Ch7. 머신러닝

목차 — Part 2

섹션	주제
A	의사결정 나무 & 랜덤 포레스트
B	부스팅 (AdaBoost, Gradient Boosting, XGBoost)
C	차원 축소 (PCA, t-SNE)
D	클러스터링 (K-Means, 계층적, DBSCAN, GMM)
E	신경망 기초 (퍼셉트론, 역전파, CNN/RNN)
F	End-to-End ML 시스템 설계 (10단계)
G	피처 엔지니어링 & 모델 선택 가이드

A. 의사결정 나무 & 랜덤 포레스트

의사결정 나무 — 기본 구조

- **CART**: Classification And Regression Trees — 분류/회귀 모두 가능
- 트리 형태의 모델: 루트 노드에서 이진 분할을 반복해 리프 노드에 도달
- **훈련 방식**: 탐욕적(greedy) + 재귀적(recursive)
  - 각 단계에서 오류를 최소화하는 피처와 분할점 선택
- 직관적이고 해석 가능 — "왜 이 예측을 했는가" 설명 용이
- **단점**: 과적합에 취약 (리프 노드가 각 관측치마다 생성 가능)

엔트로피와 정보 이득

- **엔트로피** $H(Y)$: 랜덤 변수 Y의 불확실성 측정

$$H(Y) = -\sum_{k} P(Y = k) \log P(Y = k)$$

- 베르누이 예: $p = 0.5$일 때 최대, $p = 0$ 또는 1일 때 최소
- **정보 이득** (Information Gain):

$$IG(Y, X) = H(Y) - H(Y \mid X)$$

- IG가 클수록 피처 X로 분할 시 불확실성 감소 폭이 큼
- 모든 피처를 평가 후 IG**를 최대화하는 피처** 선택 → 재귀 반복

분할 기준 비교 — Gini vs 엔트로피

기준	수식	특징
Gini Index	$G = 1 - \sum_k p_k^2$	실무에서 가장 많이 사용; 계산 빠름
Entropy	$H = -\sum_k p_k \log_2 p_k$	정보이론 기반; Gini와 유사 결과
분류 오류	$E = 1 - \max(p_k)$	가장 단순; 트리 성장에는 민감도 부족

- 실무 팁: Gini와 Entropy는 대부분 동일한 트리 생성 — **면접에서는 둘 다 설명 가능해야**
- **가지치기(Pruning)**: 과적합 방지를 위해 깊이 제한, 최소 샘플 수 설정

랜덤 포레스트 — 앙상블의 힘

- 다수의 의사결정 나무를 결합하여 **개별 트리의 과적합 문제 해결**
- 두 가지 핵심 메커니즘:

메커니즘	설명	효과
배깅 (Bagging)	부트스트랩 샘플로 각 트리 학습 → 예측 평균	분산(variance) 감소
피처 랜덤화	각 분할 시 피처의 랜덤 부분집합만 고려	트리 간 상관관계 감소

- **장점**: 해석 가능(피처 중요도), 빠른 학습(병렬 처리), 높은 예측 성능
- **면접 포인트**: "Random Forest vs Decision Tree 차이?" — 항상 나오는 질문

의사결정 나무 vs 랜덤 포레스트

의사결정 나무

- 단일 트리로 예측
- 해석 용이하지만 과적합 위험
- 학습 데이터에 민감
- 깊이 제한/가지치기 필수
- 빠른 학습, 낮은 계산 비용

랜덤 포레스트

- 다수 트리의 투표/평균
- 과적합에 강건
- 일반화 성능 우수
- 피처 중요도 자동 산출
- 학습 시간 길지만 병렬화 가능

B. 부스팅

부스팅 — 약한 학습기의 순차적 결합

- **핵심 아이디어**: 이전 모델이 틀린 데이터에 가중치를 높여 다음 모델이 집중
- 약한 학습기(weak learner) 여러 개를 **순차적**으로 결합
- 각 반복마다 잘못 예측된 데이터 포인트의 가중치 증가
- **주의**: 노이즈가 심한 데이터에서 과적합 가능성

