앙상블_샘플링&스태킹

- 참고문헌: 파이썬 머신러닝 완벽가이드, 권철민, 위키북스, 2022

https://www.youtube.com/watch?v=Vhwz228VrIk

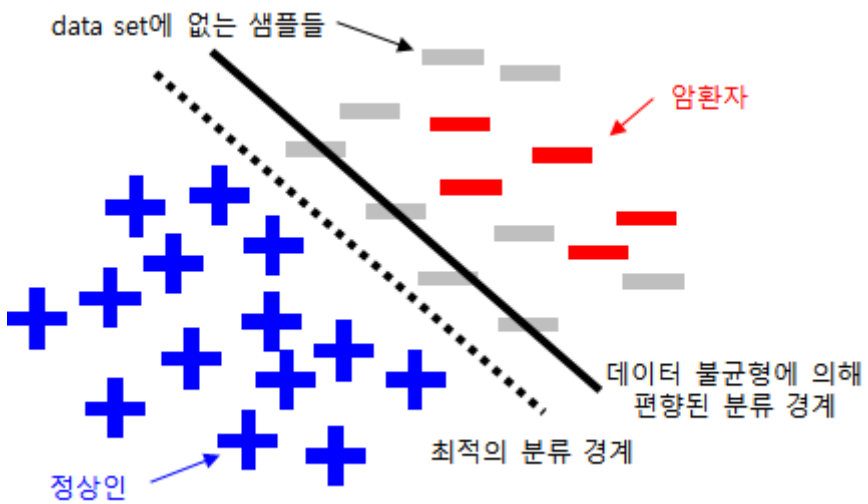
– 권수태 교수

JEONJU UNIVERSITY
The Place for Superstars

1. 언더/오버 샘플링

❖ 불균형데이터

- ➢ 정상 범주의 관측치 수와 이상 범주의 관측치 수가 현저히 차이나는 데이터
  - ✓ 암 발생 환자가 암에 걸리지 않은 사람보다 현저히 적고,
  - ✓ 신용카드 사기 거래인 경우가 정상 거래인 경우보다 현저히 적음
- ➢ 레이블이 불균형한 분포를 가진 데이터 세트를 학습시킬때 예측성능의 문제가 발생

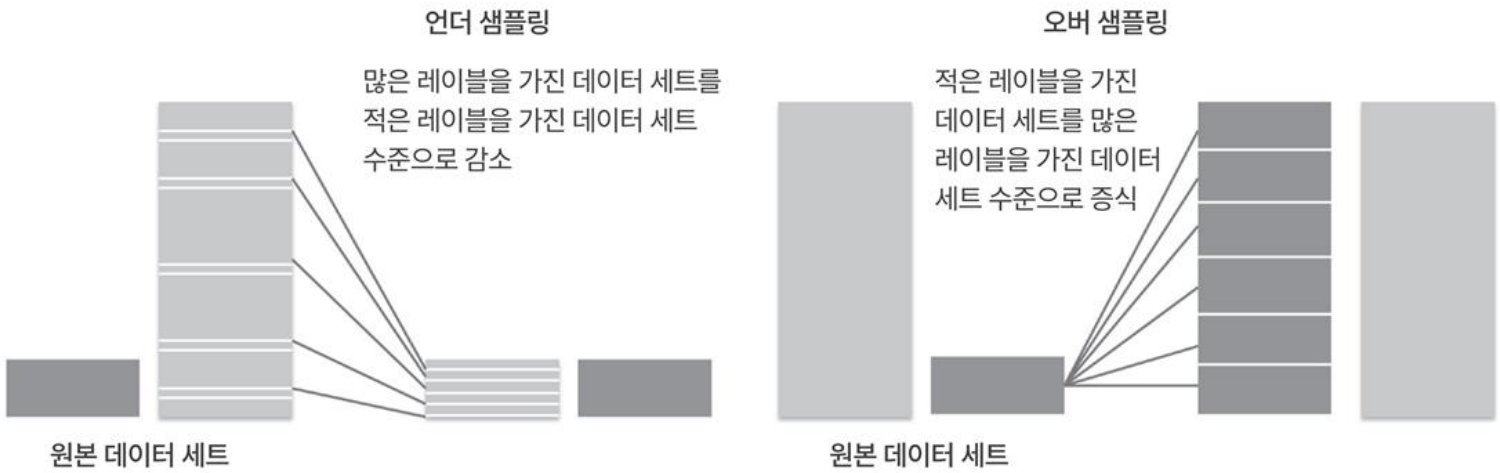


JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖ 불균형데이터

- ➢ 데이터 불균형이 심한 경우 모델의 일반화 성능을 이끌어내기 어렵기 때문에 데이터 불균형 문제를 먼저 해결하고 모델을 학습시켜야 함
- ➢ 문제해결을 위해 오버샘플링과 언더샘플링으로 적절한 학습데이터를 확보



JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 언더 샘플링

✓ 다수 범주의 데이터를 줄여서 소수 범주의 샘플 수와 비슷하게 함

✓ 다수 범주 관측치 제거로 계산 시간 감소

✓ 데이터 클랜징으로 클래스 오버랩 감소

✓ 데이터 제거로 인한 정보 손실 발생 (중요 정보가 삭제될 시 치명적)

➢ 오버 샘플링

✓ 소수 범주의 데이터를 증폭시켜서 다수 범주의 샘플 수와 비슷하게 함

✓ 정보 손실이 없음

✓ 보통 언더 샘플링보다 분류 정확도가 높음

✓ 과적합 가능성이 있고, 계산 시간이 오래 걸리며, 노이즈나 이상치에 민감

1. 언더/오버 샘플링

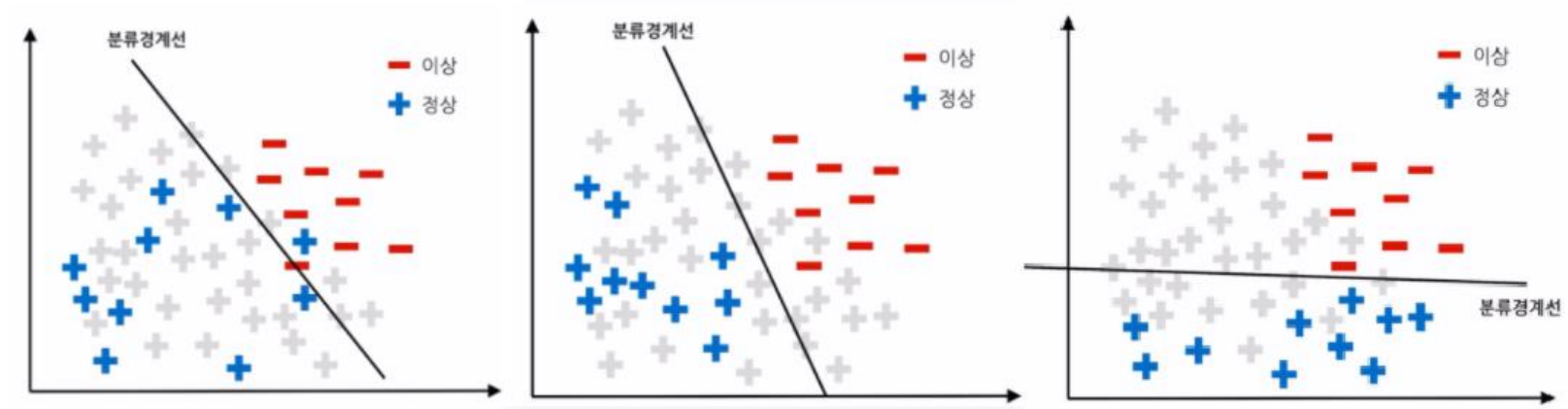
❖언더 샘플링과 오버 샘플링의 이해

➢ 언더 샘플링(Random Undersampling)

