

- 소수 클래스에 대한 가중치를 부여하는 방식으로 가능하면 소수 클래스로 분류하도록 유도

일반 모델

비용 민감 모델



I 관련 문법: class_weight

- DecisionTreeClassifier, SVC, RandomForestClassifier 등에서는 class_weight라는 파라미터가 있으며, 사전 형태로 입력함
  - ➢ key: class 이름 (예: C1)
  - ➢ value: class weight (예: 10)

- (예시) SVC(class_weight = {1: 10, -1: 1}) ⇒ 클래스 1에 클래스 -1보다 10배의 가중치를 부여

Chapter.

편향된 모델은 쓸모없어: 클래스 불균형 문제

감사합니다

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승