함께 해요! 첫 번째 Kaggle

2020. 11. 14. Sat
강천성
캐글코리아

소개

- 강천성
- 신약개발 인공지능 스타트업 스탠다임

  AI Scientist
- 캐글 코리아 운영진
- TMT with TMI

E-mail : kcs93023@gmail.com

빅데이터 시대,
4차 산업혁명 시대
무엇이 가장 중요할까요?

데이터

결정

경험, 직관

결정

Data-driven Approach

Data-driven Approach

데이터 안에 무엇이 있길래?

Pattern

패턴

머신 러닝?

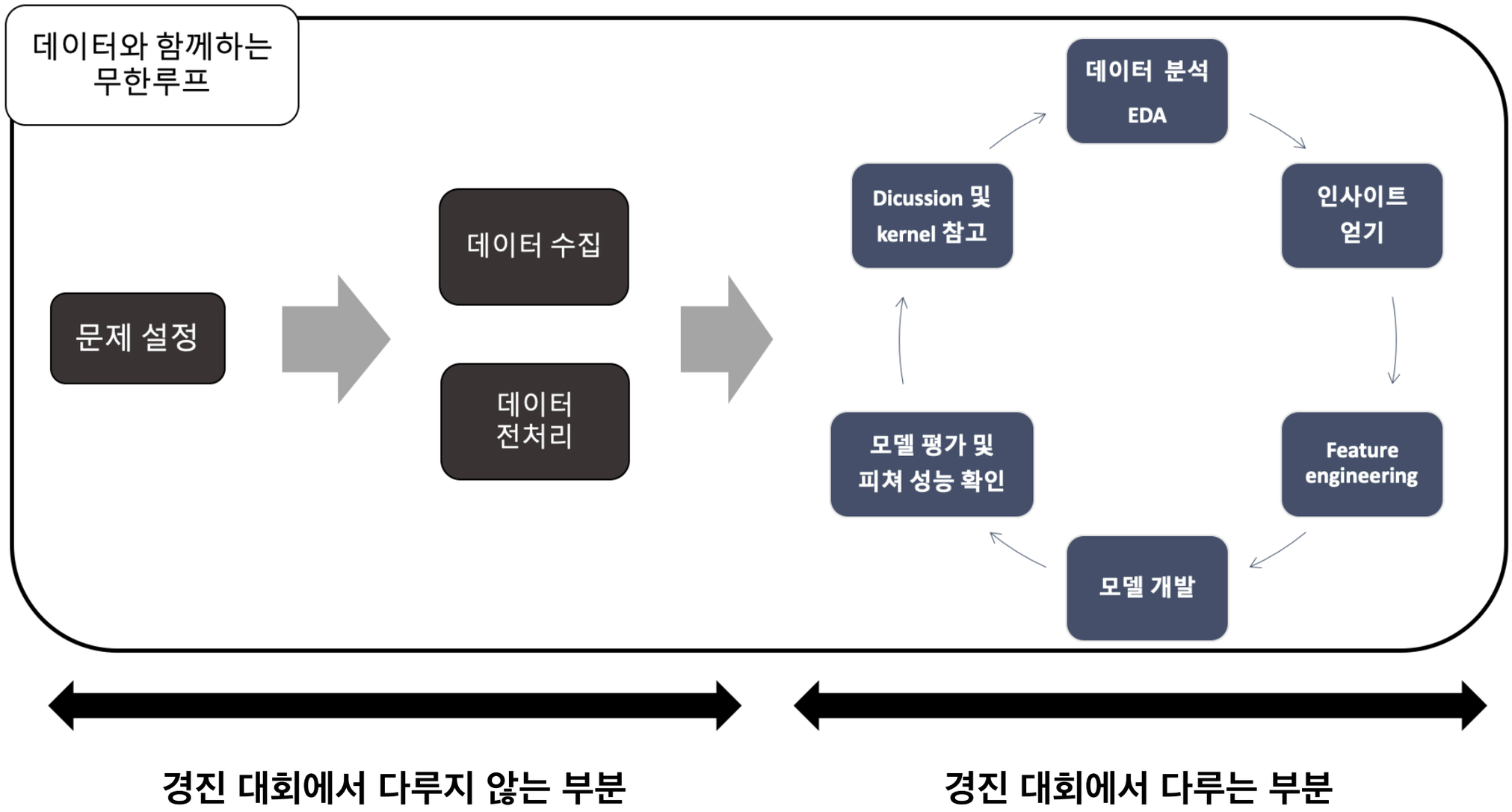
Pattern recognition

머신 러닝?

Make general function(conditions)
to obtain goal(minimize loss)

머신 러닝?

Learn statistics(correlation) between feature vs feature / feature vs target


데이터와 함께하는
무한루프
문제 설정
데이터 수집
데이터
전처리
데이터 분석
EDA
인사이트
얻기
Feature
engineering
모델 개발
모델 평가 및
피쳐 성능 확인
Dicussion 및
kernel 참고
경진 대회에서 다루지 않는 부분
경진 대회에서 다루는 부분

머신러닝을
공부하려면?

Data와 사람(자료)이
많은 곳으로 가라.

캐글 코리아

Kaggle Korea

Non-Profit Facebook Group Community

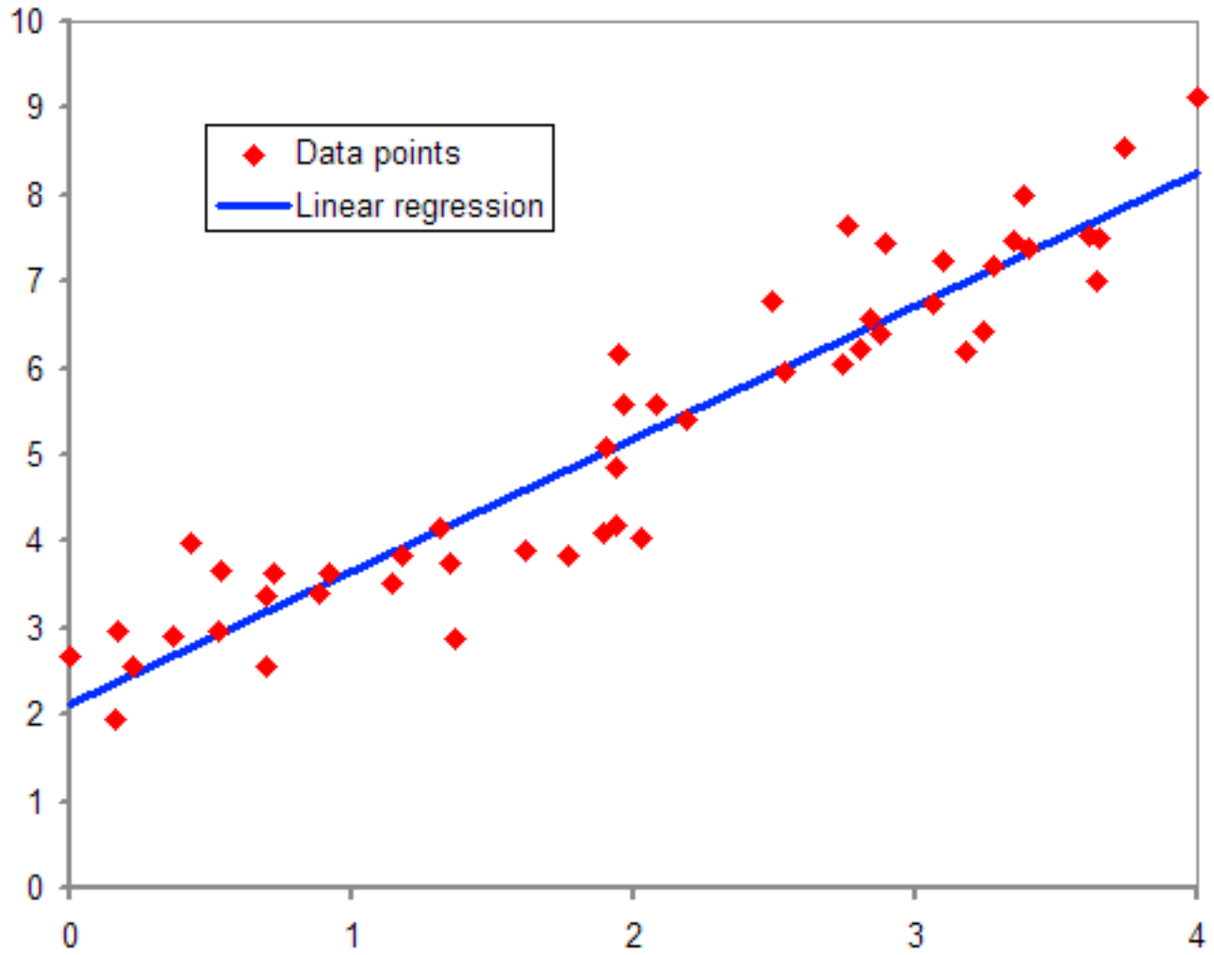
함께 공부해서, 함께 나눕시다

Study Together, Share Together

Q&A

1. Scikit-learn 기반 분류 모델

01 로지스틱 회귀 분석



로지스틱 회귀 분석?