✓다수 범주의 샘플들을 무작위로 선택

✓샘플 선택에 아무런 제약이 없기 때문에 처리 시간이 빠름

✓무작위 선택으로 인해 샘플링 결과에 따라 분류 경계면의 변동 심함



JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 언더 샘플링(Tomek Links)

✓ 두 범주 사이를 탐지하고 정리를 통해 부정확한 분류경계선을 방지하는 방법

✓ 두 범주에서 하나씩 추출한 포인트를 각각 x_i 와 x_j 라 할때,

$d(x_i, x_k) < d(x_i, x_j)$ 또는 $d(x_j, x_k) < d(x_i, x_j)$ 가 되는 관측치 x_k 가 없는 경우, 두 샘플 x_i 와 x_j 가 Tomek Link를 형성한다고 함

✓ Tomek link를 형성하는 두 샘플 중 하나는 노이즈이거나 둘 다 경계선 근처에 있음

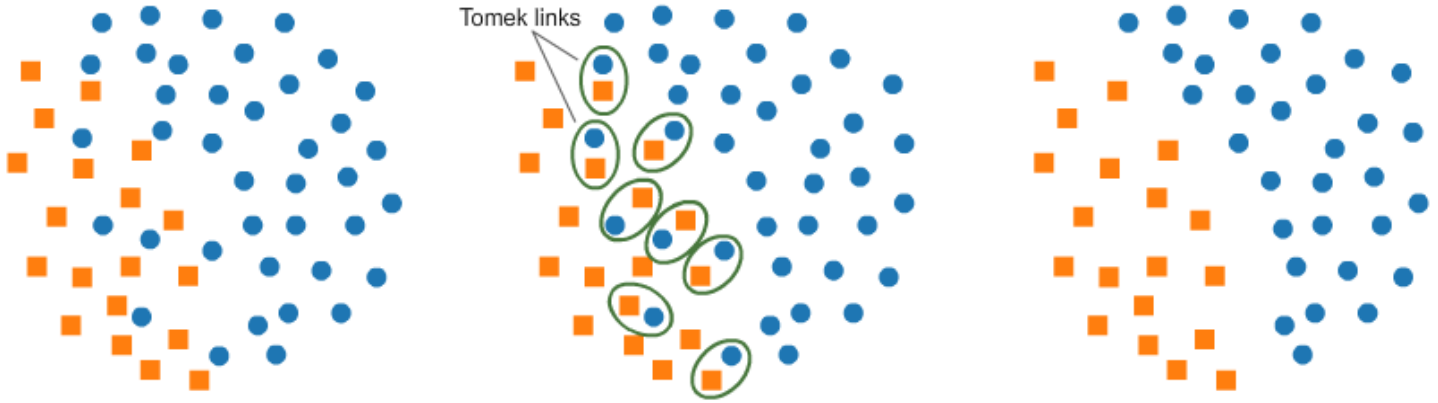
✓ Tomek link를 형성한 샘플 중 다수 범주에 속한 샘플을 제거

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 언더 샘플링(Tomek Links)

✓ Tomek link는 다수 범주의 데이터의 중심 분포는 거의 유지하면서 분류 경계를 조정하는 효과를 얻기 때문에 random undersampling에 비해 정보의 유실을 크게 줄일 수 있지만, 제거되는 샘플이 한정적이기 때문에 큰 언더 샘플링의 효과를 얻을 수는 없음



JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 언더 샘플링(Condensed Nearest Neighbor (CNN))

✓ 다수 범주에서는 하나의 샘플을 무작위로 선택하고 동시에 소수 범주에서는 모든 샘플을 선택하여 서브 데이터를 구성

✓ 원데이터를 서브 데이터를 기준으로 1-nearest neighbor(1-NN) 분류

✓ 서브 데이터를 구성할 때, 다수 범주에서 하나의 샘플을 선택하였기 때문에 반드시 1-NN을 사용하여야 함

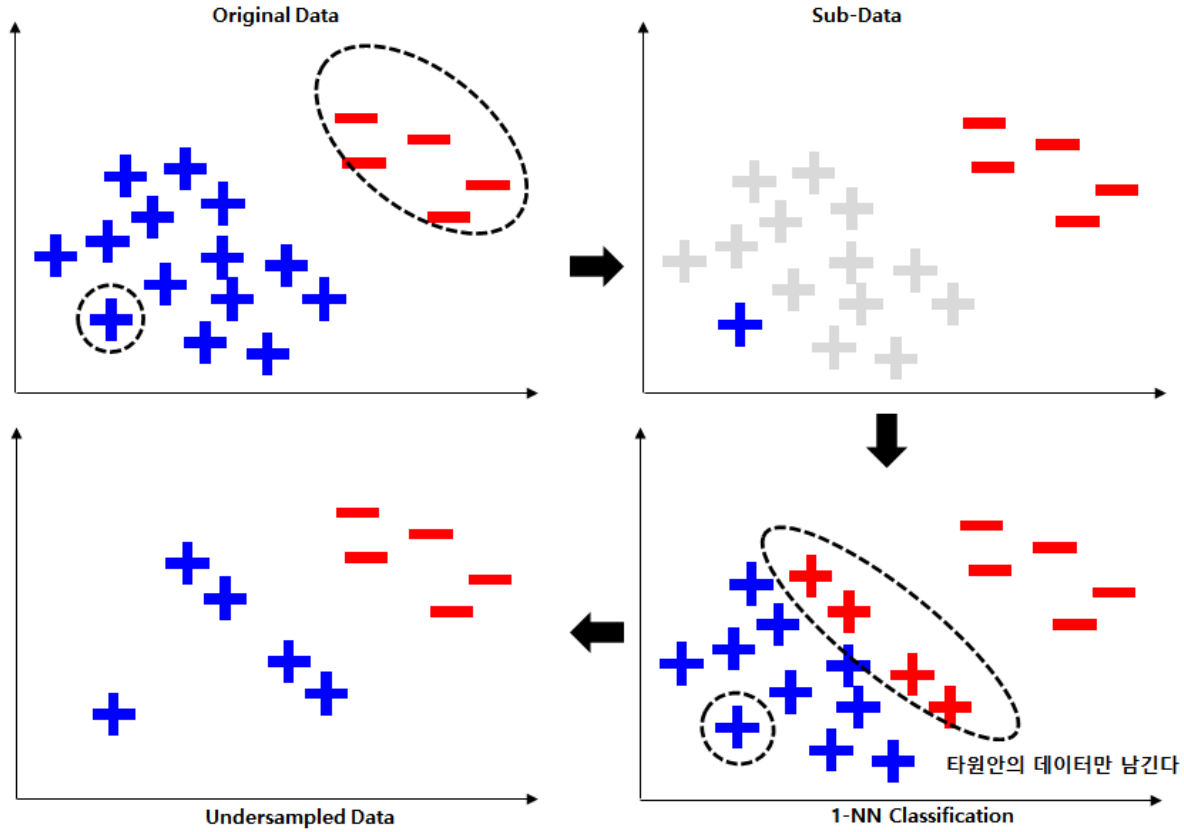
✓ 다수 범주에서 소수 범주로 분류된 샘플과 서브 데이터만을 남기고 나머지 샘플은 삭제