> 랜덤 포레스트 = 병렬(배깅), 부스팅 = 순차

AdaBoost vs Gradient Boosting vs XGBoost

AdaBoost

- 단일 분할 트리(stump)를 순차 결합
- 데이터 포인트별 가중치 재조정
- 잘못 분류된 점에 높은 가중치
- 최종: 각 분류기의 가중합

Gradient Boosting

- AdaBoost의 일반화 형태
- **그래디언트**로 이전 모델의 약점 식별
- 모든 분류기가 동일 가중치
- 유연하지만 학습 느림

XGBoost

- Gradient Boosting의 최적화 버전
- 실행 속도 + 모델 성능 극대화
- 정규화(L1/L2) 내장
- 산업계에서 가장 널리 사용

앙상블 방법 비교 — 배깅 vs 부스팅

구분	배깅 (RF)	부스팅 (XGBoost)
학습 방식	병렬 (독립적)	순차 (의존적)
주 효과	분산 감소	편향 감소
과적합 위험	낮음	상대적 높음
노이즈 데이터	강건	민감
학습 속도	빠름 (병렬)	느림 (순차)
대표 알고리즘	Random Forest	XGBoost, LightGBM

> **면접 단골 질문**: "XGBoost와 Random Forest의 차이는?"

C. 차원 축소

차원의 저주 (Curse of Dimensionality)

- 피처 수가 매우 많고 데이터가 희소한 상황
- 고차원 공간에서 데이터 포인트 간 거리가 의미 없어짐
- ML 알고리즘이 패턴을 찾기 어려움 — 유사성/근접성 개념 붕괴
- **대응 전략**:
  - 데이터 증가 (비용/비현실적일 수 있음)
  - 피처 선택 (다중공선성 제거 등)
  - **차원 축소**: 정보 손실 최소화하며 피처 수를 줄임

PCA — 주성분 분석

- 가장 널리 사용되는 차원 축소 기법
- **핵심**: 상관된 변수들을 결합하여 **분산을 최대화하는 새 축(주성분)** 생성

$$y_j = \mathbf{w}_j^T \mathbf{X}$$

- 알고리즘 절차:
  i. 분산이 최대인 첫 번째 주성분 탐색
  ii. 첫 번째와 비상관이면서 분산이 두 번째로 큰 성분 탐색
  iii. k개 성분이 전체 분산의 대부분을 설명할 때까지 반복
- **수학적 본질**: 공분산 행렬의 **고유분해(eigendecomposition)**
  - 첫 번째 주성분 = 가장 큰 고유값에 대응하는 고유벡터

PCA 가정과 한계

항목	내용
핵심 가정	변수 간 선형 관계 존재
데이터 전처리	반드시 표준화(standardization) 필요 — 단위 민감
이상치	이상치에 취약 — 분산 왜곡
해석	주성분은 원래 변수의 선형 결합 → 직관적 해석 어려울 수 있음
성분 수 결정	설명 분산 비율 임계값 (예: 95%) 기준

- **비선형 대안**: t-SNE (t-distributed Stochastic Neighbor Embedding)
  - 비선형, 비결정적 방법
  - 주로 **시각화** 목적 (2D/3D 투영)
  - PCA와 달리 비선형 구조 포착 가능

PCA vs t-SNE 비교

PCA

- 선형 변환
- 전역(global) 구조 보존
- 결정적(deterministic)
- 차원 축소 + 시각화 모두 활용
- 빠른 계산 속도
- 대용량 데이터에 적합

t-SNE

- 비선형 변환
- 지역(local) 구조 보존에 강점
- 비결정적(매번 결과 다름)
- 주로 시각화 전용
- 계산 비용 높음
- 하이퍼파라미터(perplexity) 민감

D. 클러스터링

클러스터링 — 비지도 학습의 대표

- **비지도 학습**: 레이블 없이 데이터 내 구조적 패턴을 발견
- **목표**: 유사한 데이터 포인트를 그룹으로 묶기
- 활용 사례: 고객 세그먼테이션, 이상 탐지, 데이터 시각화
- **좋은 클러스터링의 기준**:
  - 클러스터 내 유사성 높음 (high intra-cluster similarity)
  - 클러스터 간 유사성 낮음 (low inter-cluster similarity)

K-Means 클러스터링 알고리즘

- **가장 널리 사용**: 구현 쉽고, 해석 직관적

알고리즘 절차:

1. k개 클러스터로 데이터 분할, 무작위 중심점(centroid) 선택
2. 각 데이터 포인트를 가장 가까운 중심점에 할당
3. 중심점을 해당 클러스터의 평균으로 업데이트
4. 수렴할 때까지 2-3 반복