선형 회귀 모델에서 변형된 모델로 Odds 라는 어떤 일이 발생할 상대적인 비율 개념을 도입

$$y = wx + b$$

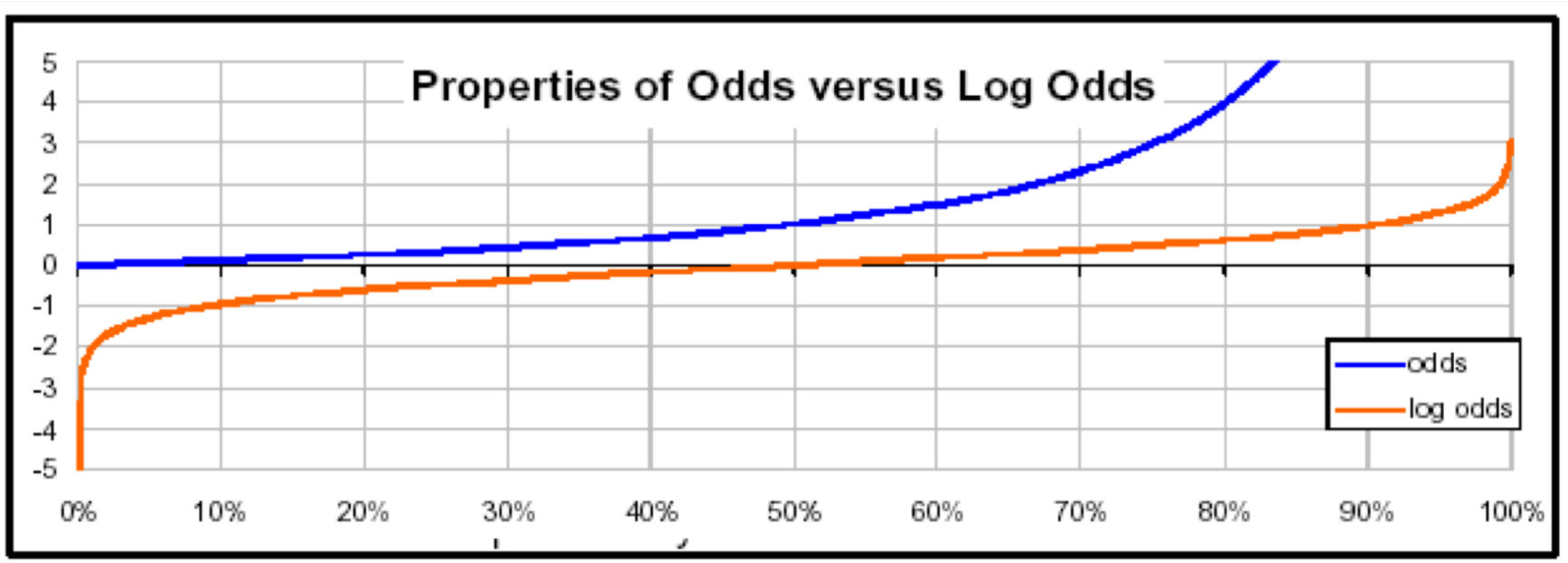
+ 확률 개념

$$Odds = \frac{p}{1-p}$$

p : 어떤 일이 발생할 확률

- 선형 회귀: https://ko.wikipedia.org/wiki/%EC%84%A0%ED%98%95_%ED%9A%8C%EA%B7%80

01 로지스틱 회귀 분석

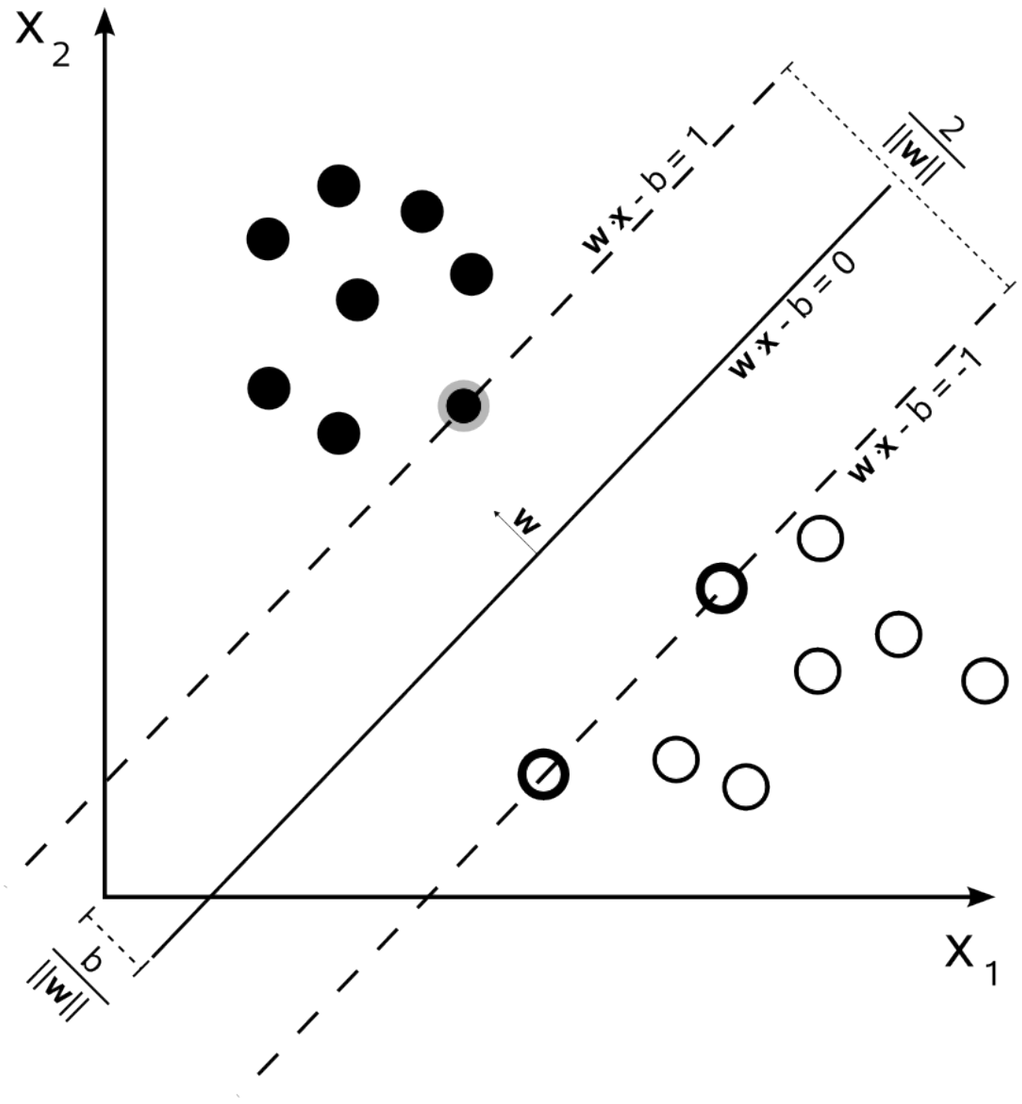


$$ln(\frac{Y}{1-Y}) = wx + b$$

$$y = \frac{1}{1 + \exp^{-(wx+b)}}$$

- Maximum Likelihood Estimation, 최대 우도 추정 : https://ratsgo.github.io/statistics/2017/09/23/MLE/

01 서포트 벡터 머신



SVM?

주어진 데이터를 바탕으로하여 클래스 사이의 간격 (Margin, 마진)을 최대화 하는 데이터 포인트 (Support Vector, 서포트 벡터)를 찾아내고, 그 서포트 벡터에 대해 수직인 경계를 통해 데이터를 분류하는 알고리즘

경험적 위험 최소화 vs 구조적 위험 최소화

- 경험적 위험 최소화 (Empirical Risk Minimization, ERM)
  - 훈련 데이터에 대해 손실을 최소화
  - 학습 알고리즘의 목표
  - 신경망, 결정트리, 선형 회귀 등..
- 구조적 위험 최소화(Structural Risk Minimization, SRM)
  - 관찰하지 않은(Unseen) 데이터에 대해서도 위험을 최소화
  - 오차 최소화를 일반화 시키는 것

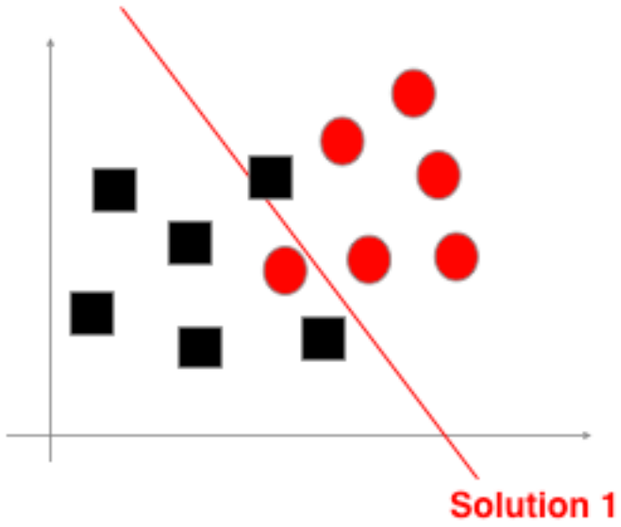
- Support Vector Machine, 서포트 벡터 머신 : https://ko.wikipedia.org/wiki/%EC%84%9C%ED%8F%AC%ED%8A%B8_%EB%B2%A1%ED%84%B0_%EB%A8%B8%EC%8B%A0

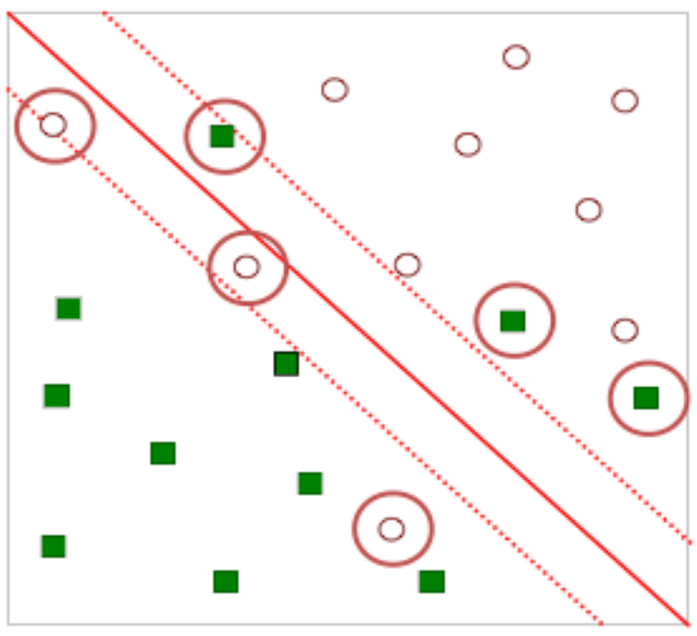
01 서포트 벡터 머신

Cost: Soft or Hard

Soft Margin은 유연한 경계면을 만들어내고, Hard Margin은 분명하게 나누는 경계면을 만들어냅니다.

다음과 같은 데이터 분포는 직선으로 두개의 클래스를 나누는 경계면을 만들기 어렵습니다.
Soft Margin은 경계면을 조금씩 넘어가는 데이터들(비용, Cost)을 감수하면서 가장 최선의 경계면을 찾습니다.



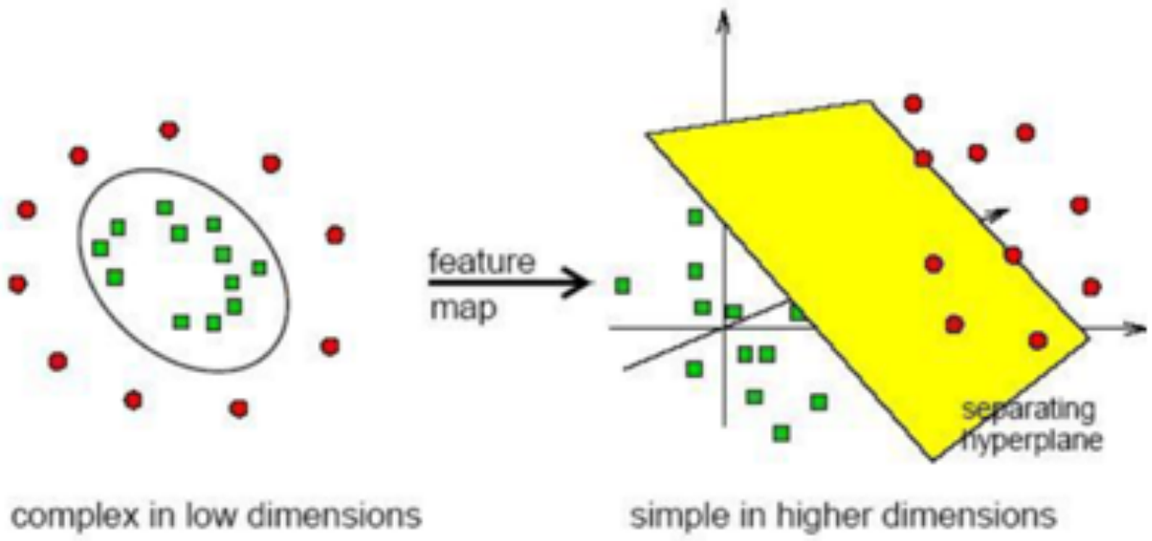


01 서포트 벡터 머신

저차원을 고차원으로 Kernel Trick

다음과 같이 저차원에서는 선형 분리가 되지 않을 수 있지만, 고차원에서는 선형 분리가 가능할 수 있습니다.
이러한 원리를 바탕으로 선형 분리가 불가능한 저차원 데이터를 선형 분리가 가능한 어떤 고차원으로 보내 선형 분리를 할 수 있습니다.

여기에서 Kernel 함수라 불리는 매핑 함수를 사용하는데, 실제로 고차원에서 연산하지 않고 저차원에서 연산하더라도 고차원에서 연산한 것과 같은 결과를 내준다 하여 Kernel Trick이라 부릅니다.