✓ 위 과정을 통해 다수 범주의 데이터는 다운 샘플링 되며, 이렇게 얻어진 샘플들을 이용해 분류 경계면을 학습

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 언더 샘플링(Condensed Nearest Neighbor (CNN))



JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 언더 샘플링(One-Sided Selection (OSS))

✓ Tomek link + CNN

✓ Tomek link로 다수 범주의 데이터를 제거 후 CNN을 사용하여 다시 한 번 언더 샘플링 수행

✓ Tomek link로 클래스의 경계면에 있는 데이터를 제거하고 CNN으로는 경계면에서 멀리 떨어진 다수 범주의 데이터를 제거

✓ Tomek link와 CNN의 한계를 상호 보완

1. 언더/오버 샘플링

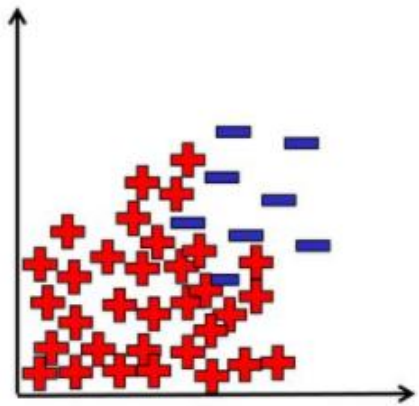
❖언더 샘플링과 오버 샘플링의 이해

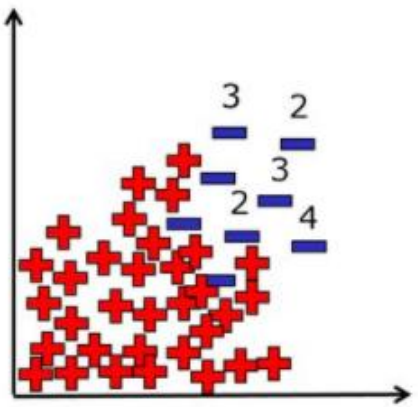
➢ 오버 샘플링(Resampling)

✓ 소수 범주의 데이터를 단순히 복제하여 데이터의 수를 늘림

✓ 단순 복제이므로 새로운 데이터를 생성하는 것은 아니지만 분류 경계면을 결정할 때 소수 클래스에 대한 가중치가 증가하기 때문에 성능 향상에 도움

✓ 소수 범주의 데이터 세트가 전체 모집단을 대표한다는 보장이 없기 때문에 소수 범주에 과적합이 발생할 가능성이 있음





1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 오버 샘플링(Synthetic Minority Oversampling TEchnique (SMOTE))

✓ 소수 범주의 데이터를 단순 복제하는 Resampling 기법과는 다르게 소수 범주의 데이터를 가상으로 만들어 냄

- 소수 범주의 데이터 중 무작위로 하나를 선택(X)
- 무작위로 선택된 데이터를 기준으로 KNN을 수행
- 선택된 K-nearest neighbor 중 하나를 임의로 선택(x(NN))
- X와 X(NN)를 잇는 라인의 임의의 위치에 가상의 데이터를 생성. 여기서 μ은 0~1사이의 균일 분포에서 추출된 임의의 값

$$x_{synthetic} = x + \mu(x(NN) - x)$$

- 소수 범주 내 모든 데이터에 대하여 과정을 반복

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢오버 샘플링(Synthetic Minority Oversampling TEchnique (SMOTE))

✓SMOTE를 구현한 대표적인 파이썬 패키지는 imbalanced-learn

✓conda install -c conda-forge imbalanced-learn 설치

< SMOTE 수행 절차 >

JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 오버 샘플링(Borderline SMOTE)

✓Borderline 부분에 대해서만 SMOTE 방식을 사용하는 것

✓소수 클래스 x_i 에 대해서 k개 주변을 탐색하고 k개 중 다수 클래스의 수를 확인. 다수 클래스의 수를 m이라고 할 때,

- k=m: Noise 관측치 (borderline이 아니다.)
- k/2<m<k: Danger 관측치 (borderline이라고 판단)
- 0≤m≤k/2: Safe 관측치 (borderline이 아니다.)

✓즉, k개 모두 다수 클래스는 아니면서 절반 이상은 다수 클래스인 경우 x_i 가 borderline에 위치해 있다고 판단

1. 언더/오버 샘플링

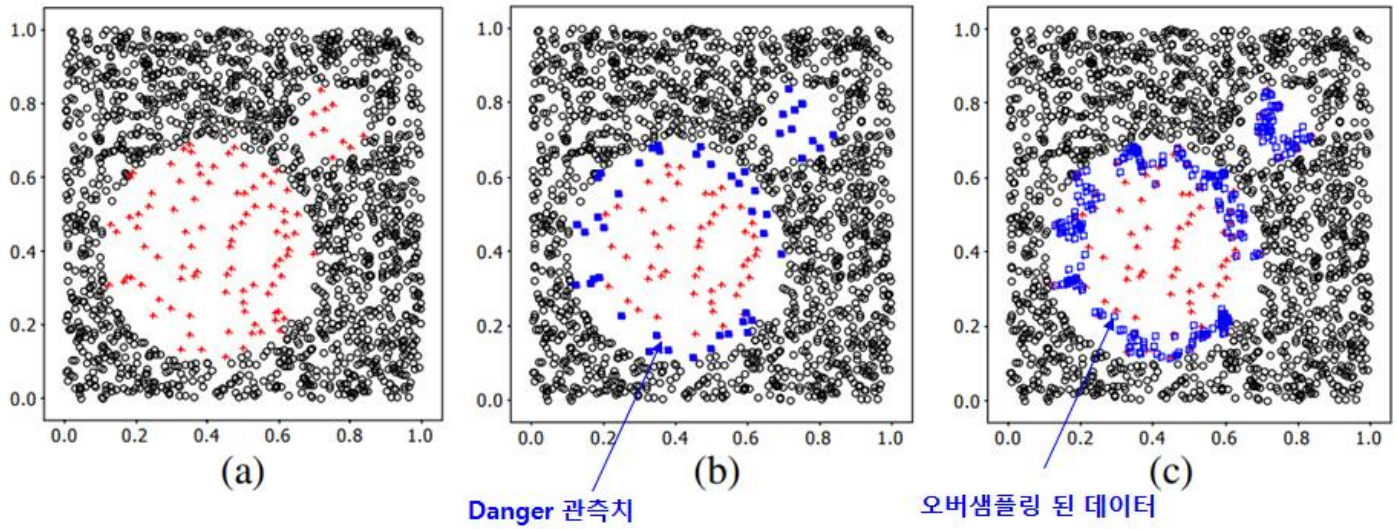
❖언더 샘플링과 오버 샘플링의 이해

➢ 오버 샘플링(Borderline SMOTE)

✓Danger 관측치에 대하여만 SMOTE 적용

✓SMOTE 적용할 때 KNN 알고리즘 적용시에는 소수 관측치의 모든 데이터를 사용

✓결과적으로 borderline 근처에만 새로운 데이터들이 생기게 됨



JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 오버 샘플링(ADAptive SYNthetic sampling approach (ADASYN))