손실 함수 (유클리드 거리 기반):

$$L = \sum_{j=1}^{k} \sum_{\mathbf{x} \in S_j} \|\mathbf{x} - \boldsymbol{\mu}_j\|^2$$

- **주의**: k는 사용자가 설정 — 엘보우 방법, 실루엣 점수 등으로 최적화

K-Means 대안 클러스터링 기법

계층적 클러스터링

- 각 점을 개별 클러스터로 시작
- 가장 가까운 클러스터를 병합
- 덴드로그램으로 시각화
- k 사전 지정 불필요
- 해석적이고 정보가 풍부

DBSCAN

- 밀도(density) 기반 클러스터링
- 클러스터 수를 자동 추론
- 임의 형태의 클러스터 탐지
- **이상치 탐지에 강점**
- 파라미터: eps, min_samples

GMM

- 데이터가 k개 가우시안 혼합이라 가정
- 평균 + 분산 모두 학습
- K-Means보다 유연 (타원형 클러스터)
- 확률적 할당 가능
- ML 엔지니어 면접에서 출제

클러스터링 알고리즘 선택 가이드

상황	추천 알고리즘	이유
k를 알고 있고 빠른 결과 필요	K-Means	단순, 빠름, 해석 용이
k 모르고 계층 관계 파악 필요	**계층적**	덴드로그램으로 구조 시각화
이상치 탐지가 주 목적	DBSCAN	노이즈 포인트 자동 분리
클러스터 형태가 비구형	DBSCAN/GMM	임의 형태 클러스터 탐지
확률적 소속 필요	GMM	소프트 할당(확률값) 제공
대용량 + 빠른 처리	K-Means	$O(nk)$ 복잡도

E. 신경망

퍼셉트론 — 신경망의 기본 단위

- **생물학적 뉴런 모방**: 입력 → 가중합 → 활성화 함수 → 출력
- 수식 표현:

$$z = \sigma\left(\sum_i w_i x_i + b\right)$$

- 입력 x_i에 가중치 w_i를 곱해 선형 결합
- **활성화 함수** σ가 비선형성 부여
- 퍼셉트론을 여러 층 결합 → **다층 퍼셉트론**(MLP) = 신경망

주요 활성화 함수 비교

함수	수식	용도
Sigmoid	$\sigma(z) = \frac{1}{1+e^{-z}}$	로지스틱 회귀, 이진 분류 출력
Tanh	$\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$	은닉층 (출력 범위 -1~1)
ReLU	$\max(0, z)$	현대 신경망 기본 활성화 함수
Softplus	$\ln(1 + e^{z})$	ReLU의 부드러운 버전
Step	$\phi(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases}$	초기 퍼셉트론

- **ReLU가 가장 많이 사용**: 기울기 소실 문제 완화, 계산 효율적

신경망 구조 — 은닉층과 깊이

- **Feed-forward 구조**: 입력층 → 은닉층(들) → 출력층
- **은닉층(Hidden Layers)**: 입력/출력이 아닌 중간 변환 층
  - 각 층이 특정 패턴을 학습 (예: 정지 신호, 신호등 등)
  - 층 수가 많을수록 "깊은" 학습 = **딥러닝**
- **하이퍼파라미터**:
  - 은닉층 수, 각 층의 뉴런 수
  - 활성화 함수 종류
  - 배치 크기, 학습률
  - 각 하이퍼파라미터가 학습 시간과 성능에 미치는 영향 이해 필수

역전파 (Backpropagation)

- 신경망의 **핵심 학습 알고리즘**
- 예측값과 실제값의 차이(손실)를 역방향으로 전파하여 가중치 업데이트

체인룰 적용:

$$\frac{\partial L(z,y)}{\partial w} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial w}$$

가중치 업데이트:

$$w \leftarrow w - \alpha \frac{\partial L(z,y)}{\partial w}$$

- **학습률** α: 너무 작으면 학습 정체, 너무 크면 수렴 실패
- **손실 함수**: 회귀 = MSE, 분류 = Cross-Entropy

기울기 소실과 폭발 문제

- **기울기 소실 (Vanishing Gradient)**:
  - 체인룰에서 작은 수를 반복 곱셈 → 초기 층의 기울기가 0에 수렴
  - Sigmoid, Tanh 등 기울기 범위(0~1)인 함수에서 특히 심각
  - 결과: 초기 층의 가중치가 거의 업데이트되지 않음
- **기울기 폭발 (Exploding Gradient)**:
  - 기울기가 기하급수적으로 증가 → 가중치 발산
- **해결책**:
  - ReLU 활성화 함수 사용
  - ResNet: 잔차 연결로 기울기 직접 전달
  - LSTM: 게이트 메커니즘으로 장기 의존성 학습

