불균형 데이터

- 정상 범주 관측치 수와 이상 범주 관측치 수 차이가 크게 나는 Data.
- 중요한 이유: 일반적으로 이상(소수)를 분류하는게 더 중요하기 때문.
  → 암, 사기거래, 불량
  → 적절한 분류경계선 형성 X → 이상 찾아낼 수 X
- Confusion matrix

ex)

		예측	
		이상	정상
실제	이상	5	5
	정상	0	40

→ $Accuracy = \frac{5+40}{5+5+0+40} = 0.9$

↓

모델 성능 왜곡

불균형 데이터 해결방안

- 데이터 조정해서 해결 → Sampling method (샘플링 기법)
- 모델 조정해서 해결 → Cost sensitive learning (비용기반학습)
  → Novelty detection (단일 클래스 분류기법)

샘플링 기법
- 언더 샘플링
  - Random undersampling
  - Tomek links
  - Condensed Nearest Neighbor Rule
  - One-sided Selection
- 오버 샘플링
  - Resampling
  - SMOTE
  - Borderline-SMOTE
  - ADASYN

언더샘플링

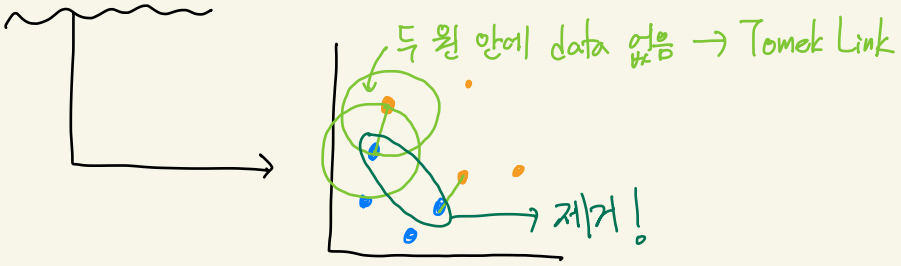
1) Random Undersampling

: 다수 범주에서 무작위로 샘플링

→ 무작위로 sampling → 항상 다른 결과

2) Tomek links

: Tomek links 형성 후, 다수 범주에 속한 관측치 제거.



3) Condensed nearest neighbor (CNN)

: 소수범주 전체, 다수범주 중 data 1개 무작위로 선택 (서브데이터 만들기)

→ 1-NN으로 원래 데이터 분류.

→ 다수범주로 분류된거 제거

* kNN ($k \neq 1$)로 하면?

→ 무조건 소수범주로 분류됨. CNN은 무조건 1-NN으로!

4) One-side selection (OSS)

: Tomek links + CNN

- Tomek links → Borderline 쪽 지움.
- CNN → 안전한 정상만 지움

Undersampling

- 장점:
  - 다수 범주 관측치 제거 → 계산시간 감소
  - 데이터 클렌징 → 클래스 오버랩 감소
- 단점: 데이터 제거 → 정보 손실

오버샘플링

1) Resampling

→ 소수범주 관측치 늘리기

ex) 다수범주 300개 / 소수범주 10개 → 다수범주 300개 / 소수범주 300개

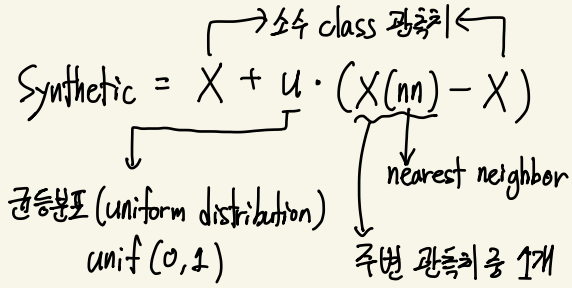
소수범주 copy

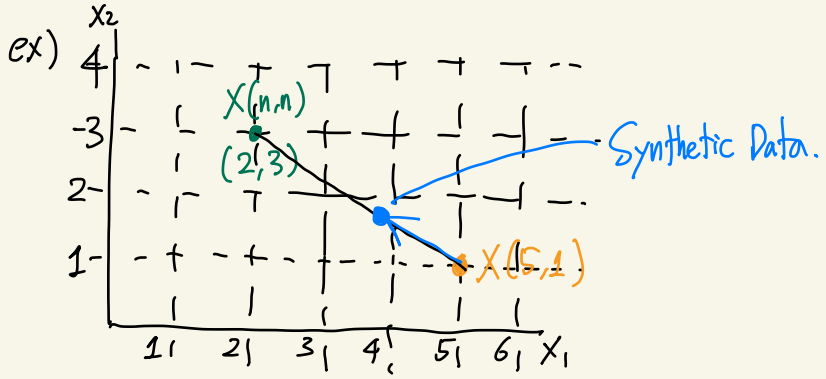
단점: 소수 클래스에 과적합 발생가능. → 보완책: 가상 관측치 생성

2) SMOTE (Synthetic Minority Oversampling Technique)

: 소수범주에서 가상 데이터 생성하는 방법.

