

# 4. 모델 학습

**1** 모델 선택

## 모델 선택이란?

1.

모델 선택

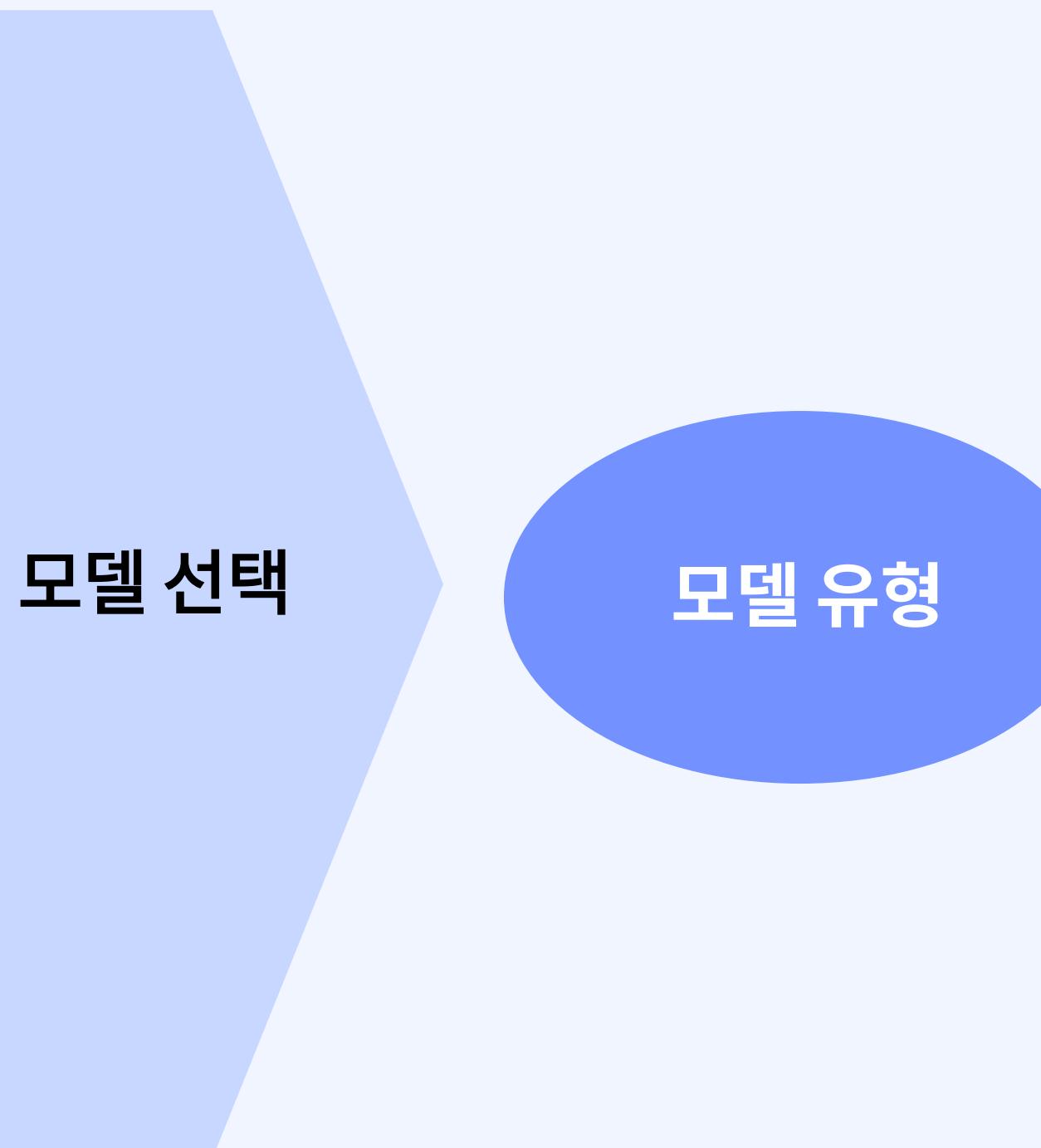
모델 선택이란 데이터 크기, 과제 유형, 특징 유형 등을 봤을 때, 가장 좋은 성능을 낼 것이라 예상되는 모델의 유형<sup>1)</sup>을 고르는 문제입니다.