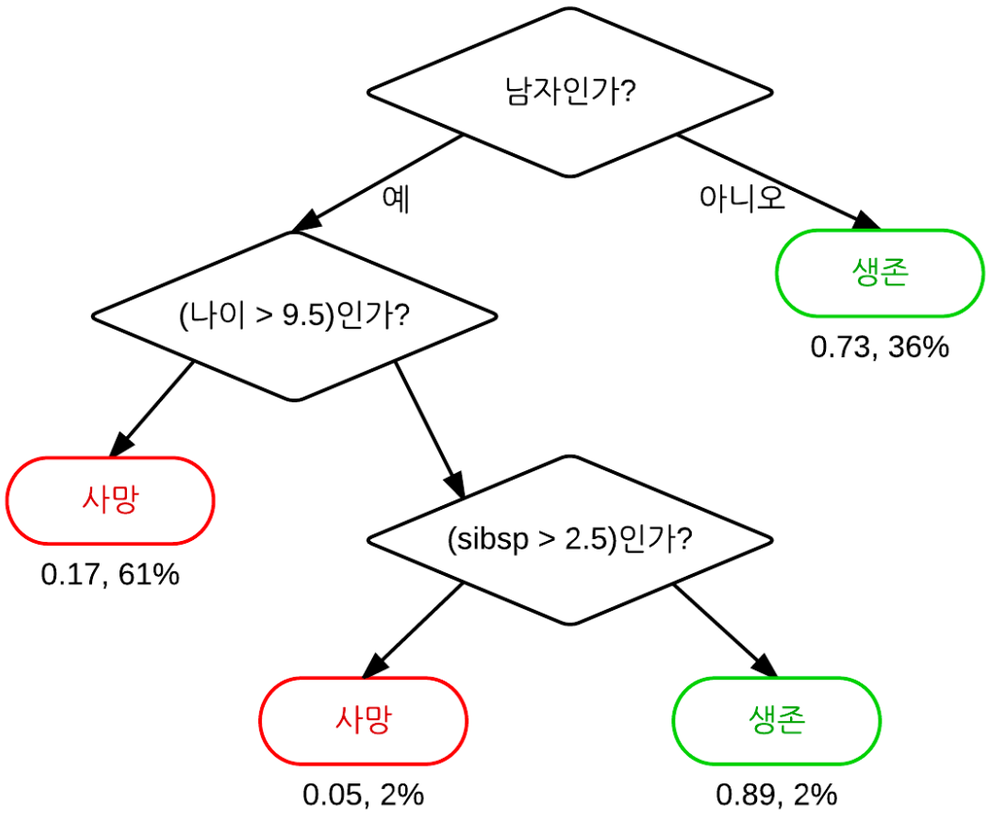


대표적인 Kernel 함수

- Linear (선형 함수)
- Poly (다항식 함수)
- RBF (방사기저함수)
- Hyper-Tangent (쌍곡선 탄젠트 함수)

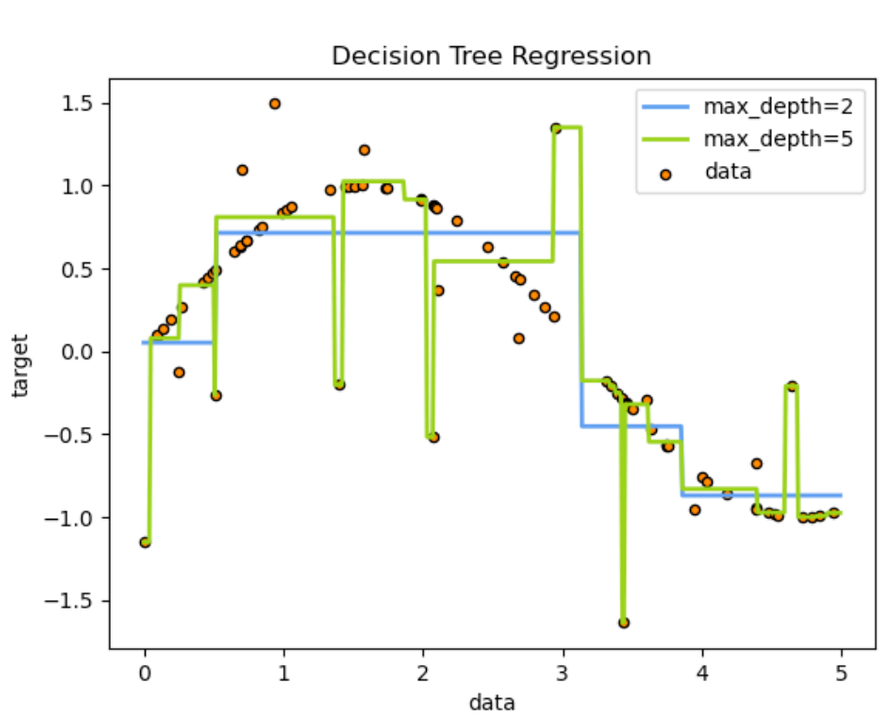
- Support Vector Machine : https://www.devteam.space/blog/top-10-data-mining-algorithms/

01 결정트리 & 랜덤포레스트



결정트리?

- 사람의 논리적 사고 방식을 모사하는 분류 방법론
- IF-THEN rule 조합으로 class 분류
- Greedy 한 알고리즘 (최적 트리 생성을 보장하지 않음)
- 축에 직교하는 분기점
- 변수가 가진 정보량에 따라 트리를 분기 (지니계수, 엔트로피)

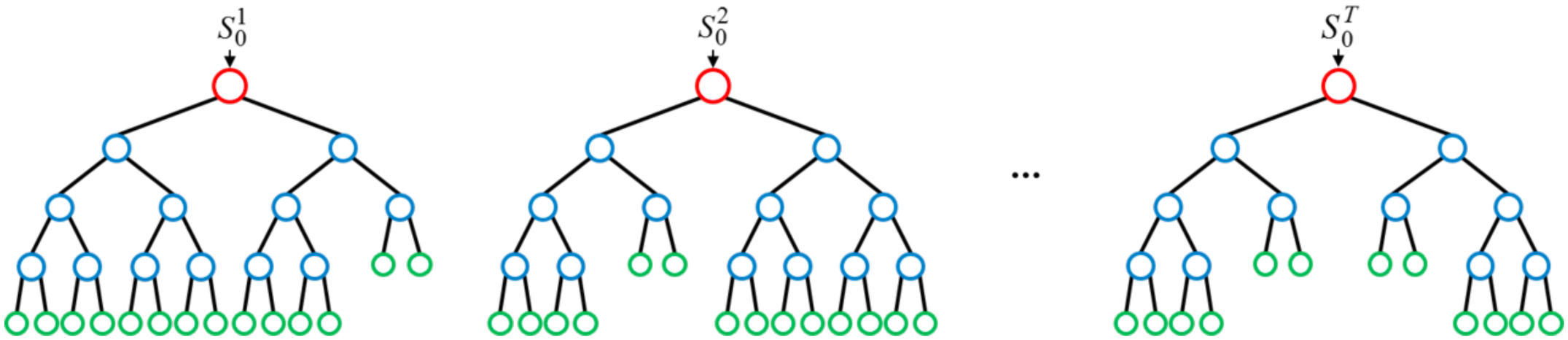


- Decision Tree : https://scikit-learn.org/stable/modules/tree.html

01 결정트리 & 랜덤포레스트

랜덤 포레스트?

- 결정 트리를 여러 개 만들어 숲을 만드는 알고리즘
- 단지 트리를 여러개 만드는 것이 아닌 데이터 샘플과 변수를 샘플링하여 앙상블 하는 부트스트래핑 방식의 앙상블 알고리즘
- 간편하고 빠른 학습 및 테스트 알고리즘
- 변수 소거 없이 수천개의 입력 변수들을 다루는 것이 가능
- 좋은 일반화 성능



- Random Forest : https://ko.wikipedia.org/wiki/%EB%9E%9C%EB%8D%A4_%ED%8F%AC%EB%A0%88%EC%8A%A4%ED%8A%B8

01 XGBoost & LightGBM

XGBoost?

- Boosting Tree 알고리즘
- Gradient Boosting Model 대비 빠른 수행 시간
- 과적합 규제 기능
- 가지치기 기능
- 자체 교차 검증 기능
- 결측치 자체 처리
- Early Stopping 기능
- 균형 트리 분할 방식 (대칭 트리 형성)

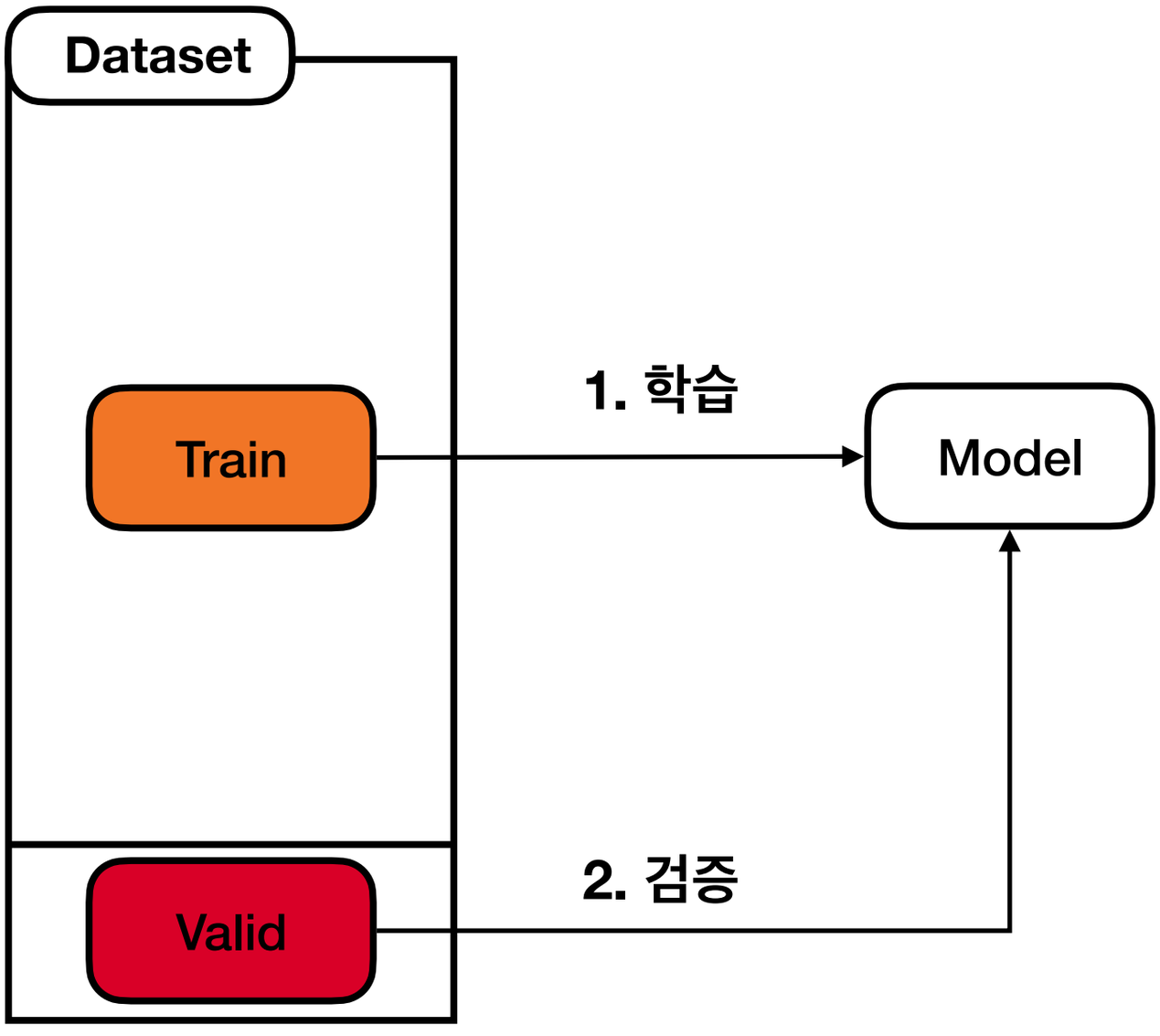
LightGBM?