✓ Borderline SMOTE와 유사하지만 샘플링하는 개수를 위치에 따라 다르게 적용

- 먼저, 모든 소수 범주 데이터에 대해 주변의 K개의 데이터를 탐색하고 그중 다수 범주 데이터의 비율을 계산
- $r_i = m_i/k, m_i$: 다수클래스의 수
- 모든 i 에 대한 r_i 의 계산이 끝나면 r_i 의 합이 1이 되도록 정규화
- 정규화 된 r_i 를 $\hat{r_i}$ 라 하고, 다수 클래수와 소수 클래스 간의 데이터 개수 차이를 G라고 하면, 앞에서 계산된 r 에 G를 곱하고 반올림. 이렇게 주어진 숫자가 x_i 를 기준으로 오버샘플링 될 데이터의 개수가 됨
- 모든 i 에 대하여 계산된 개수만큼 SMOTE 알고리즘을 적용하여 오버샘플링을 수행

1. 언더/오버 샘플링

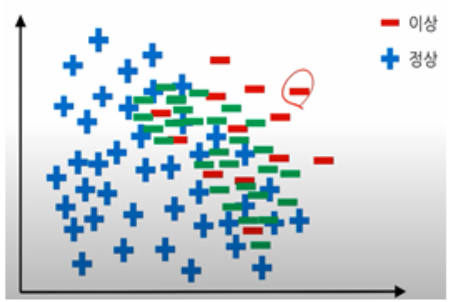
❖언더 샘플링과 오버 샘플링의 이해

➢ 오버 샘플링(ADAptive SYNthetic sampling approach (ADASYN))

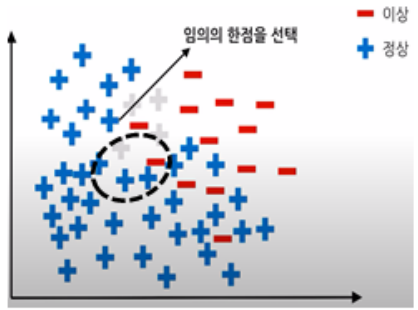
소수 클래스	r_i
1	0.4
2	0
3	0
⋮	⋮
11	0.8
12	0.6
13	1

소수 클래스	$\hat{r}_i$
1	0.1428
2	0
3	0
⋮	⋮
11	0.2857
12	0.2142
13	0.4

소수 클래스	생성되는 소수 클래스 수
1	14.28 (14)
2	0
3	0
⋮	⋮
11	28.57 (29)
12	21.42 (21)
13	40



오버샘플링 된 결과 (예시)



소수 클래스를 seed로 하여 할당된 개수만큼 SMOTE 적용

"다수 클래수 수 – 소수 클래수 수"를 곱하고 반올림

JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢오버 샘플링(GAN)

✓GAN (Generative Adversarial Nets) 는 생성자와 구분자로 구성되어 있고 모델은 딥러닝을 사용하는 최신 오버 샘플링 기법

✓무작위로 노이즈를 생성하고 생성자를 통해 가짜 샘플을 만듬

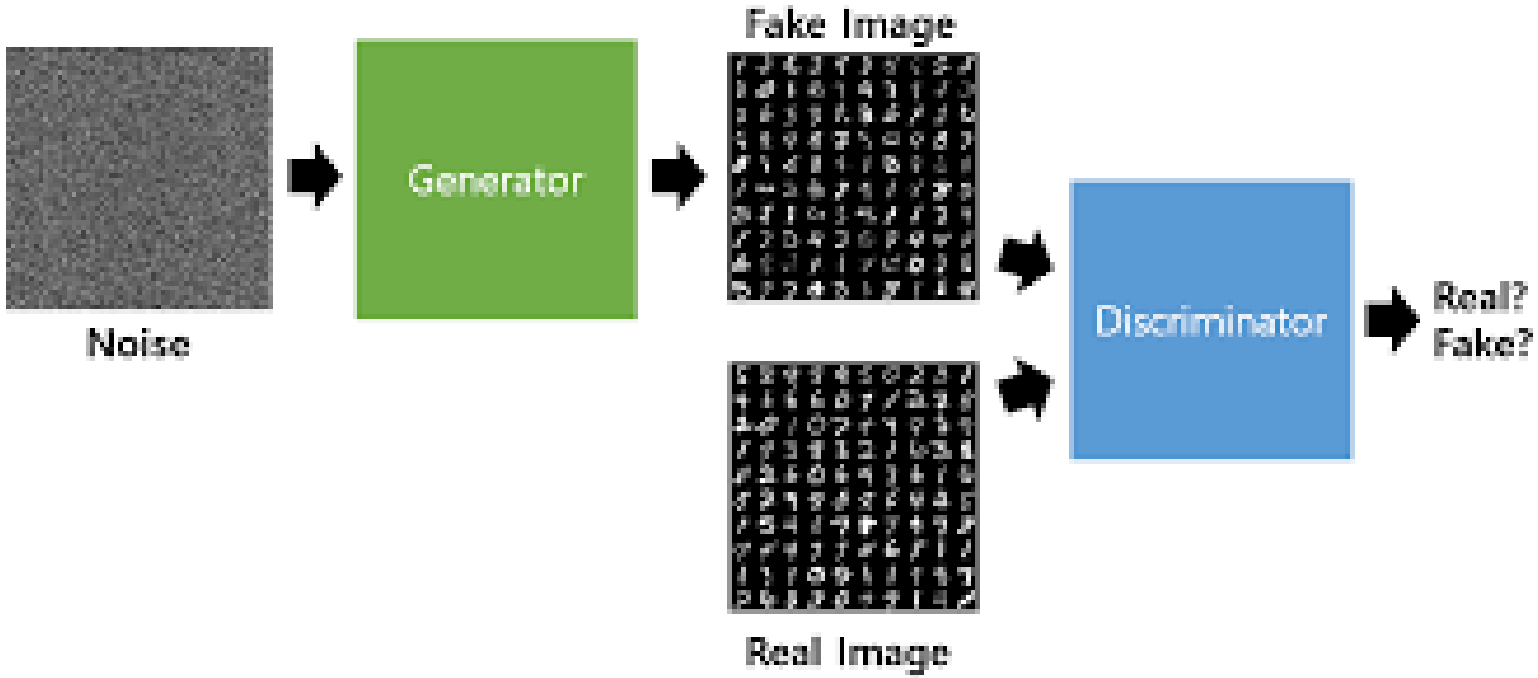
✓그 후 구분자에서 진짜 샘플과 가짜 샘플을 판별하고 너무 쉽게 판별될 경우 생성자에게 피드백을 줌

✓그러면 생성자는 더욱 진짜 샘플과 비슷한 가짜 샘플을 만들어내고 구분자에게 판별을 시킴

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢오버 샘플링(GAN)