샘플 개수

과제 유형

특징 유형

차원 크기



문제가 어려운 이유

- 선택할 수 있는 모델 유형이 매우 많음
- 어느 모델이 가장 적합할지 분석적으로 판단할 수 없음
- 예외가 매우 많아, 이러한 종류의 데이터에 대해서는 이러한 유형의 모델을 사용해야 한다고 단정할 수 없음

<sup>1)</sup> 같은 종류의 모델(예: 신경망)이더라도 하이퍼 파라미터나 학습에 사용한 데이터가 다르면 다른 모델입니다. 따라서 명확한 구별을 위해 모델 유형이라는 표현을 사용했습니다.

# 1.

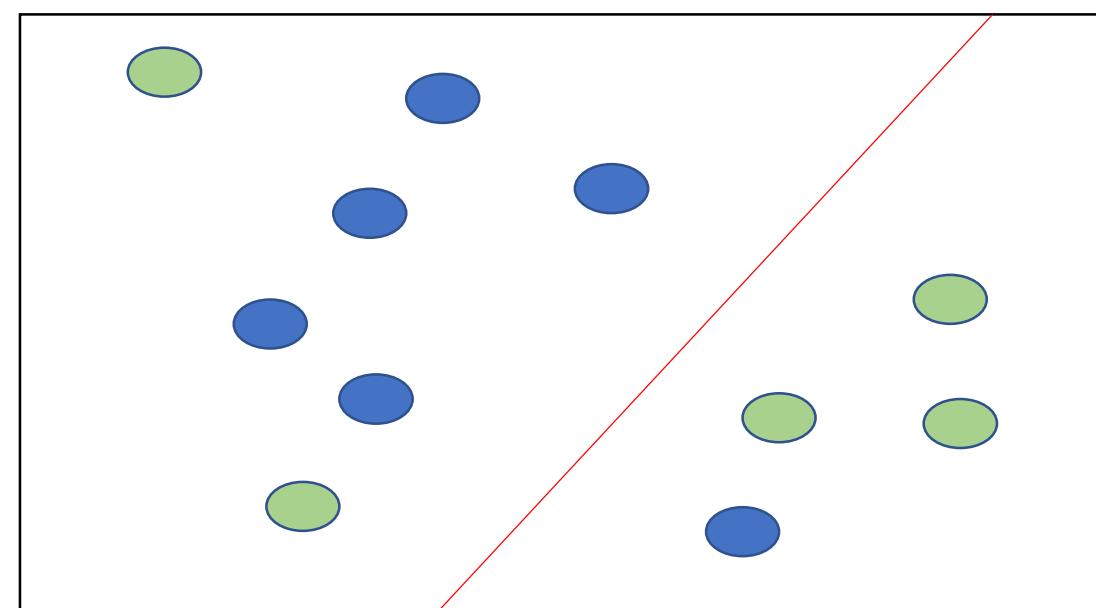
## 모델 선택

## 모델 복잡도

모델 복잡도(model complexity)란 모델이 데이터 공간을 얼마나 복잡하게 설명하는지를 측정한 값입니다.