- XGBoost의 장점 +
- XGBoost보다 가볍고 빠른 모델, 하지만 더 나은 학습 성능
- 리프 중심 트리 분할 방식 (비대칭 트리 형성)
- 적은 데이터에서 과적합 우려

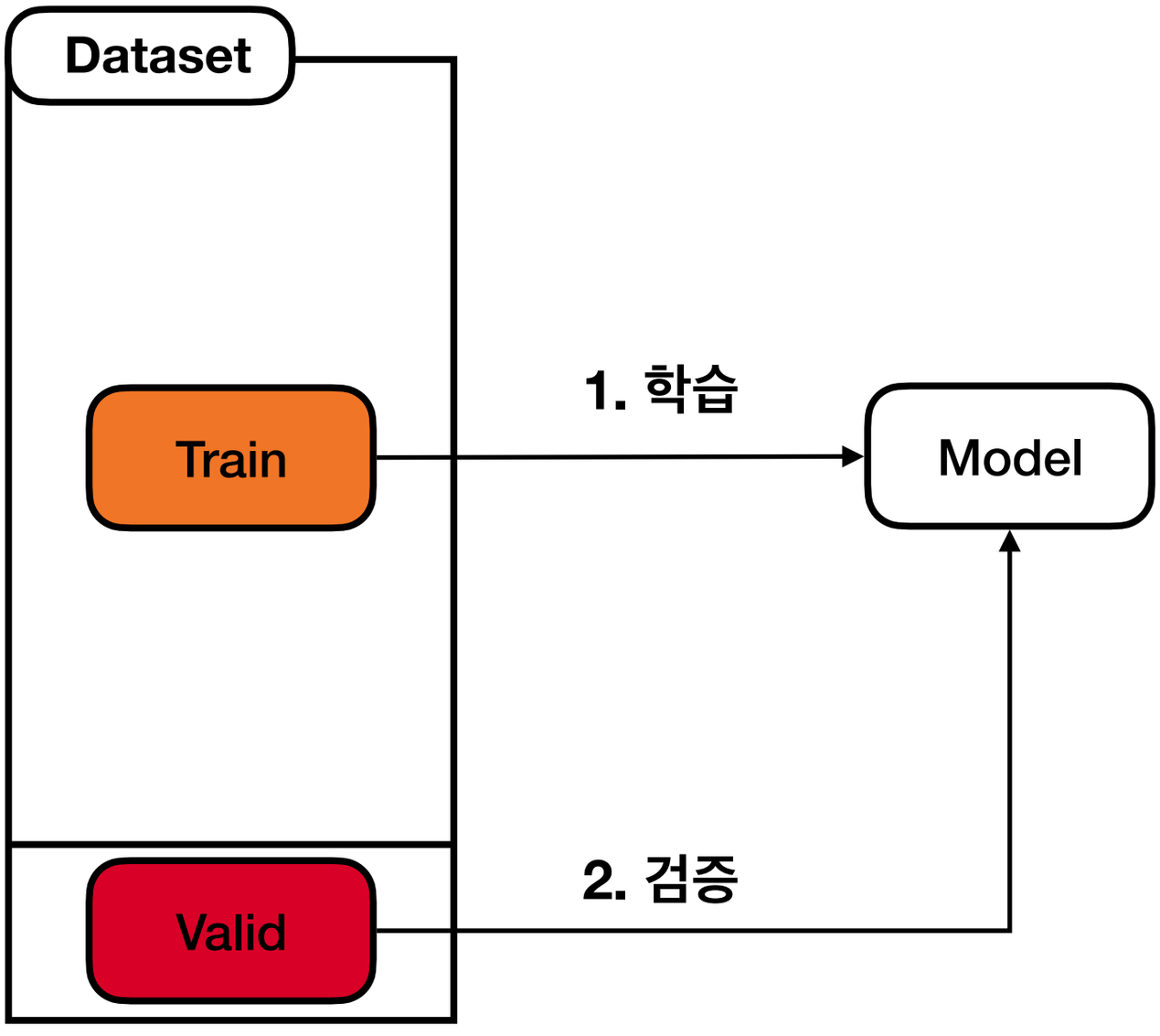
Q&A

2. k-Fold Cross Validation

02 k-Fold Cross Validation

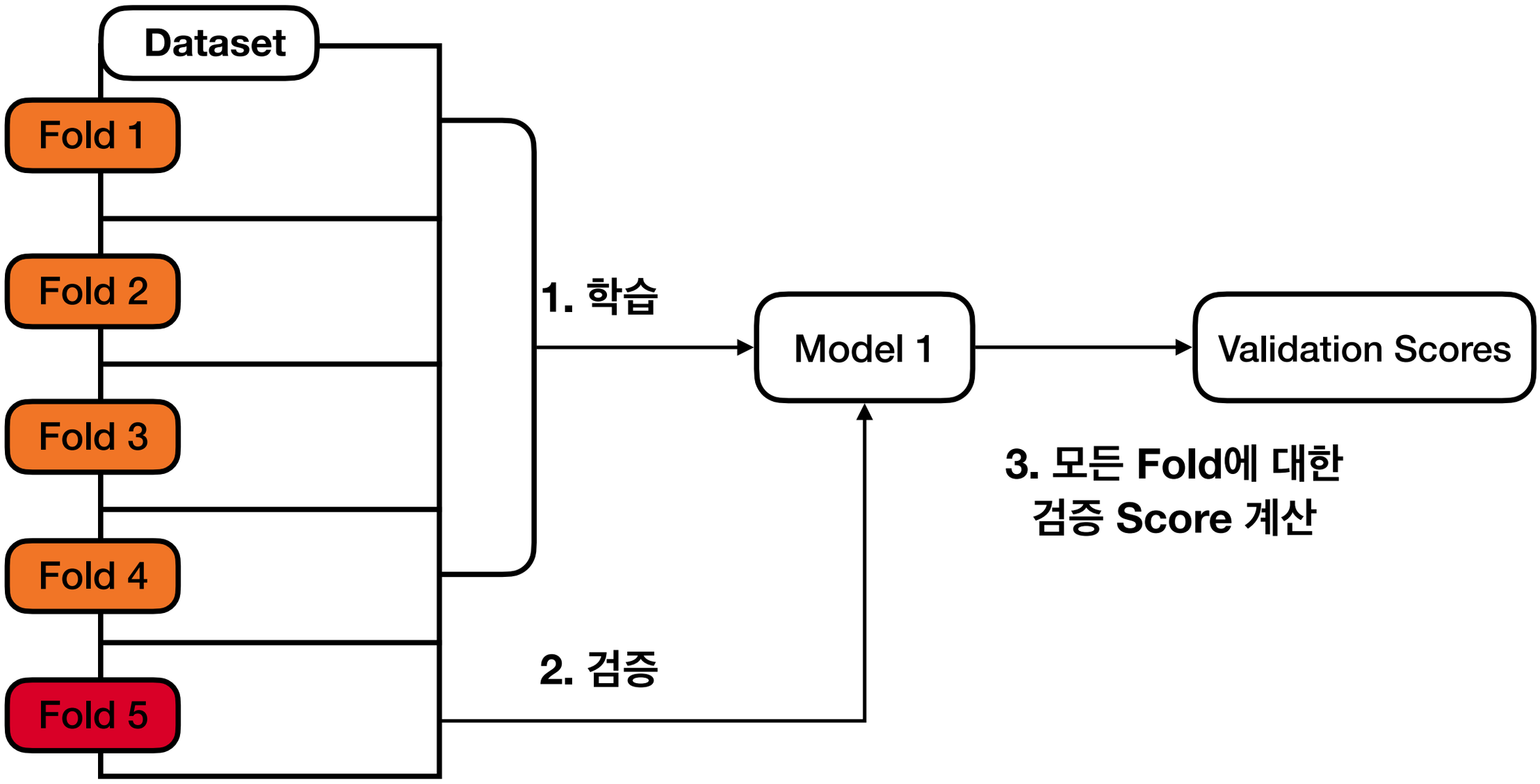


02 k-Fold Cross Validation



충분한 검증이 되는가?