1. 언더/오버 샘플링

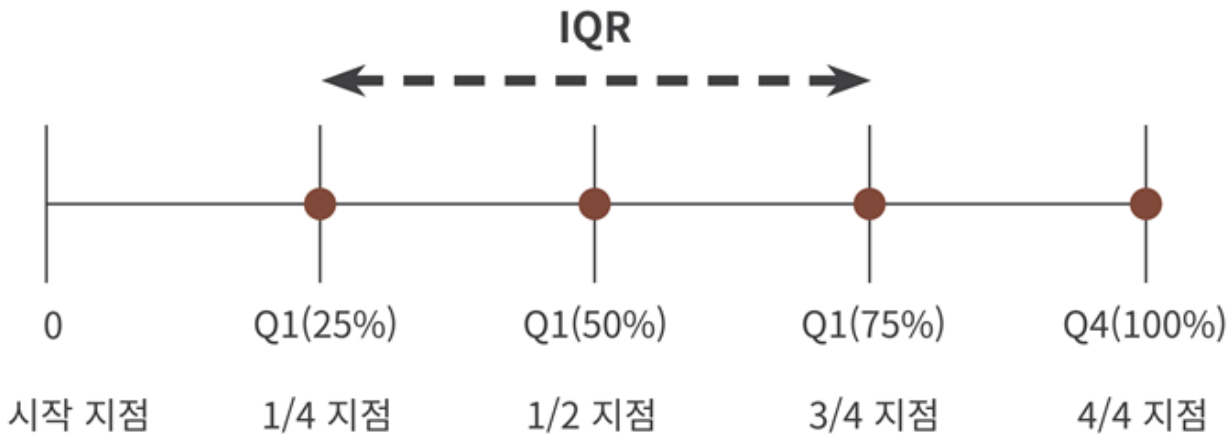
❖언더 샘플링과 오버 샘플링의 이해

➢ 이상치 데이터 제거 후 모델 학습/예측/평가

✓ 이상치 데이터(outlier) : 전체 데이터의 패턴에서 벗어난 이상값을 가진 데이터

✓ 이상치를 찾아내는 방법과 이들 데이터를 제거한 뒤 다시 모델 평가

✓ IQR(Inter Quantile Range) 방식적용



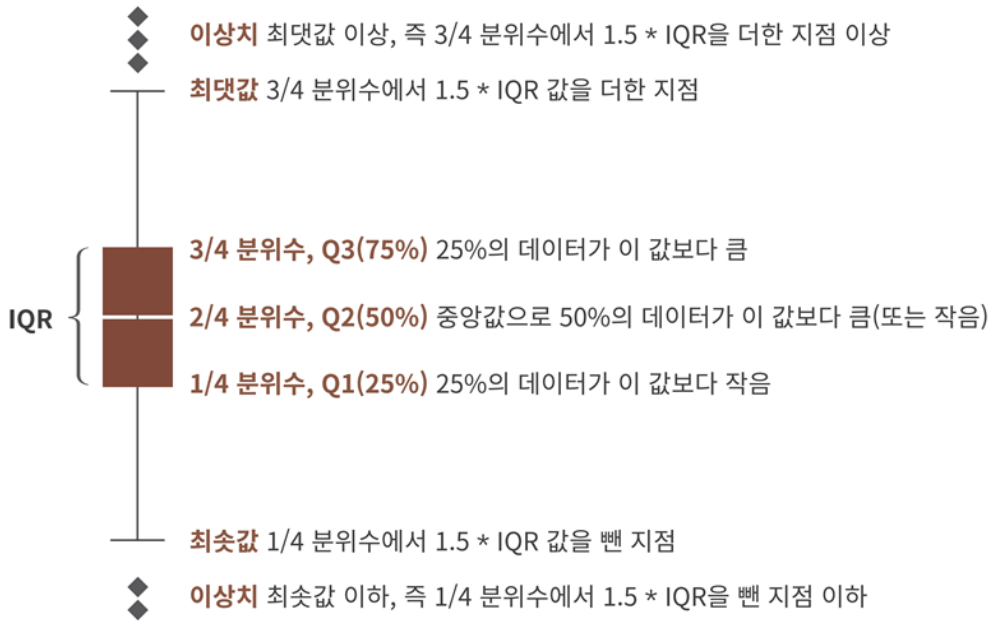
1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ 이상치 데이터 제거 후 모델 학습/예측/평가

✓ 이상치 데이터 검출 방식

- 매우 많은 피처가 있을 경우, 결정값(레이블)과 가장 상관성이 높은 피처들을 위주로 이상치를 검출



JEONJU UNIVERSITY

1. 언더/오버 샘플링

❖언더 샘플링과 오버 샘플링의 이해

➢ SMOTE 오버 샘플링 적용 후 모델 학습/예측/평가

데이터 가공 유형	머신러닝 알고리즘	평가지표		
		정밀도	재현율	ROC-AUC
원본 데이터 가공 없음	로지스틱 회귀	0.8738	0.6081	0.9707
	LightGBM	0.9492	0.7568	0.9797
데이터 로그 변환	로지스틱 회귀	0.8824	0.6081	0.9721
	LightGBM	0.9575	0.7635	0.9786
이상치 데이터 제거	로지스틱 회귀	0.8829	0.6712	0.9747
	LightGBM	0.9690	0.8288	0.9831
SMOTE 오버 샘플링	로지스틱 회귀	0.0540	0.9247	0.9737
	LightGBM	0.9323	0.8493	0.9789