단순

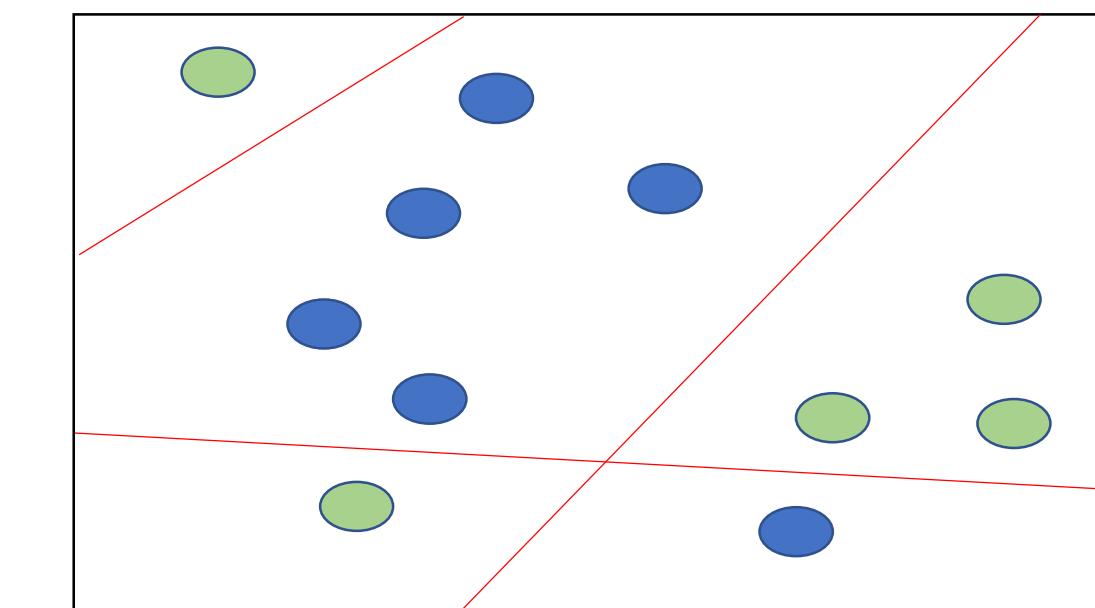
분류



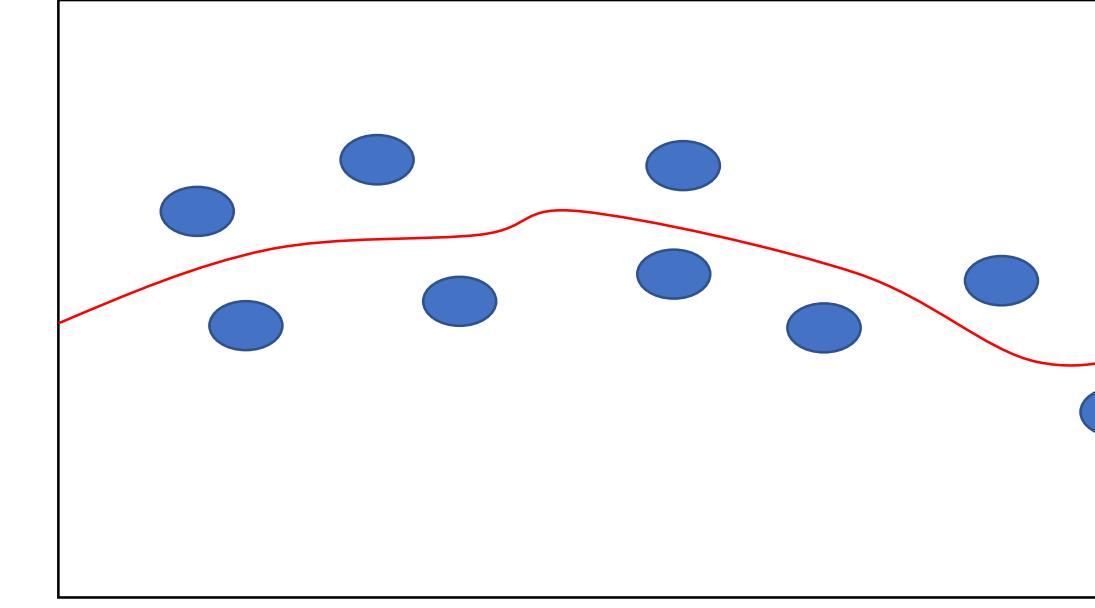
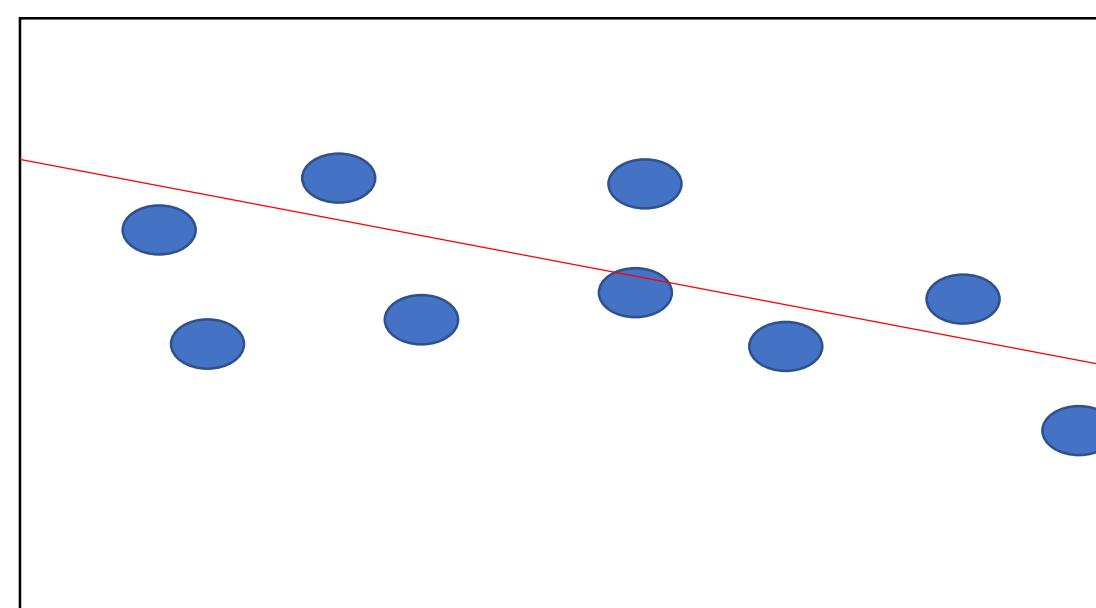
복잡