→ 소수범주에서 임의로 하나 선택, kNN처럼 가까운 k개 지정, 그 중 랜덤으로 하나 선택





→ 모든 소수범주 data에 대해 다 함. → 소수범주data 많아짐.

* k=1로 하면 늘어난 모양으로 생겨버림. k=1로 X.

3) Borderline - SMOTE

: Borderline 부분만 oversampling 해보자

<1> Borderline 찾기

소수 class x_i에 대해 kNN 해봄 → k개중 다수 class 개수 확인

$\begin{pmatrix} \text{Safe} \\ \text{Danger} \\ \text{Noise} \end{pmatrix}$로 분류 $\begin{pmatrix} 0 \le k' \le k/2 \rightarrow \text{safe} \\ k/2 < k' < k \rightarrow \text{Danger} \\ k = k' \longrightarrow \text{Noise} \end{pmatrix}$

<2> Danger 에만 SMOTE 적용. → Borderline 쪽에만 Oversampling 됨.

4) ADASYN (adaptive synthetic sampling approach)

: Over Sampling 개수를 위치에 따라 다르게

$$r_i = \frac{\Delta_i}{K} \quad (i = 1, \cdots, m)$$

Δ_i : 소수 class x_i의 주변 k개 중 다수 class 관측치 개수

→ 각 소수 class x_i마다 주변에 다수 class 얼마나 있나?

모든 소수 class 관측치에 대해 r_i 구하기.

→ scaling : $\hat{r}_i = \frac{r_i}{\sum_{i=1}^{m} r_i}$

G : 다수클래스개수 − 소수클래스 개수

→ $\hat{r}_i \times G$ 한 후 반올림한 값 만큼 oversampling.

→ 소수 class 주변 다수 class 수에 따라 유동적으로 oversampling 가능!

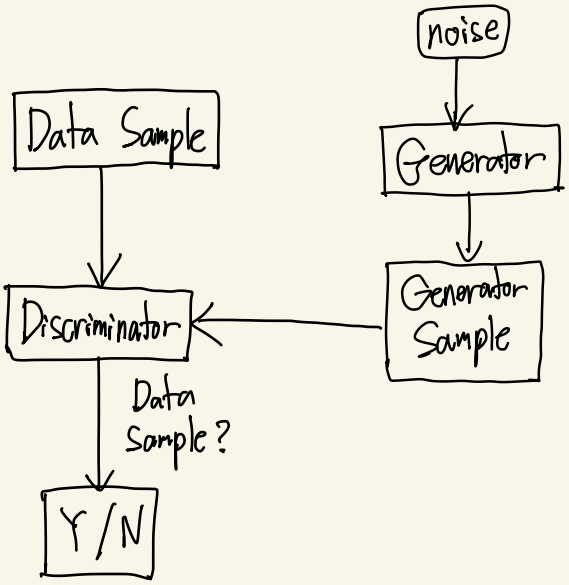
5) GAN (Generative Adversarial Nets)

<1> 무작위로 noise 생성 (0,1)

<2> Generator로 가짜 Sample 생성

<3> Discriminator로 진짜/가짜 판별

<4> 반복적으로 Generator 업데이트 → 진짜 Sample과 유사한 Data 생성



Oversampling
- 장점
  - 정보손실X
  - 대부분의 경우 undersampling보다 정확도↑
- 단점
  - 과적합 가능성
  - 계산시간 증가
  - noise 또는 이상치에 민감.

비용기반학습

: 오분류 비용 다름 → modeling에 오분류 비용 고려

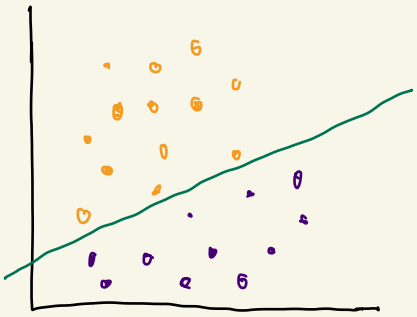
- 비용 기반 데이터 가중치 부여
  - 의사 결정 트리에 적용 → Cost-sensitive Decision Tree
  - 인공 신경망에 적용 → Cost-sensitive Neural Network
  - → 기존 방법들에 결합.

단일 클래스 분류 기법

: 다수 범주만 고려해서 분류

- 가능한 모든 다수범주 포함 분류경계선 설정 ex) 원

ex) Two-class classification problem



One-class classification problem