신경망 학습 최적화 기법

Momentum

- SGD의 노이즈 문제 해결
- 이전 기울기 방향을 "관성"으로 유지
- 학습의 일관된 방향과 속도

Dropout

- 매 학습 단계에서 일부 뉴런을 무작위 비활성화
- 다양한 아키텍처를 시뮬레이션
- 과적합 방지 정규화 기법

Batch Normalization

- 은닉층의 활성값을 배치 단위로 정규화
- 각 층이 독립적으로 학습 가능
- 학습 속도 향상 + 정규화 효과

Transfer Learning

- 사전 학습된 모델을 재활용
- 데이터 부족 시 특히 유용 (예: BERT, ImageNet)
- 새 도메인에 미세 조정(fine-tuning)

과적합 방지 전략 요약

기법	원리	적용 시점
데이터 추가	더 많은 학습 데이터 확보	데이터 접근 가능 시 가장 효과적
입력 표준화	피처별 평균 0, 분산 1	항상 (학습 가속 + 안정화)
Batch Norm	은닉층 활성값 정규화	심층 신경망에서 기본 적용
Dropout	뉴런 무작위 비활성화	과적합 징후 시
정규화 (L1/L2)	가중치 크기 제한	모델 복잡도 제어
Early Stopping	검증 손실 증가 시 학습 중단	학습 곡선 모니터링 시

CNN vs RNN — 구조와 용도

CNN (합성곱 신경망)

- **컴퓨터 비전**에 특화
- 필터로 **공간적 의존성** 포착
- 구조: 합성곱층 → 풀링층 → 완전연결층
  - 합성곱: 엣지, 색상, 기울기 추출
  - 풀링: 차원 축소, 위치 불변성
- 이미지 분류, 객체 탐지

RNN (순환 신경망)

- **순차 데이터** 처리에 특화
- 내부 상태(memory)로 시간적 의존성 학습
- 오디오, 비디오, 텍스트 등
- 임의 길이 입출력 가능
- **한계**: 장기 의존성 학습 어려움
- → LSTM이 실무 대체

LSTM과 강화학습 개요

LSTM (Long Short-Term Memory):

- RNN의 장기 의존성 한계를 극복
- **3개 게이트**: 입력(쓸지), 출력(얼마나 쓸지), 망각(얼마나 지울지)
- 정보 흐름을 세밀하게 제어 → 실무에서 RNN 대신 LSTM 사용

강화학습 (Reinforcement Learning):

- 지도/비지도 학습과 다른 패러다임
- 에이전트가 환경에서 행동 → 보상 최대화
- **4대 구성요소**: 보상 함수, 정책, 모델, 가치 함수
- 게임(AlphaGo), 로봇공학에 활용
- 일반 데이터 사이언스 면접에서는 드물게 출제

신경망 아키텍처 선택 가이드

데이터 유형	추천 아키텍처	핵심 특징
이미지	CNN	공간적 패턴 포착 (필터)
시계열/텍스트	RNN/LSTM	순차적 의존성 학습
표형 데이터	MLP (또는 트리 앙상블)	범용적, XGBoost와 비교
자연어 처리	Transformer/BERT	어텐션 메커니즘, 사전학습
이상치 탐지	Autoencoder	재구성 오류 기반 탐지
생성 작업	GAN/VAE	새로운 데이터 샘플 생성

F. End-to-End ML 시스템 설계

ML 면접의 본질

알고리즘 하나 아는 것이 아니라, 비즈니스 문제를 ML 시스템으로 풀어내는 전체 과정을 설계하는 능력

ML 워크플로우 10단계 개요

1. 문제 정의

- 비즈니스 목표 명확화
- 기술적 제약 조건 파악

2. 메트릭 설정

- 단일 평가 지표 선정
- 비즈니스 KPI와 연결

3. 데이터 이해

- 데이터 소스 파악
- 품질/편향 확인

4~10. 실행

- 탐색 → 정제 → 피처 → 모델 → 배포 → 반복

Step 1: 문제와 제약 조건 명확화

비즈니스 측면 질문:

- 모델링할 종속 변수는 무엇인가?
- 기존 접근법과 비교 기준(baseline)은?
- ML이 정말 필요한가? 규칙 기반으로 충분하지 않은가?
- 법적/윤리적 제약은? (예: 대출 심사에서 인종 변수 사용 불가)
- 잘못된 예측의 비즈니스 영향은?