- 데이터에 비해 모델이 복잡할수록 과적합 가능성이 크고 단순할수록 과소 적합 가능성이 큼
- 따라서 데이터의 크기 및 밀도에 적합한 복잡도를 갖는 모델을 선택해야 함

복잡



회귀



## 모델 유형에 따른 복잡도

복잡도	분류	회귀
매우 단순	<ul style="list-style-type: none"> <li>깊이가 얕은 결정 나무</li> <li>나이브 베이즈</li> </ul>	<ul style="list-style-type: none"> <li>깊이가 얕은 결정 나무</li> <li>라쏘(Lasso)</li> <li>릿지(Ridge)</li> <li>엘라스틱넷(ElasticNet)</li> </ul>
단순	<ul style="list-style-type: none"> <li>로지스틱 회귀 모델</li> <li>은닉층이 한 개이고 노드가 적은 신경망</li> <li>깊이가 깊은 결정 나무</li> </ul>	<ul style="list-style-type: none"> <li>선형 회귀 모델</li> <li>은닉층이 한 개이고 노드가 적은 신경망</li> <li>깊이가 깊은 결정 나무</li> </ul>
복잡	<ul style="list-style-type: none"> <li>서포트 벡터 머신 (다항 커널 X)</li> <li>k-최근접 이웃</li> </ul>	<ul style="list-style-type: none"> <li>k-최근접 이웃</li> </ul>
매우 복잡	<ul style="list-style-type: none"> <li>다항 커널 서포트 벡터 머신</li> <li>은닉층이 여러 개인 신경망</li> <li>트리 앙상블</li> </ul>	<ul style="list-style-type: none"> <li>서포트 벡터 회귀</li> <li>은닉층이 여러 개인 신경망</li> <li>트리 앙상블</li> </ul>

모델의 복잡도는 하이퍼 파라미터 등에 따라 크게 다를 수 있으므로 참고용으로만 보기로 권장함

## 데이터 크기에 따른 적절한 모델 복잡도

### 1. 모델 선택

		데이터 공간 크기	
		좁음	넓음
샘플 개수	적음	단순	매우 단순
	많음	단순 혹은 복잡	매우 복잡

- 데이터 공간의 크기는 특징 개수와 특징의 상태 공간의 크기에 비례함. 즉, 연속형 특징이 많을수록 데이터 공간이 큼
- 데이터 공간이 클수록 복잡한 모델이 필요함
- 샘플이 적을 때 복잡한 모델을 학습하면 과적합 될 우려가 있음
- 데이터 공간 크기가 넓고 샘플 개수가 적을 때 매우 단순한 모델을 쓰는 이유는 데이터 공간이 크고 샘플 개수가 적을 때 복잡한 모델을 학습하면 과적합 가능성이 매우 크기 때문

샘플 개수와 데이터 공간의 크기를 고려하여 적절한 복잡도의 모델을 선택해야 함

## 설명력에 따른 모델 선택

1.

모델 선택

모델의 예측력뿐만 아니라 높은 설명력이 필요한 과제에서는 설명력을 고려한 모델 선택이 필요합니다.