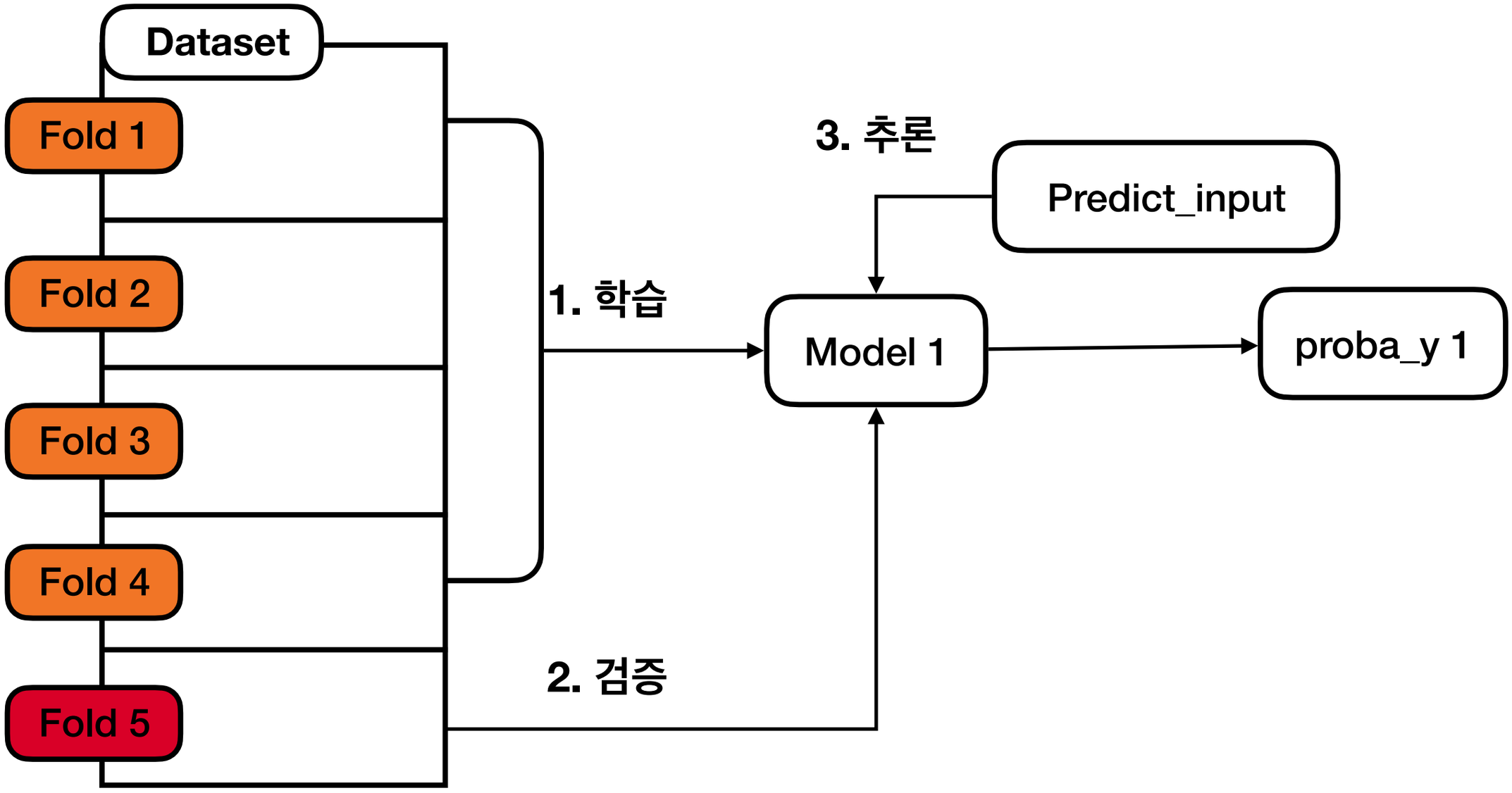
02 k-Fold Cross Validation



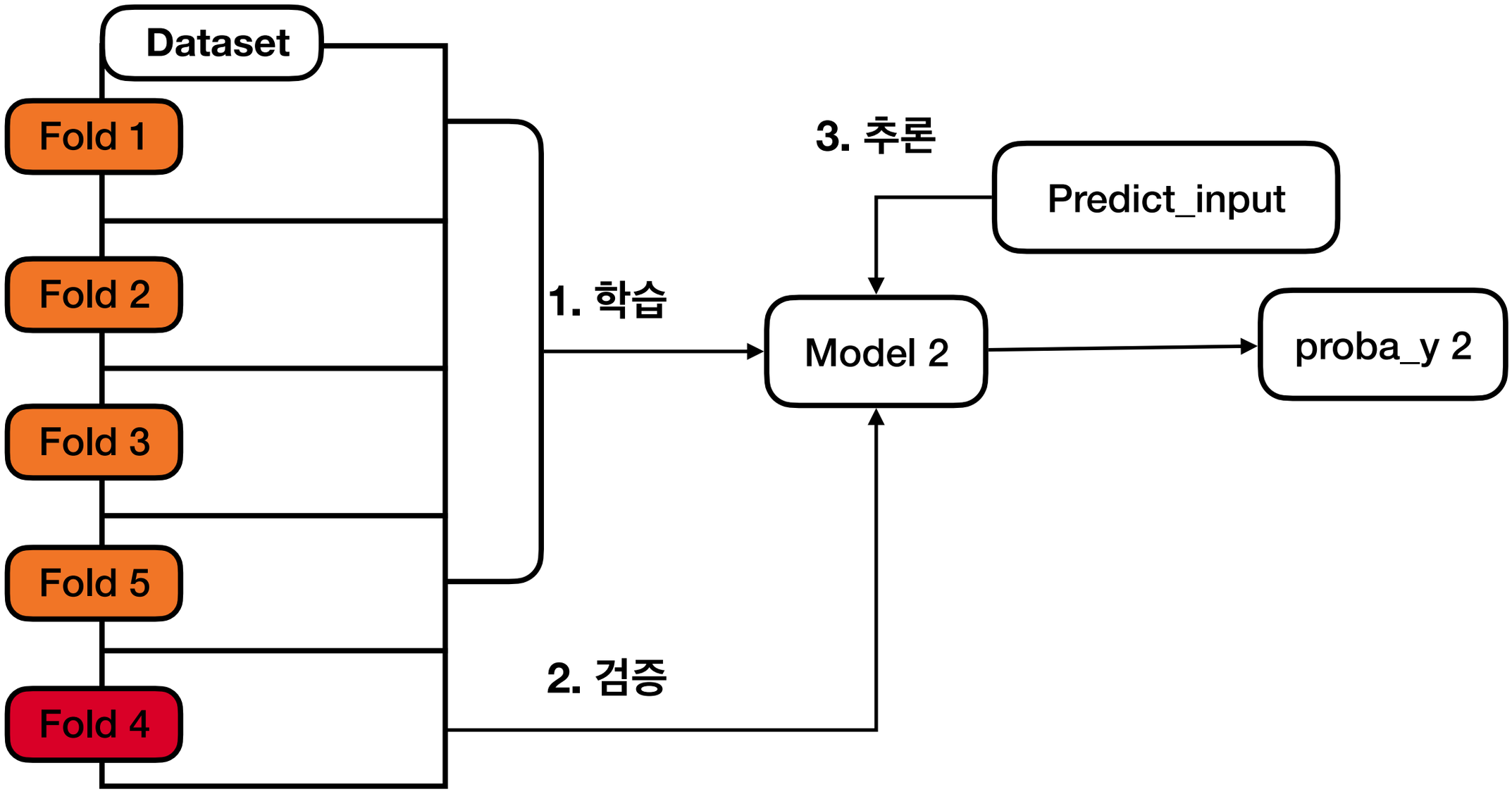
Q&A

3. Out-of-Fold 앙상블 & Stacking

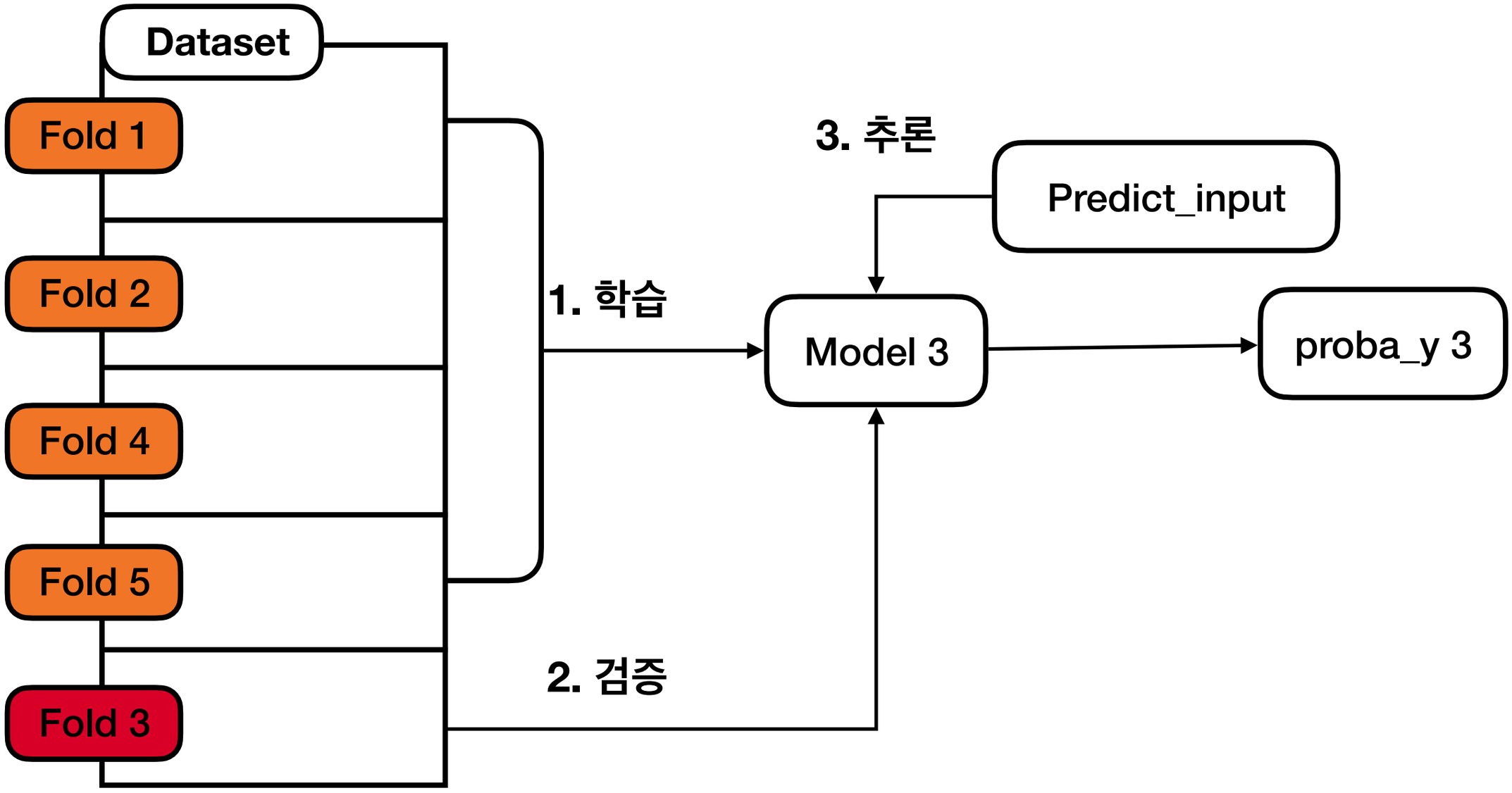
03 Out-of-Fold 앙상블



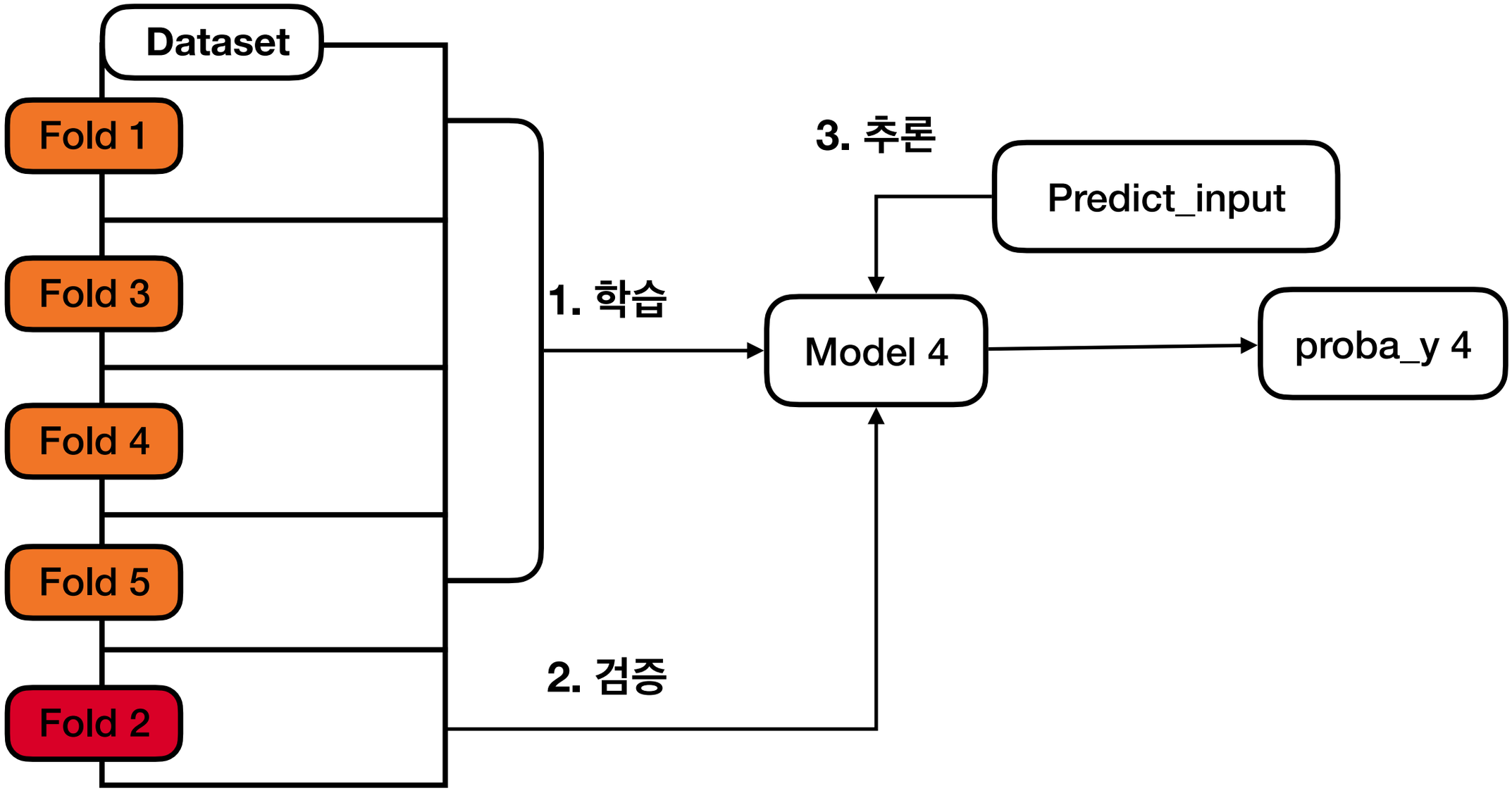
03 Out-of-Fold 앙상블