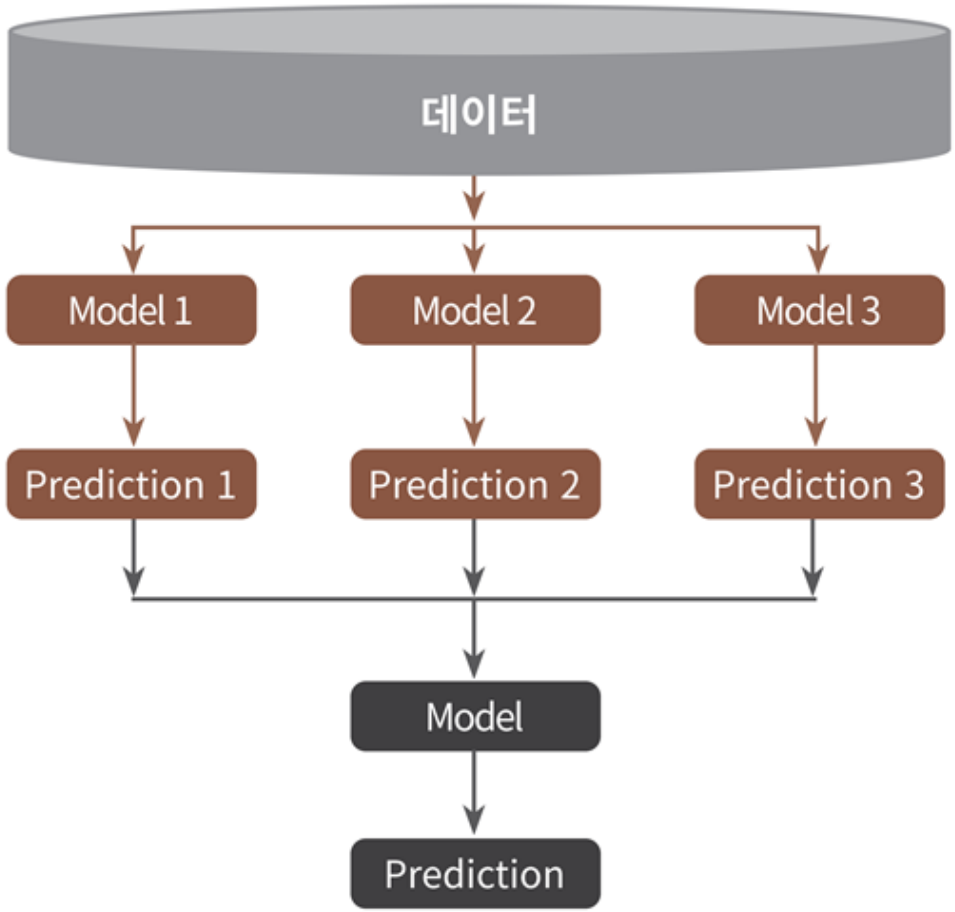
2. 스태킹

❖스태킹 앙상블

- 개별적인 여러 알고리즘을 서로 결합해 예측결과 도출
- 차이점 : 개별 알고리즘으로 예측한 데이터를 기반으로 다시 예측을 수행
- 즉, 개별 알고리즘의 예측 결과 데이터 세트를 최종적인 메타 데이터 세트로 만들어 별도의 ML 알고리즘으로 최종학습을 수행하고, 테스트 데이터를 기반으로 다시 최종 예측을 수행하는 방식
- 개별적인 기반모델과 이 개별 기반 모델의 예측데이터를 학습데이터로 만들어서 학습하는 최종 메타모델이 있음
- 스태킹 모델의 핵심은 여러 개별 모델의 예측 데이터를 각각 스태킹 형태로 최종 메타 모델의 학습용 피처 데이터 세트와 테스트용 피처 데이터 세트를 만드는 것
- 스태킹을 적용할 때는 많은 개별 모델이 필요
- 일반적으로 성능이 비슷한 모델을 결합해 좀 더 나은 성능향상을 도출하기 위해 적용

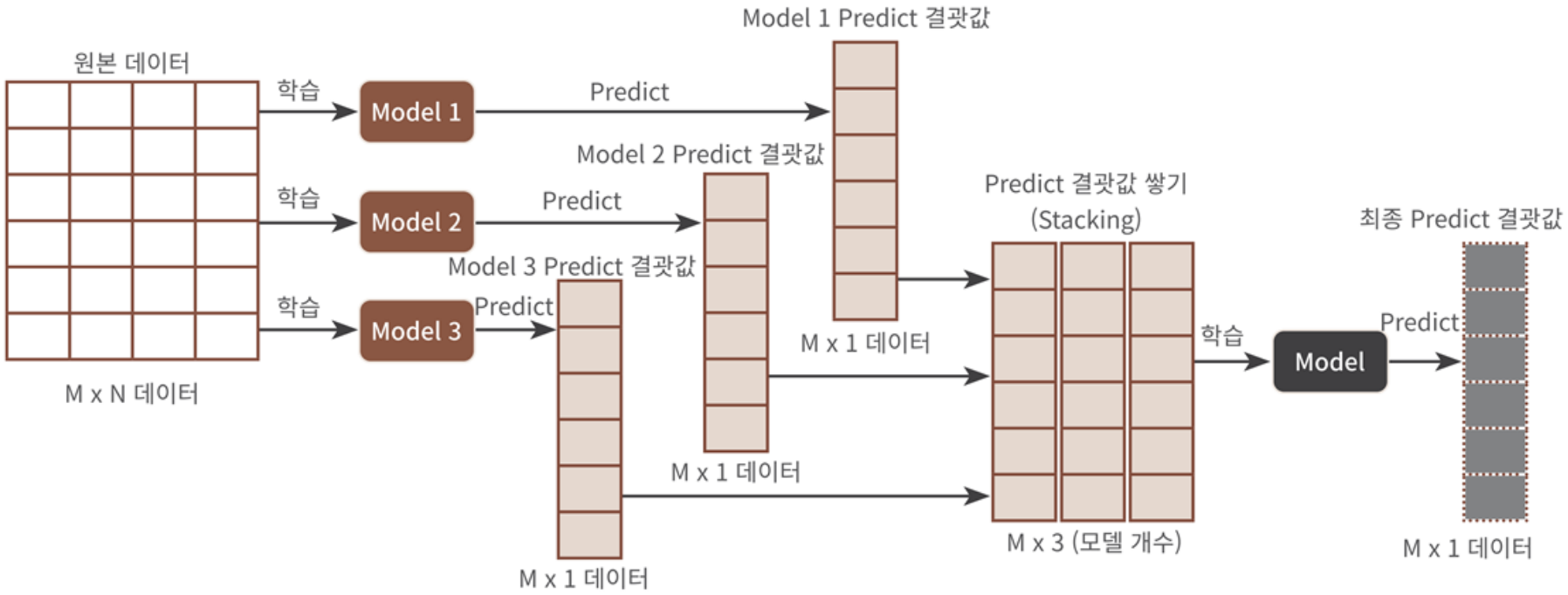
2. 스태킹

❖스태킹 앙상블


데이터
Model 1
Model 2
Model 3
Prediction 1
Prediction 2
Prediction 3
Model
Prediction

2. 스태킹

❖스태킹 앙상블


원본 데이터
M x N 데이터
학습
Model 1
Predict
Model 1 Predict 결괏값
M x 1 데이터
학습
Model 2
Predict
Model 2 Predict 결괏값
M x 1 데이터
학습
Model 3
Predict
Model 3 Predict 결괏값
M x 1 데이터
Predict 결괏값 쌓기
(Stacking)
M x 3 (모델 개수)
학습
Model
Predict
최종 Predict 결괏값
M x 1 데이터