설명력(explanatory power)이란 모델의 학습 및 예측 과정을 얼마나 잘 설명할 수 있는지를 나타내는 척도임

상황에 따라 모델의 예측력뿐만 아니라 높은 설명력이 필요하기도 함

(예시 1) 대출 심사 시스템과 같이 사람을 대상으로 하는 머신러닝 모델을 개발하는 과제

(예시 2) 알파고와 같이 복잡하지만 우수한 성능의 모델의 작동 과정을 이해하는 과제

최근 설명 가능한 인공지능이 급부상하면서 복잡한 딥러닝 모델을 해석하기 위한 시도가 다수 이뤄졌으나, 단순한 모델을 해석하는 것보다 여전히 훨씬 어려운 문제임

모델의 예측 과정을 사람이 이해할 수 있는 규칙 집합 형태로 설명할 수 있는 결정 나무가 설명력이 좋은 대표적인 모델임

# 4. 모델 학습

## 2 하이퍼 파라미터 튜닝

# 하이퍼 파라미터

하이퍼 파라미터란 사용자가 직접 설정하는 모델의 파라미터로 모델의 학습과 성능에 크게 영향을 끼치는 일종의 사용자 옵션이라고 볼 수 있습니다.

## 모델별 하이퍼 파라미터 예시

모델	하이퍼 파라미터
결정 나무	<ul style="list-style-type: none"> <li>최대 깊이</li> <li>분지 기준</li> <li>잎 노드 내 최소 샘플 수 등</li> </ul>
신경망	<ul style="list-style-type: none"> <li>은닉 층 구조</li> <li>최대 이터레이션 수</li> <li>활성화 함수 등</li> </ul>
서포트 벡터 머신	<ul style="list-style-type: none"> <li>커널</li> <li>계수 패널티 등</li> </ul>
라쏘	<ul style="list-style-type: none"> <li>계수 패널티 등</li> </ul>
나이브 베이즈	<ul style="list-style-type: none"> <li>확률 분포</li> <li>사전 확률 등</li> </ul>

## 하이퍼 파라미터의 분류

구분	설명	예시
필수 하이퍼 파라미터 (mandatory hyper-parameter)	반드시 설정 해야 하는 하이퍼 파라미터	<p>서포트 벡터 머신의 커널 함수</p> <ul style="list-style-type: none"> <li>예 1: 다항 커널, <math>(\gamma(\mathbf{x}_1^T \mathbf{x}_2) + \theta)^d</math></li> <li>예 2: 가우시안 커널, <math>\exp(-\gamma \ \mathbf{x}_1 - \mathbf{x}_2\ ^2)</math></li> </ul>
조건부 하이퍼 파라미터 (conditional hyper-parameter)	필수 하이퍼 파라미터에 따라 설정하지 않아도 되는 하이퍼 파라미터	다항 커널의 절편 ( $\theta$ )

## 하이퍼 파라미터 튜닝

하이퍼 파라미터 튜닝이란 하이퍼 파라미터를 모델의 성능을 기준으로 비교하여 최적의 하이퍼 파라미터를 선택하는 문제입니다.



여러 개의 하이퍼 파라미터를 직접 평가하고 비교해서 최적의 하이퍼 파라미터를 선택해야 함

## 2. 하이퍼 파라미터 튜닝

# 하이퍼 파라미터 튜닝 : 탐색 범위 설정

하이퍼 파라미터가 서로 상충하기도 하며, 어떤 하이퍼 파라미터는 모델 성능에 많은 영향을 끼치지 않습니다.