기술 측면 질문:

- 지연시간(latency) 요구사항은?
- 처리량(throughput) 요구사항은?
- 모델 배포 환경은? (클라우드/엣지/온디바이스)

Step 2: 평가 메트릭 설정 전략

- **단일 메트릭 선호**: 팀 정렬과 모델 순위 비교에 유리
- 면접에서는 단일 메트릭 제시 후 추가 메트릭 언급으로 깊이 보여주기

전략	예시
F1 Score	Precision과 Recall의 조화 평균
Satisficing	Recall $\geq$ 0.95 제약 하에서 Precision 최적화
OEC	여러 서브 메트릭의 가중 합산
비즈니스 KPI 연결	모델 정확도 90% → 티켓 재라우팅 50% 감소 → 해결 시간 10% 단축

- **성공 기준**: 100% 정확도가 아닌 **현실적 임계값** 설정
  - 기존 시스템 대비 얼마나 개선해야 성공인가?

Step 3-4: 데이터 소스와 탐색

데이터 소스 확보 전략:

- 내부 데이터 + 크라우드소싱(Mechanical Turk)
- 사용자 온보딩 과정에서 수집
- 2nd/3rd party 데이터셋 구매
- 데이터 증강/합성 (엣지 케이스 보강)
- 시뮬레이션 (자율주행 등)

탐색적 데이터 분석(EDA):

- 각 컬럼의 유용성, 분산, 결측치, 이상값 파악
- 히스토그램, 상관 행렬 시각화
- 요약 통계: 평균, 중앙값, 분위수
- **"그림 한 장이 천 마디보다 낫다"** — John Tukey

Step 5: 데이터 정제 핵심

- **현실**: 데이터 사이언티스트 업무 시간의 ~80%가 데이터 정제

문제	대응 방법
중복 데이터	행/열 중복 제거
부정확한 값	스키마와 불일치하는 값 수정, 타이포 패턴 탐지
결측치	평균/중앙값 대체, 모델 기반 대체, 행 삭제(최후 수단)
이상치	원인 파악 → 제거/절단/유지 결정
다변량 이상치	개별 변수는 정상이나 조합이 비정상 (예: 4세 + 150cm)

G. 피처 엔지니어링 & 모델 선택

피처 엔지니어링 — 수치형 vs 범주형

수치형 데이터

- **변환**: log, capping, flooring → 왜도 보정
- **비닝(Binning)**: 연속 변수를 구간으로 이산화
- **차원 축소**: PCA로 비상관 피처 생성
- **정규화**: Min-Max → [0, 1]
- **표준화**: Z-score → 평균 0, 분산 1
  - K-Means 등 거리 기반 알고리즘에 필수

범주

-
-
-

텍스트 데이터 전처리 기법

기법	설명
Stemming	어근 추출 ("liked" → "like") — 단순 문자 삭제
Lemmatization	문맥 고려 어근화 ("caring" → "care")
Filtering	불용어(stop words) 제거 + 구두점 제거
Bag-of-Words	단어와 빈도로 텍스트 표현
N-grams	N개 연속 단어를 하나의 단위로
Word Embeddings	단어를 벡터로 변환 — 의미적 유사성 보존 (Word2Vec, GloVe)

- NLP 프로젝트가 이력서에 있다면 면접에서 물어볼 확률 높음

모델 선택 시 고려 요소

1. **학습/예측 속도**: 선형 회귀 ≫ 신경망 (동일 데이터 기준)
2. **예산**: 신경망은 GPU 필요, 트리 모델은 CPU로 충분
3. **데이터 볼륨/차원**: 신경망은 대용량·고차원 처리, K-NN은 고차원에 취약
4. **피처 유형**: 선형 회귀는 원핫 필요, 트리는 범주형 직접 처리
5. **설명 가능성**: 규제 산업에서는 해석 가능한 모델 필수 (선형 회귀, 의사결정 나무)
6. **과적합 위험**: 데이터 적으면 단순 모델, 많으면 복잡 모델 고려