2. 스태킹

❖스태킹 앙상블

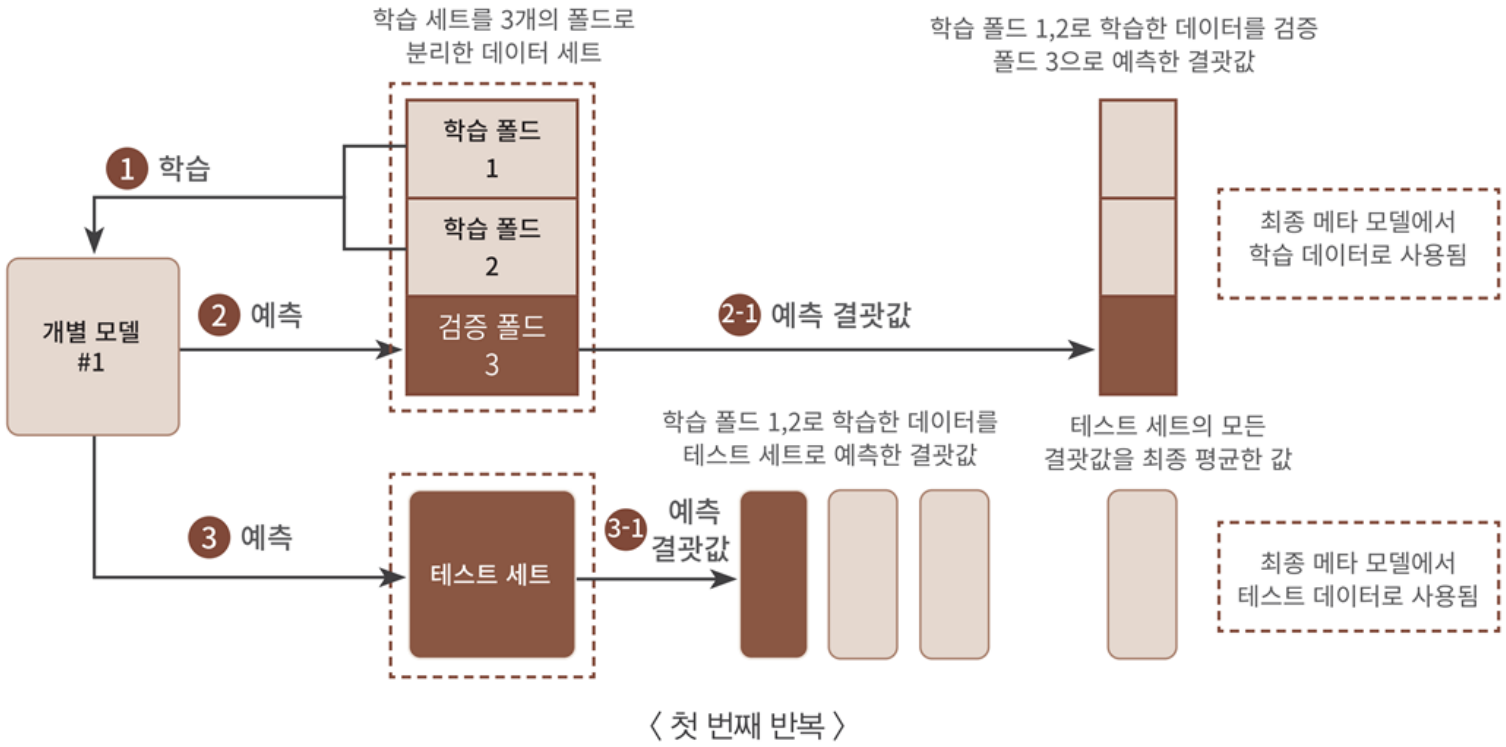
➢ CV 셋 기반의 Stacking

- ✓ 과적합을 개선하기 위해 최종 메타 모델을 위한 데이터 세트를 만들때 교차 검증 기반으로 예측된 결과 데이터 세트를 이용
- ✓ step1: 각 모델별로 원본 학습/테스트 데이터를 예측한 결과 값을 기반으로 메타 모델을 위한 학습용/테스트용 데이터를 생성
- ✓ step2: step 1에서 개별 모델들이 생성한 학습용 데이터를 모두 스태킹 형태로 합쳐서 메타모델이 학습할 최종 학습용 데이터 세트를 생성. 마찬가지로 각 모델들이 생성한 테스트용 데이터를 모두 스태킹 형태로 합쳐서 메타 모델이 예측할 최종 테스트 데이터세트 생성. 메타모델은 최종적으로 생성된 학습데이터세트와 원본 학습데이터의 레이블 데이터를 기반으로 학습한 뒤, 최종적으로 생성된 테스트 세트를 예측하고, 원본 테스트 데이터의 레이블 데이터를 기반으로 평가