## 상충하는 파라미터

### 결정 나무의 최대 깊이와 잎 노드 내 최소 샘플 수:

결정 나무의 최대 깊이를 크게 설정하더라도 잎 노드 내 최소 샘플 수가 크면 더 이상 분지를 하지 못하기에 얇은 나무가 학습됨

## 성능에 많은 영향을 끼치지 않는 파라미터

### 결정 나무의 분지 기준:

분지 기준이 지니 계수이든 엔트로피이든 성능 차이가 크지 않은 것으로 알려져 있음

## 2. 하이퍼 파라미터 튜닝

# 하이퍼 파라미터 튜닝 : 복잡도 하이퍼 파라미터

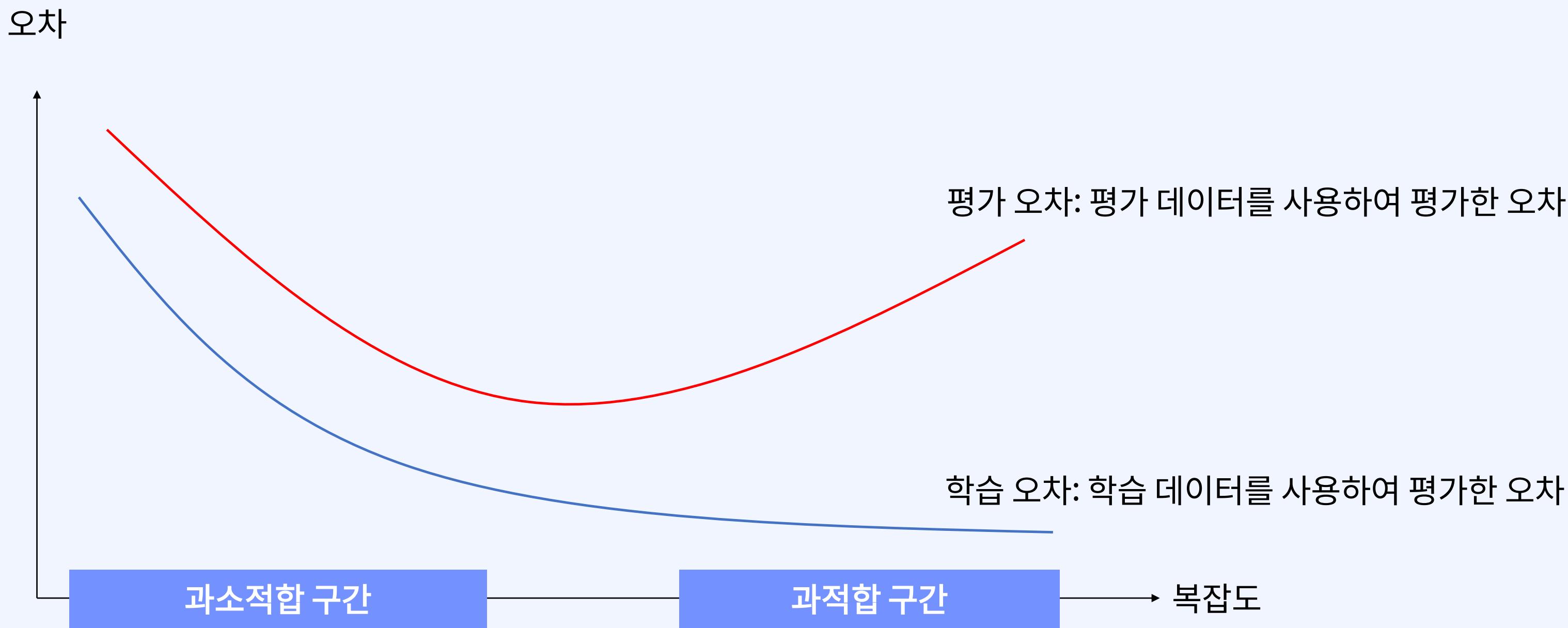
복잡도 하이퍼 파라미터란 모델의 복잡도를 결정짓는 하이퍼 파라미터를 말합니다.

모델별 복잡도 하이퍼 파라미터 예시

모델	하이퍼 파라미터
결정 나무	최대 깊이
신경망	<ul style="list-style-type: none"><li>• 은닉 층 구조</li><li>• 최대 이터레이션 수</li></ul>
서포트 벡터 머신	계수 패널티