알고리즘 선택 치트시트 — 분류

조건	추천 모델
속도 + 설명 가능성 필요	Naive Bayes / 로지스틱 회귀
대규모 데이터 + 높은 정확도	Random Forest / XGBoost
비선형 경계 + 중간 규모	Kernel SVM
이미지/음성/텍스트	신경망 (CNN/RNN/Transformer)
해석 가능성 최우선	의사결정 나무 / 로지스틱 회귀
피처 수 >> 샘플 수	Naive Bayes / SVM

알고리즘 선택 치트시트 — 회귀

조건	추천 모델
선형 관계 + 해석 필요	선형 회귀
비선형 관계 + 속도	의사결정 나무
높은 정확도 최우선	Random Forest / Gradient Boosting
대규모/고차원 데이터	신경망
설명 가능성 + 정규화	Ridge/Lasso 회귀
피처 선택 자동화 필요	Lasso (L1) / Elastic Net

알고리즘 선택 치트시트 — 비지도 학습

조건	추천 모델
클러스터 수 알려진 경우	K-Means
클러스터 수 모름 + 이상치 존재	DBSCAN
계층적 구조 파악	Hierarchical Clustering
확률적 소속 필요	GMM
고차원 → 저차원 (선형)	PCA
시각화 목적 (비선형)	t-SNE / UMAP

Step 7-8: 모델 학습과 평가 전략

학습 프로세스:

- Train / Validation / Test 분할
- 교차 검증 (Cross-Validation)
- 하이퍼파라미터 튜닝

대처해야 할 이슈들:

이슈	대처 방법
데이터 불균형	SMOTE, 언더/오버 샘플링
편향 학습 데이터	적절한 평가 메트릭 선택
대규모 데이터	랜덤/층화 샘플링
과적합	정규화, 검증셋, 학습 곡선 모니터링

Step 9: 배포 전략 — Online vs Batch

Online 배포

- 실시간 예측 (저지연 필수)
- 캐싱 레이어로 피처 서빙
- 검색 자동완성, 추천 등
- 높은 인프라 비용
- 강건한 모니터링 필요

Batch 배포

- 주기적 예측 생성
- 즉시 결과 불필요 시 적합
- 대부분의 추천 시스템
- 새 데이터 반영 지연
- 비용 효율적

모델 열화와 반복 개선

모델 열화 (Model Degradation):

- 데이터 분포 변화 → Training-Serving Skew
- 예: 겨울에 학습한 추천 모델이 여름에도 패딩 추천
- Feature Drift: 시간에 따라 피처의 의미/분포 변화

대응 전략:

- 재학습 주기 설정 (이벤트 트리거 포함)
- 로깅으로 성능 모니터링
- 신규 데이터 vs 과거 데이터 비율 결정

Step 10: **반복**(Iterate):

- **오류 분석**: 잘못된 예측을 수동 분석 → 원인 유형별 분류
- 원인별 우선순위로 개선 프로젝트 수행

실무에서 가장 중요한 판단 프레임워크

판단 기준	질문
ML 필요성	규칙 기반으로 충분하지 않은가?
윤리/법률	이 데이터와 모델 사용이 합법적인가?
비용 대비 효과	모델 개발·유지 비용 vs 비즈니스 가치
실패 영향	잘못된 예측의 최악의 시나리오는?
설명 가능성	이해관계자에게 "왜"를 설명할 수 있는가?
반복 가능성	데이터와 환경 변화에 얼마나 빨리 적응하는가?

> 학생 상담 시 핵심: **"어떤 알고리즘을 쓸까"보다 "이 문제에 ML이 적합한가"가 먼저**

Part 2 핵심 요약

영역	기억할 것
트리 모델	Decision Tree → 과적합 → Random Forest(배깅) → 해결
부스팅	순차적 약학습기, XGBoost가 산업계 표준
차원 축소	PCA(선형, 전역), t-SNE(비선형, 시각화)
클러스터링	K-Means(기본), DBSCAN(이상치), GMM(확률적)
신경망	퍼셉트론 → MLP → CNN(이미지)/RNN(순차) → LSTM
ML 시스템 설계	10단계: 문제정의 → 메트릭 → 데이터 → … → 반복
피처 엔지니어링	수치형(표준화), 범주형(원핫), 텍스트(임베딩)
모델 선택	속도·해석성·정확도·데이터 크기의 트레이드오프