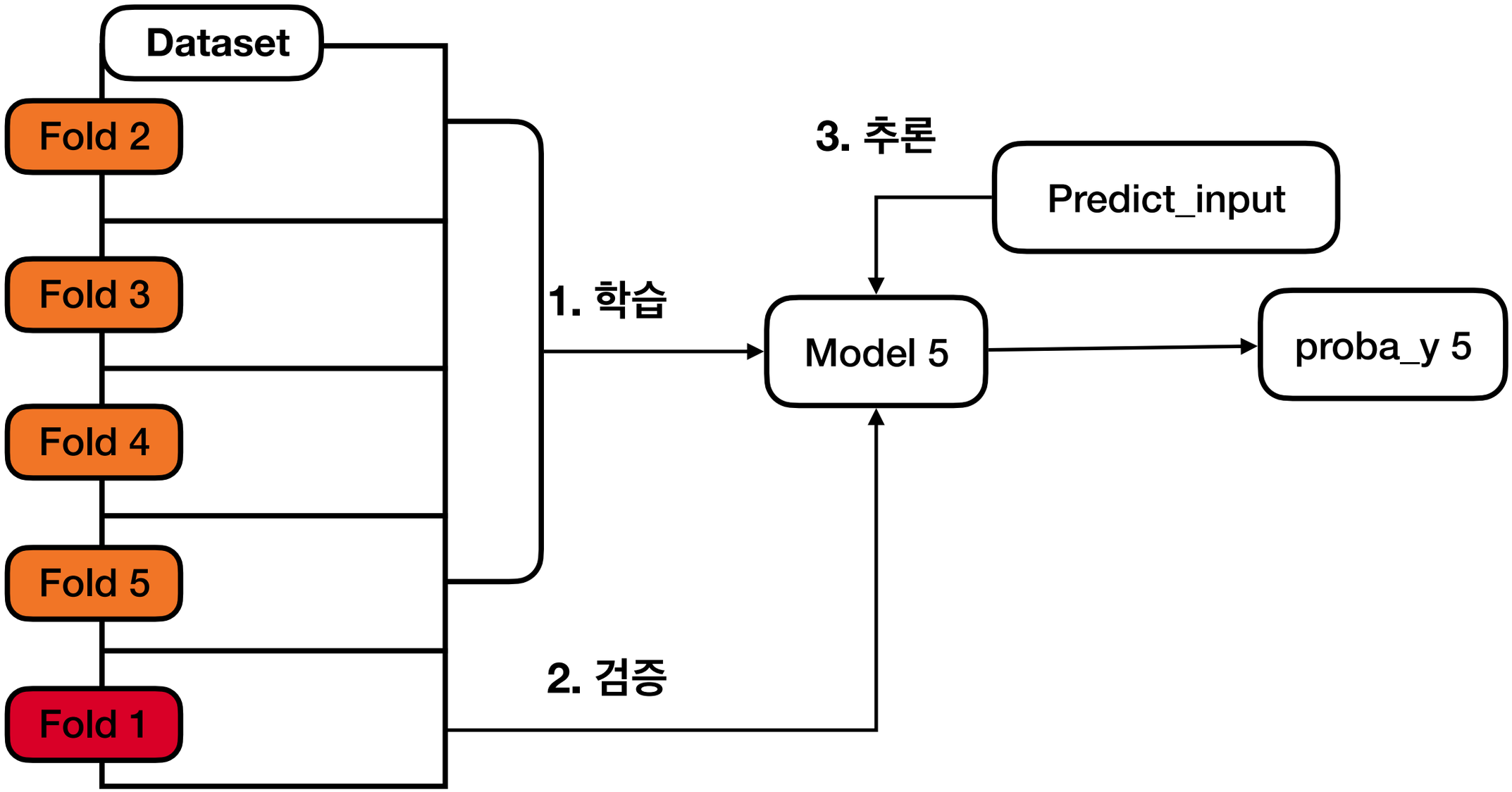
03 Out-of-Fold 앙상블



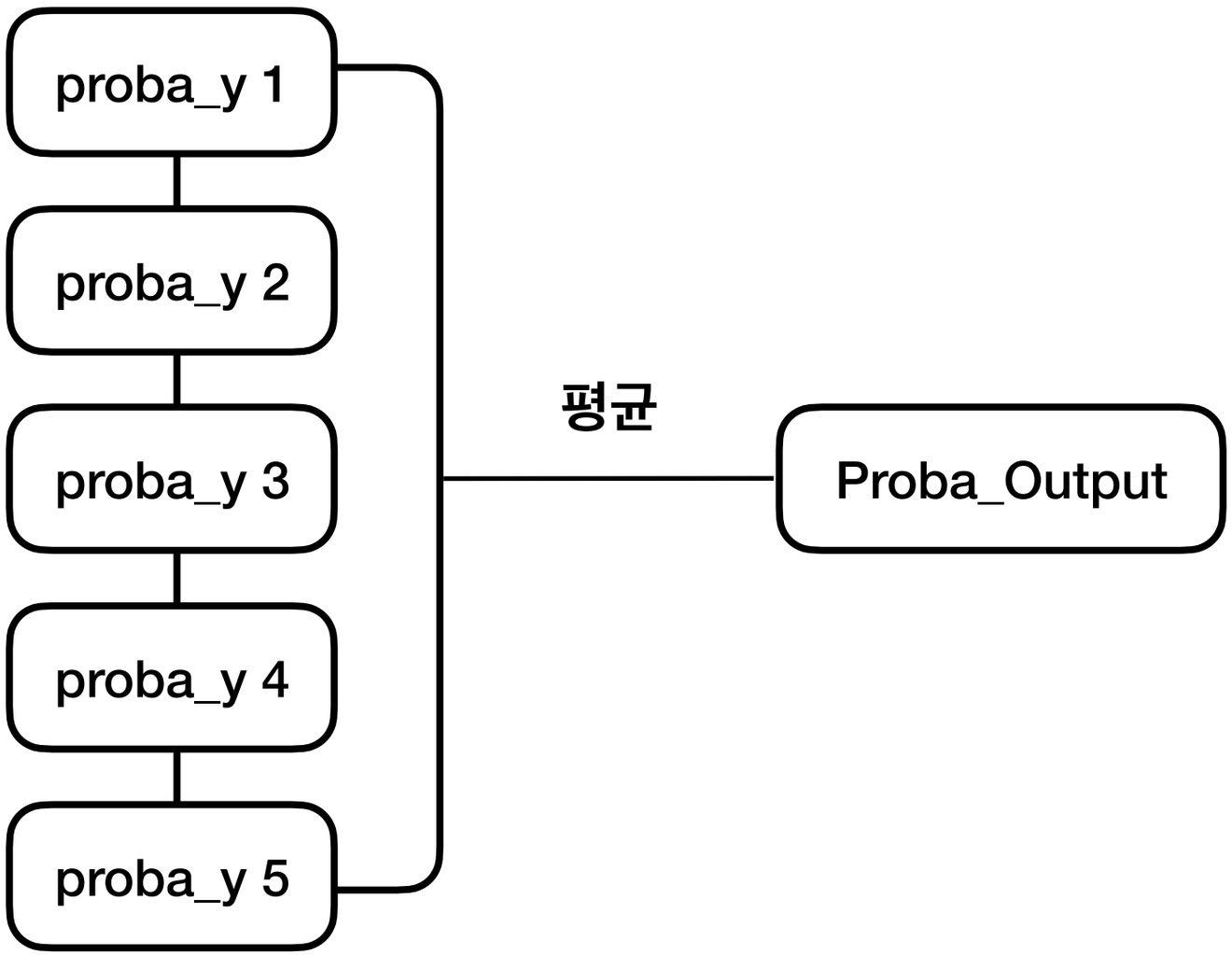
03 Out-of-Fold 앙상블



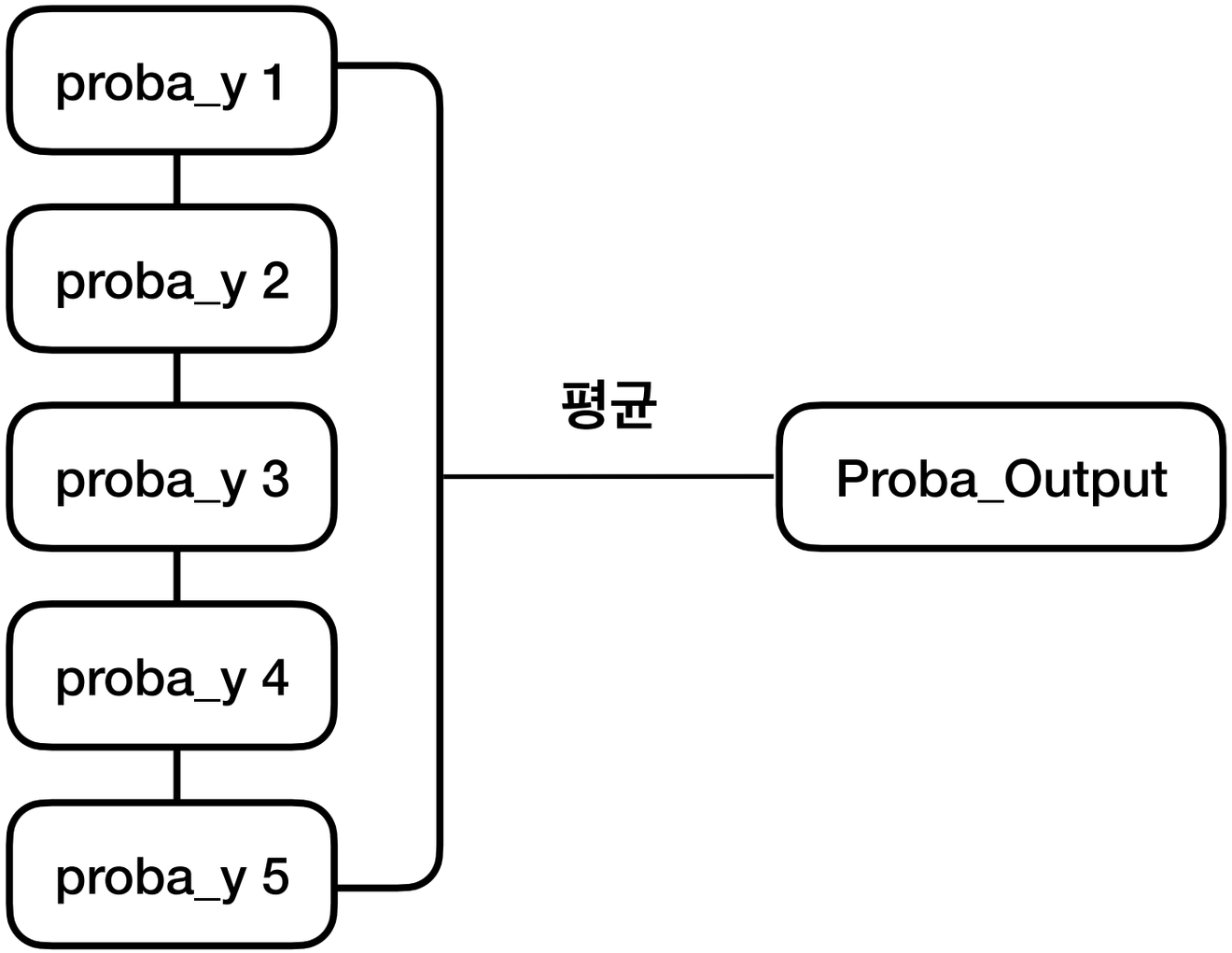
03 Out-of-Fold 앙상블



03 Out-of-Fold 앙상블

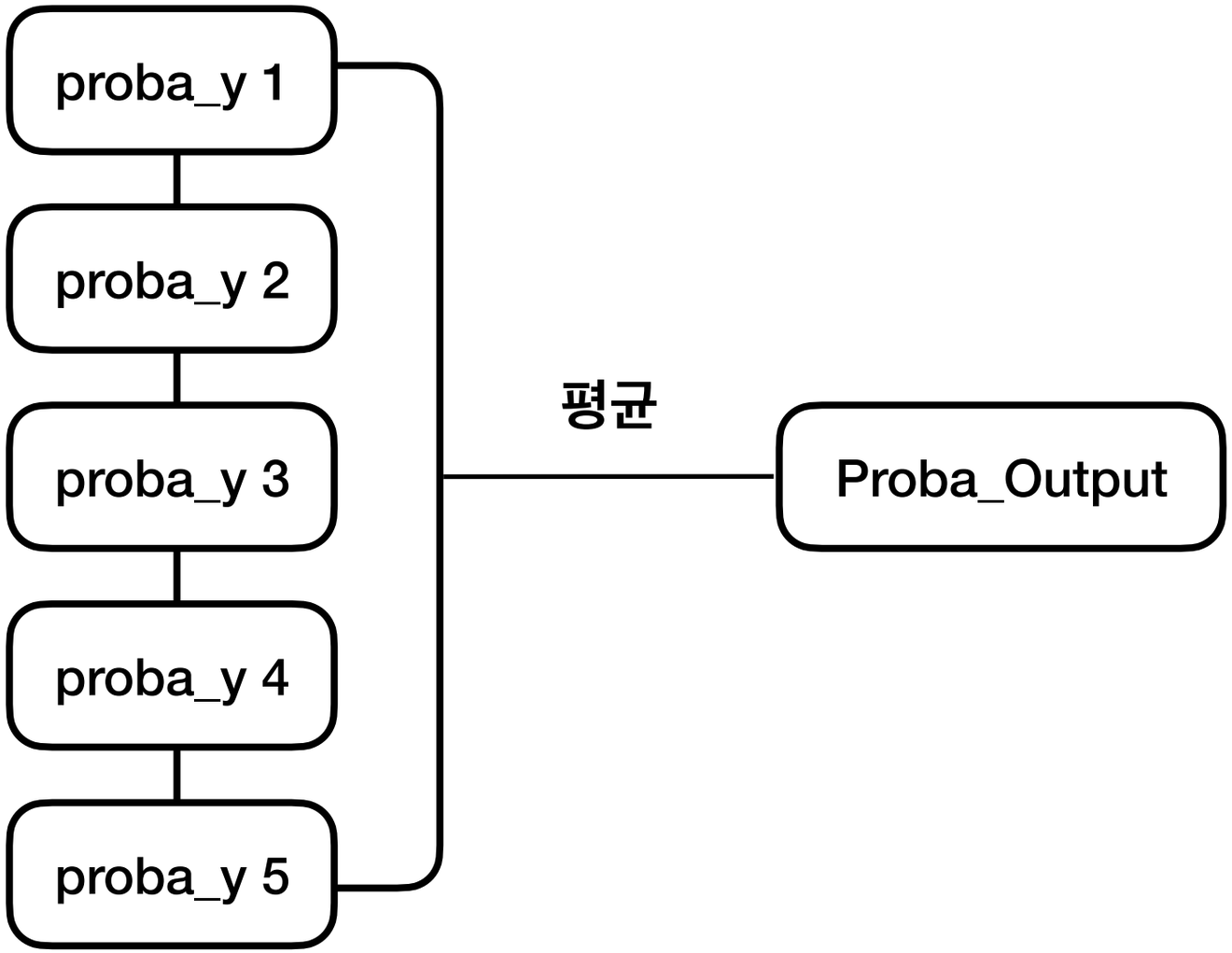


03 Stacking 앙상블



추가적으로 모델을 구성하지 않고 기존 모델의 성능을 극대화 할 수 없을까?