2. 스태킹

❖스태킹 앙상블

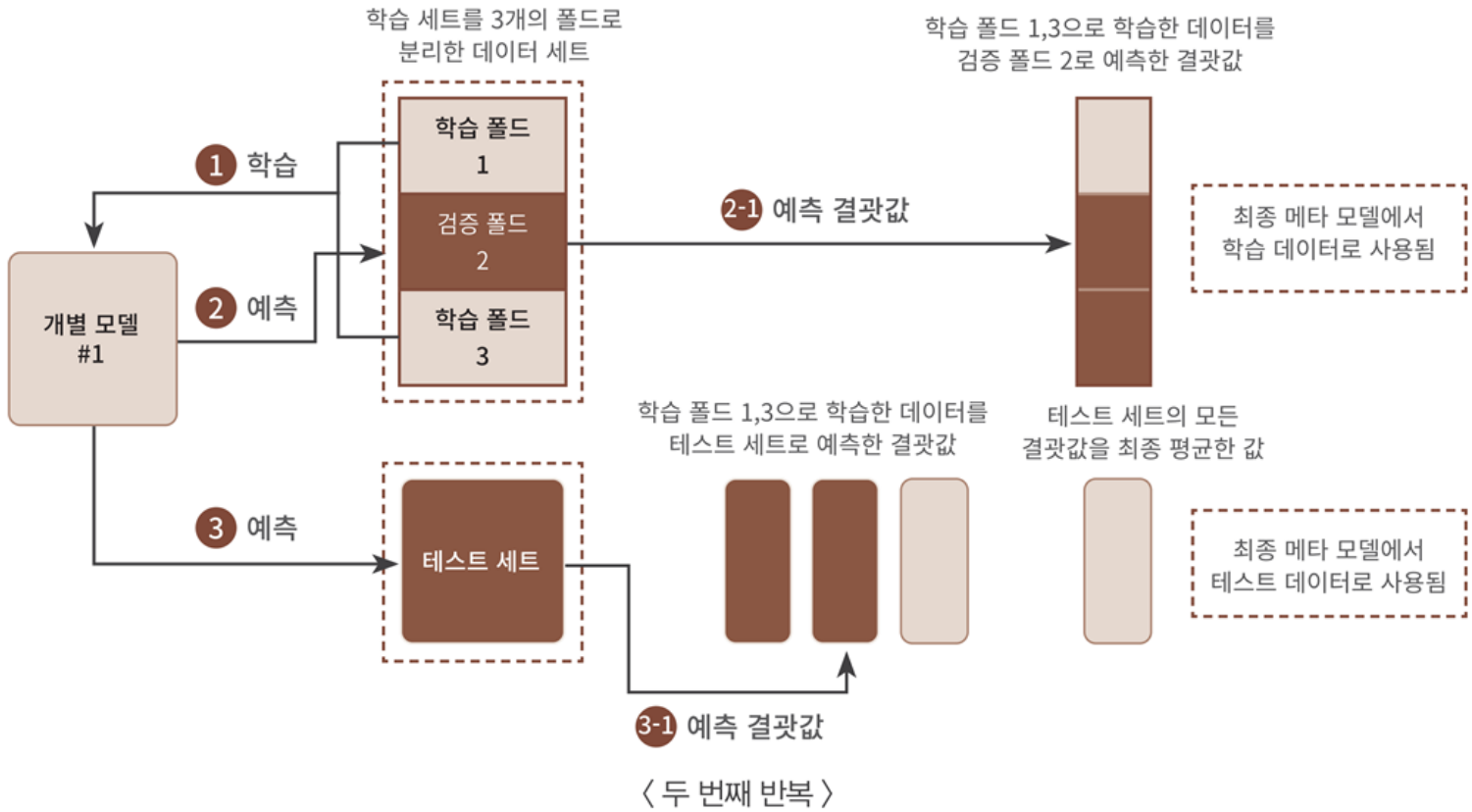
➢ CV 셋 기반의 Stacking



2. 스태킹

❖스태킹 앙상블

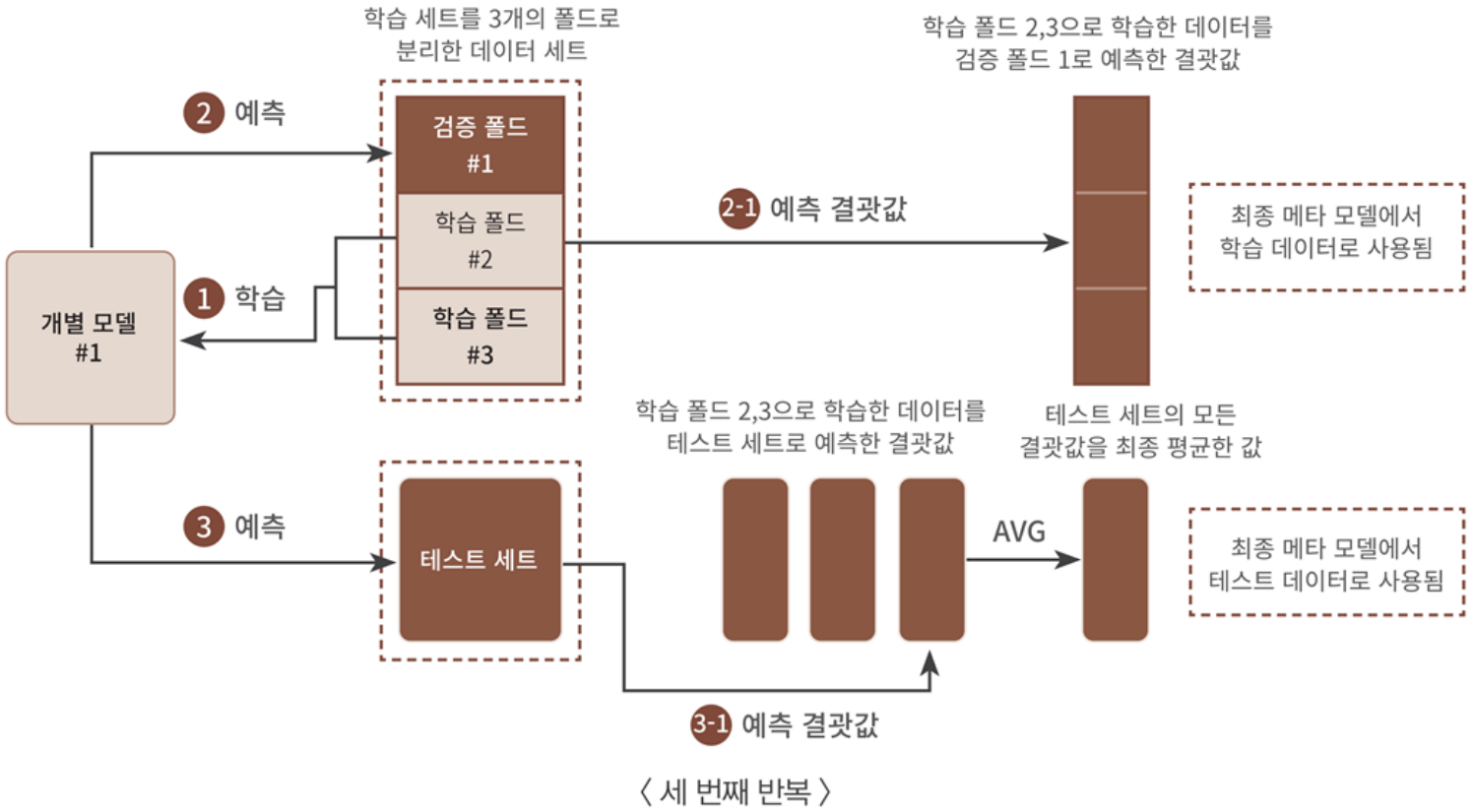
➢ CV 셋 기반의 Stacking



2. 스태킹

❖스태킹 앙상블

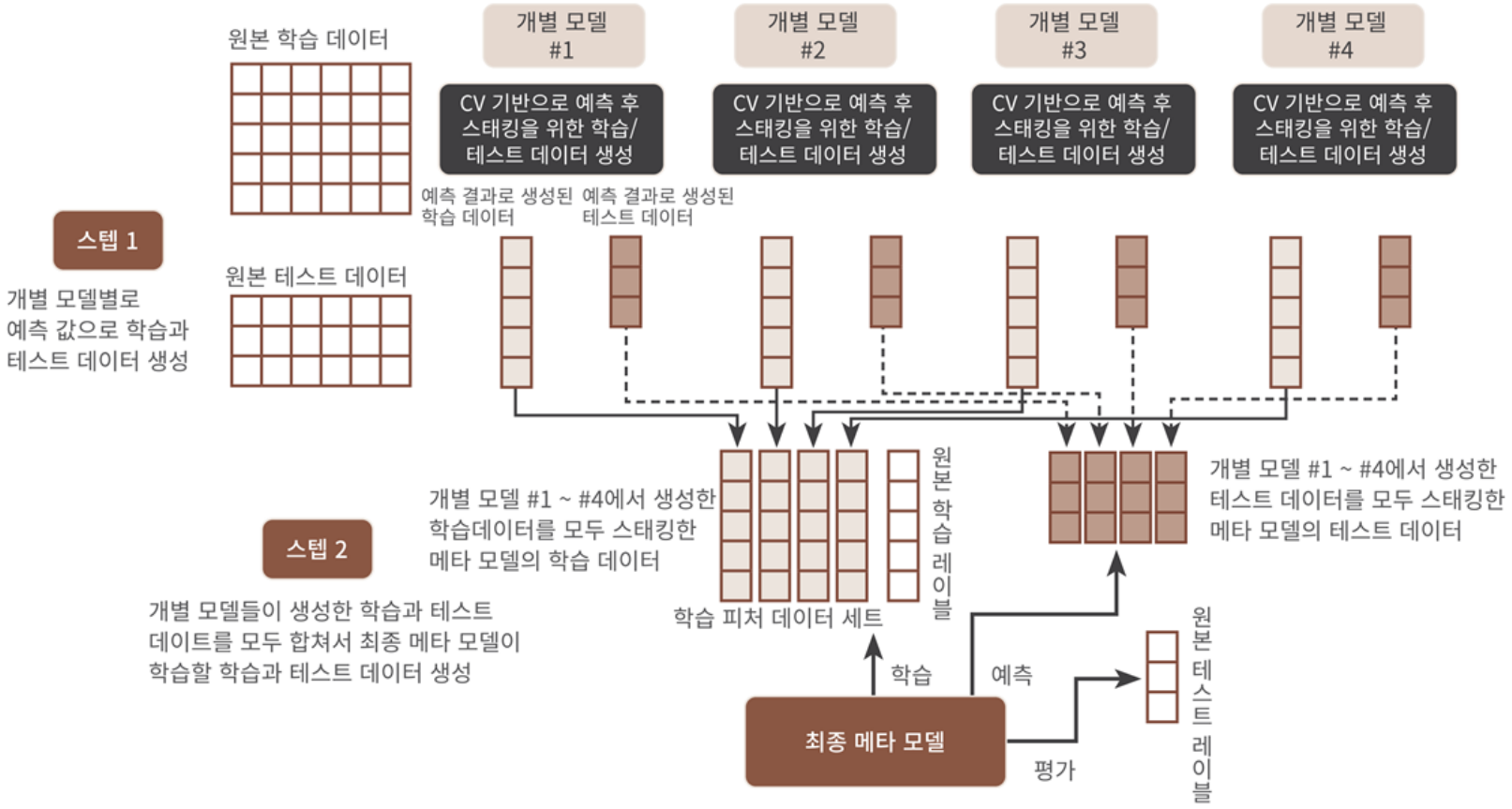
➢ CV 셋 기반의 Stacking



2. 스태킹

❖스태킹 앙상블

➢ CV 셋 기반의 Stacking



Thank You !

JEONJU UNIVERSITY
The Place for Superstars