## 복잡도에 따른 오차 분포

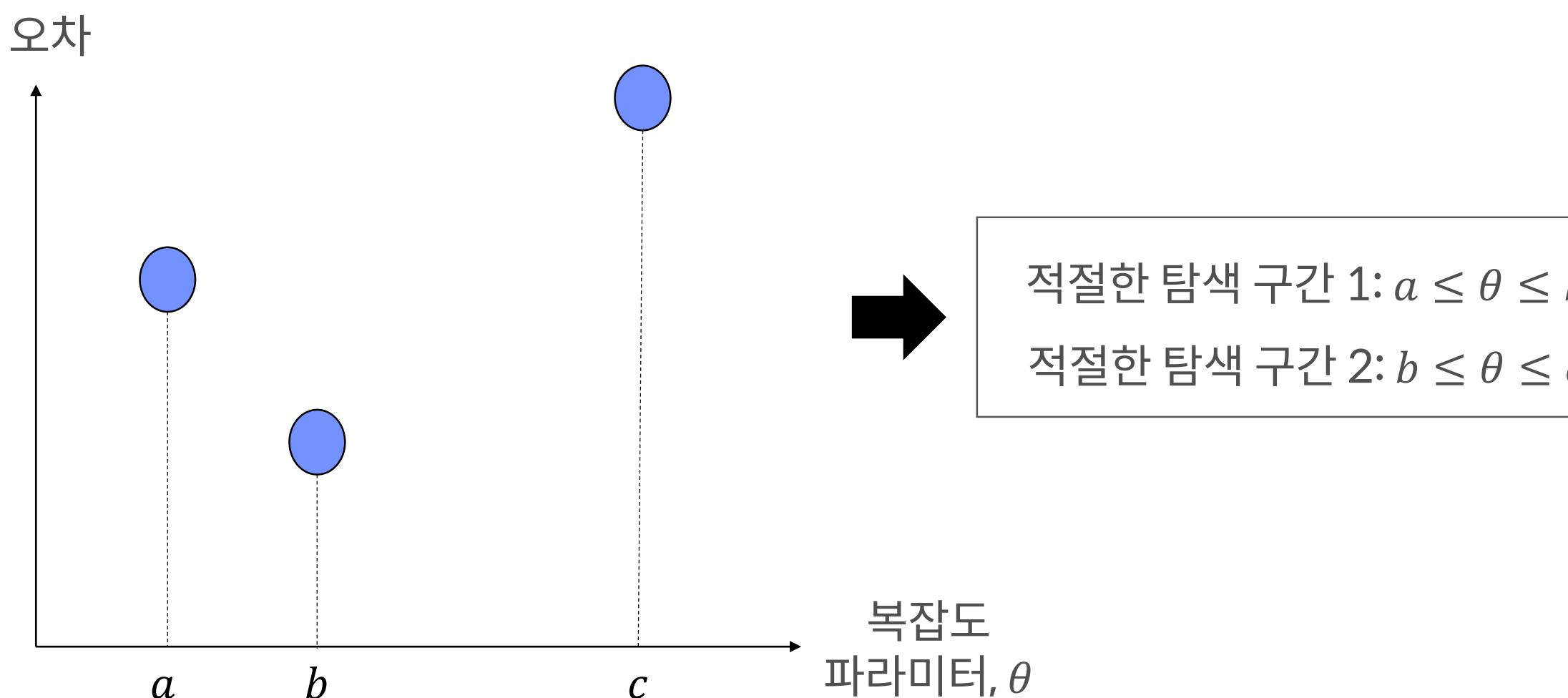
복잡도가 커질수록 학습 오차는 계속해서 감소하고 평가 오차는 감소했다가 증가합니다.



복잡도에 따른 오차 분포는 우연성을 최대한 배제하고 복잡도 하이퍼 파라미터를 튜닝하는데 응용할 수 있음

## 복잡도 하이퍼 파라미터 튜닝

복잡도에 따른 일반적인 오차 분포를 고려하면 복잡도 하이퍼 파라미터를 튜닝할 수 있습니다.



- 우연이 개입하지 않는다는 전제하에,  $\theta$ 에 따른 오차는 아래로 볼록한 형태임
- $a$ 보다는 크고  $c$ 보다는 작은 구간에서 가장 작은 오차를 갖는 하이퍼 파라미터가 있으므로, 이 구간에 있는 값으로 설정해야 함
- 혹은  $a$ 보다는 크고  $c$ 보다는 작은 구간에 속하는 여러 개의 하이퍼 파라미터를 비교하여, 탐색 범위를 더 좁히거나 혹은 가장 작은 오차를 갖는 하이퍼 파라미터를 찾아 사용해야 함

복잡도 하이퍼 파라미터 튜닝은 몇 개의 값을 평가하여 탐색 구간을 좁혀 나가는 방식으로 수행할 수 있음

## 하이퍼 파라미터 튜닝 : 자동화의 핵심

하이퍼 파라미터 튜닝은 모델 선택과 함께 머신러닝 자동화의 핵심이라고 할 수 있습니다.

상황에 따라 다르게 적용하는 데이터 탐색, 클래스 불균형 문제 해소, 특징 공학 등과 다르게, 하이퍼 파라미터 튜닝은 모든 지도 학습 모델 개발에서 비슷하게 적용할 수 있기 때문

다른 단계에 비해 경험이나 도메인 지식이 크게 필요하지 않아 자동화가 수월하기 때문

모델 성능에 매우 큰 영향을 끼치기 때문