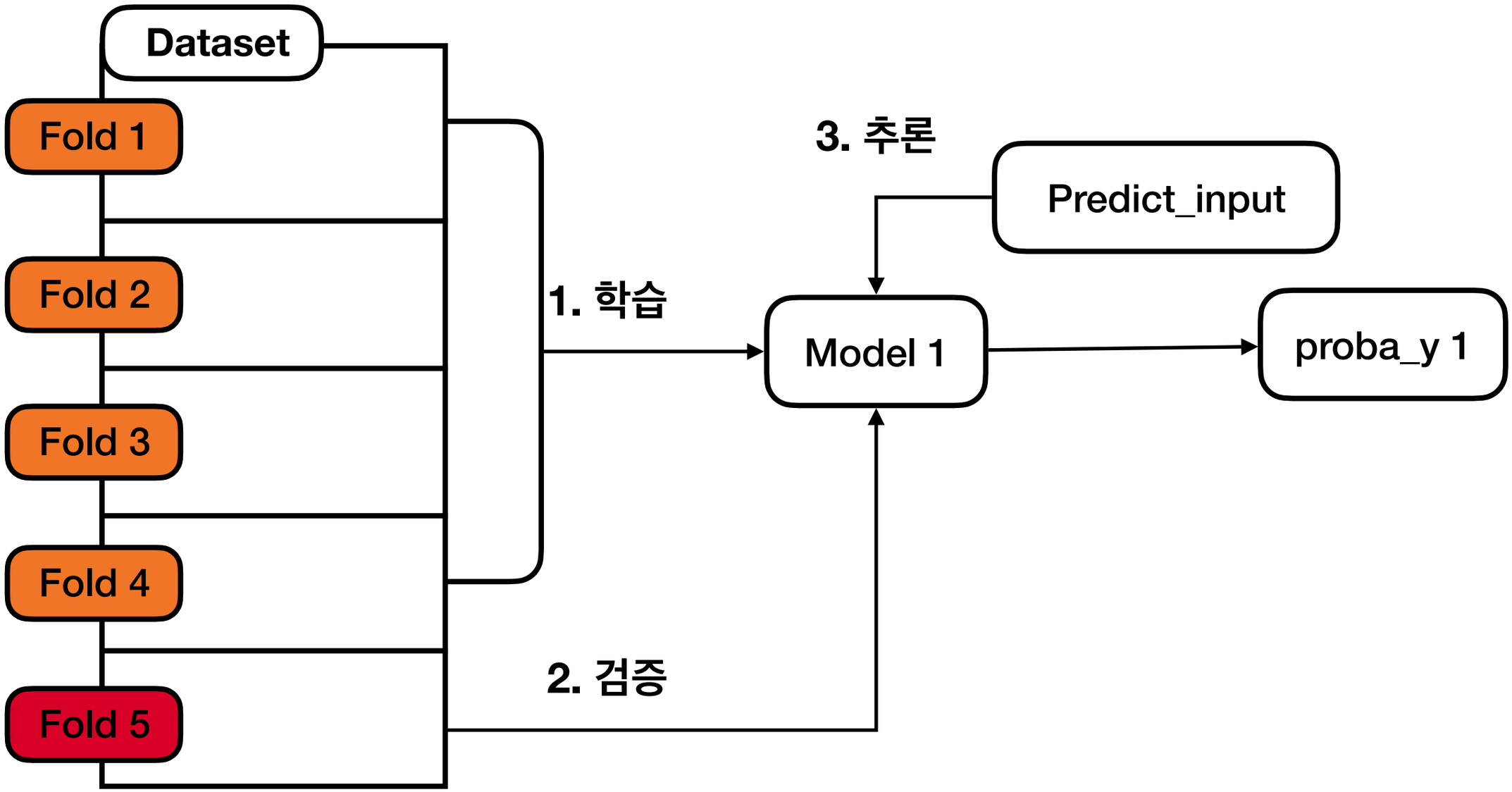
03 Stacking 앙상블



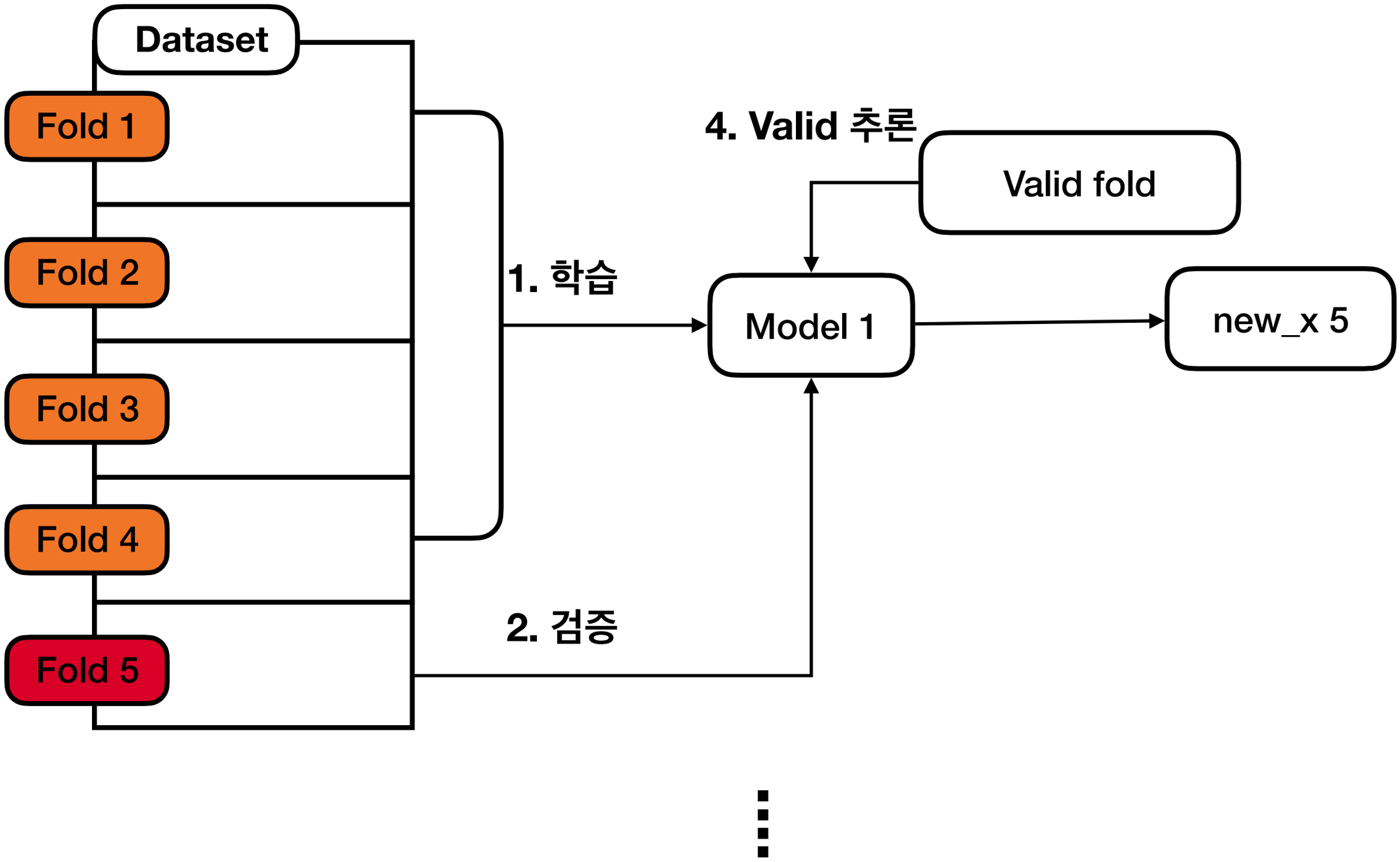
추가적으로 모델을 구성하지 않고 기존 모델의 성능을 극대화 할 수 없을까?

2-Level Stacking

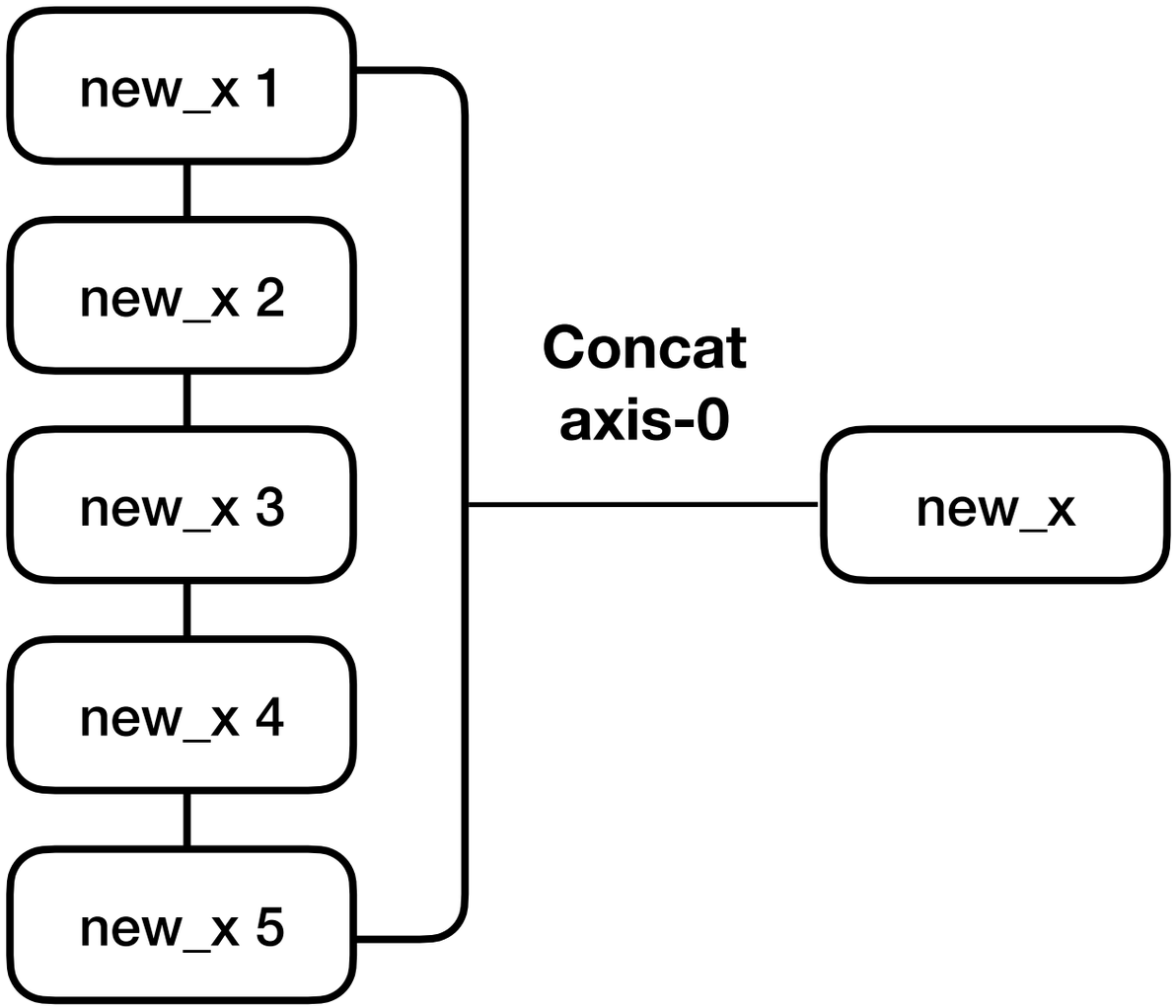
03 Stacking 앙상블



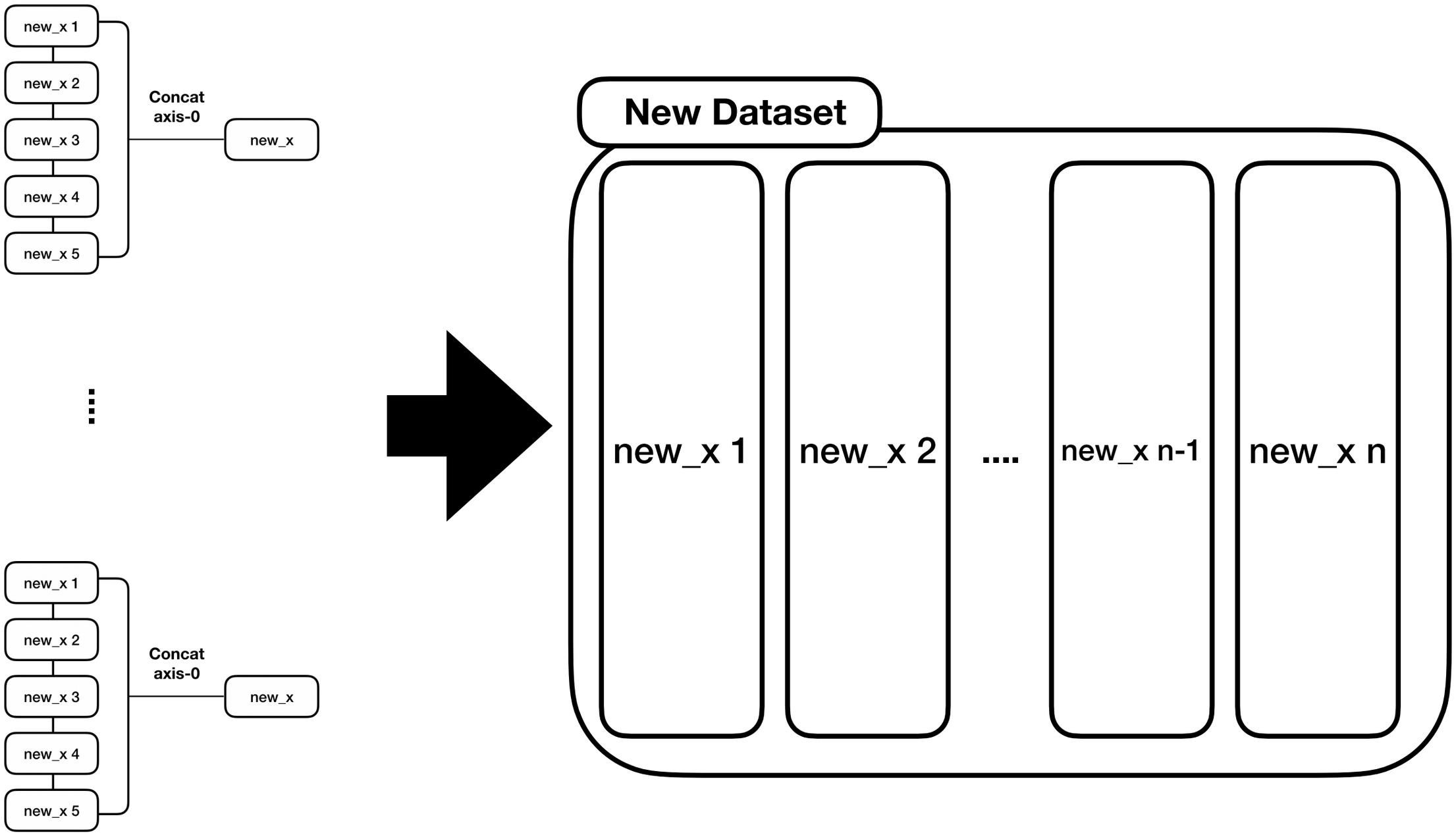
03 Stacking 앙상블



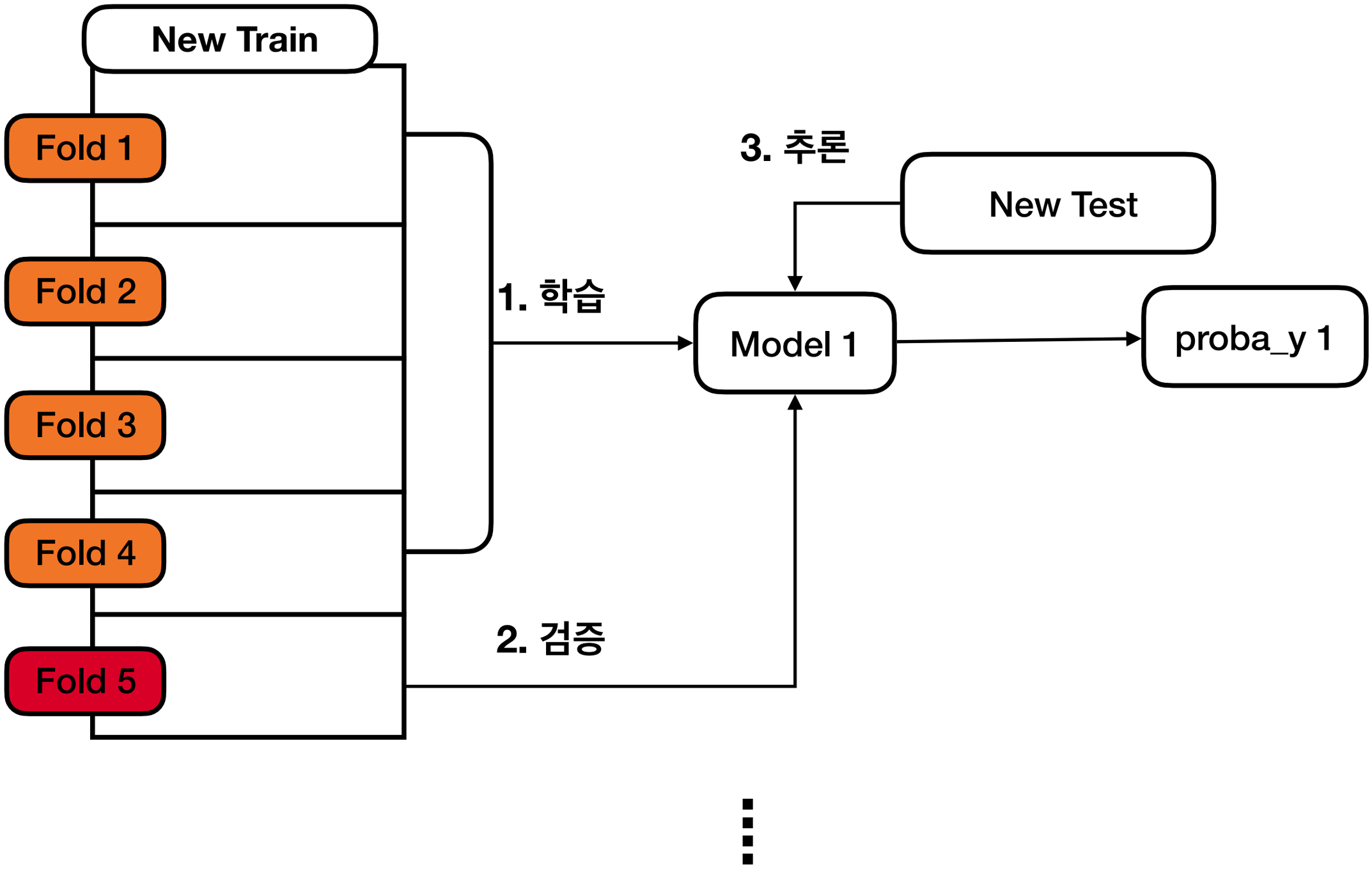
03 Stacking 앙상블



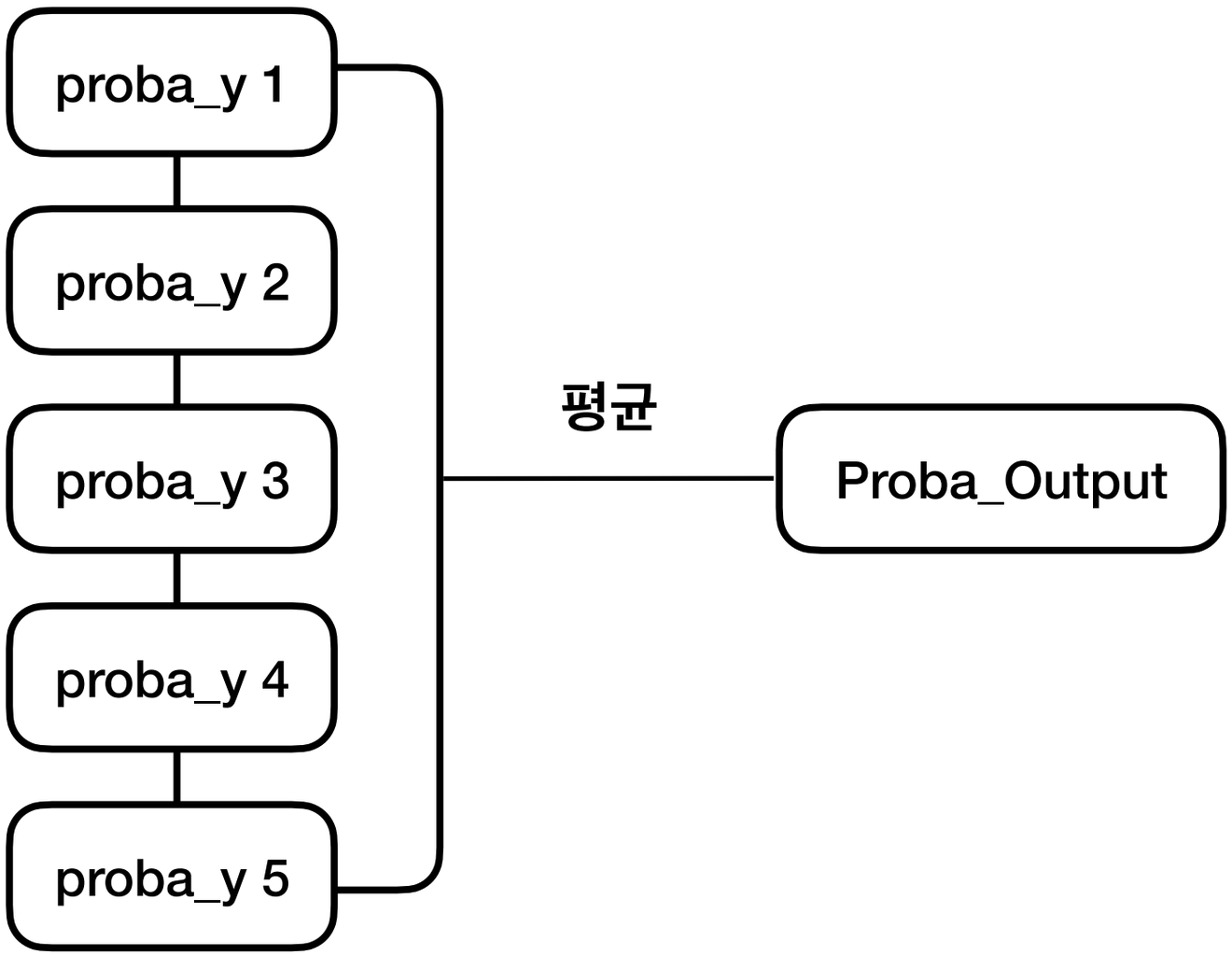
03 Stacking 앙상블



03 Stacking 앙상블



03 Stacking 앙상블



Q&A