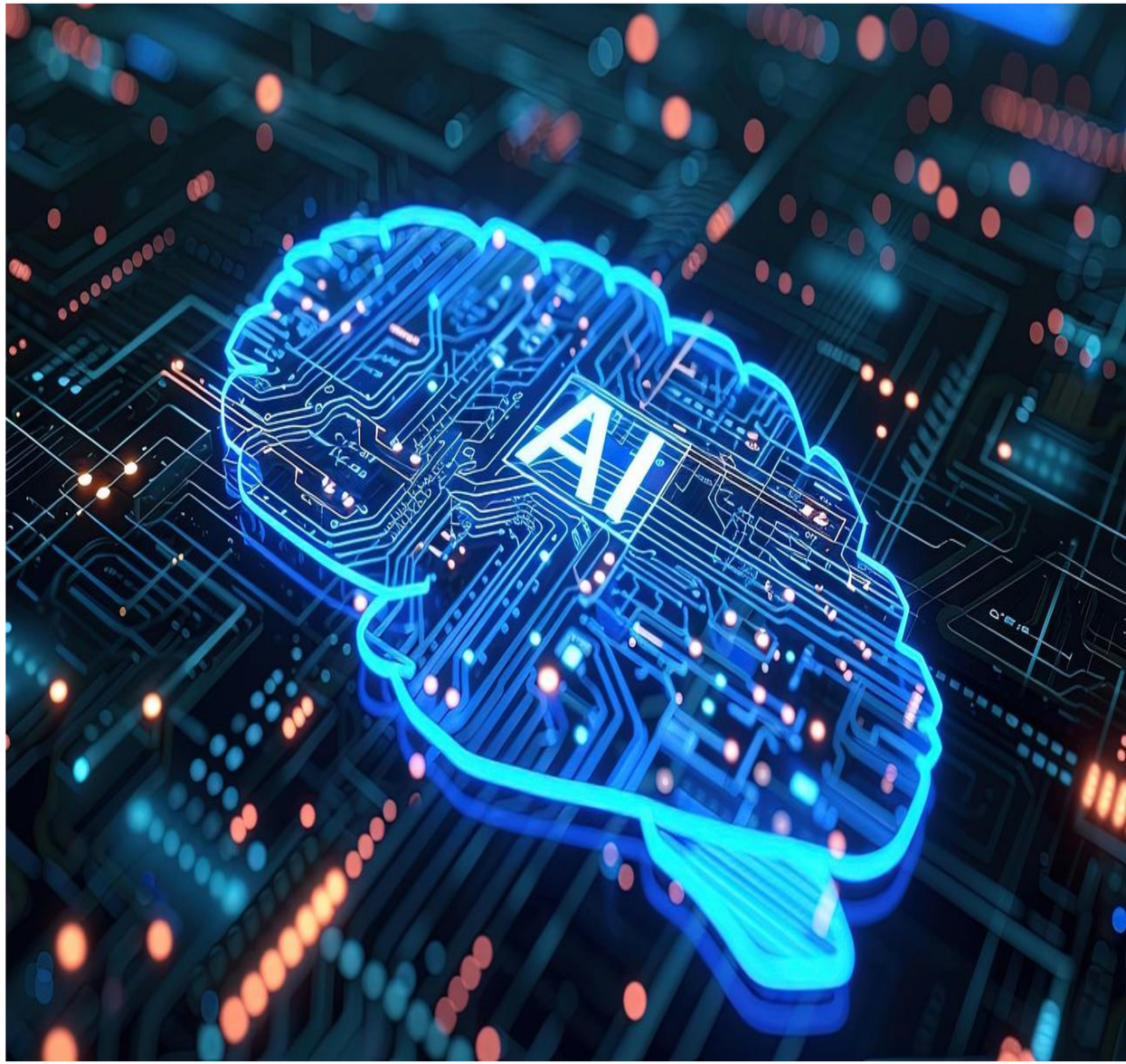
주요 머신러닝 (Machine Learning) 알고리즘 실습과 구현

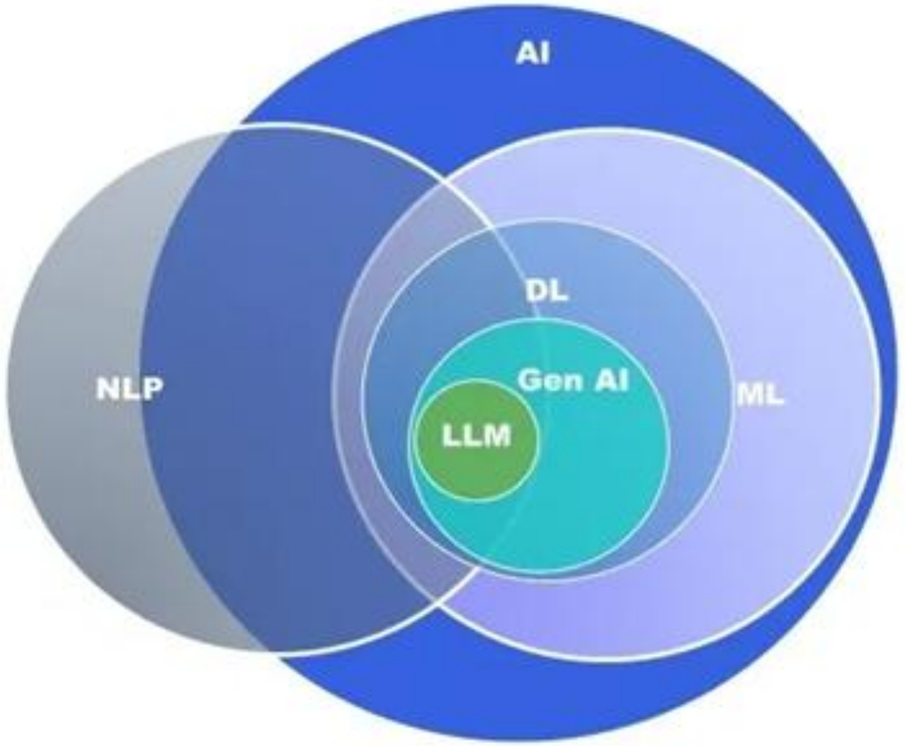


AID 30+ 집중캠프 교육과정

홍익대학교
이 한 표

CONTENTS

인공지능이란



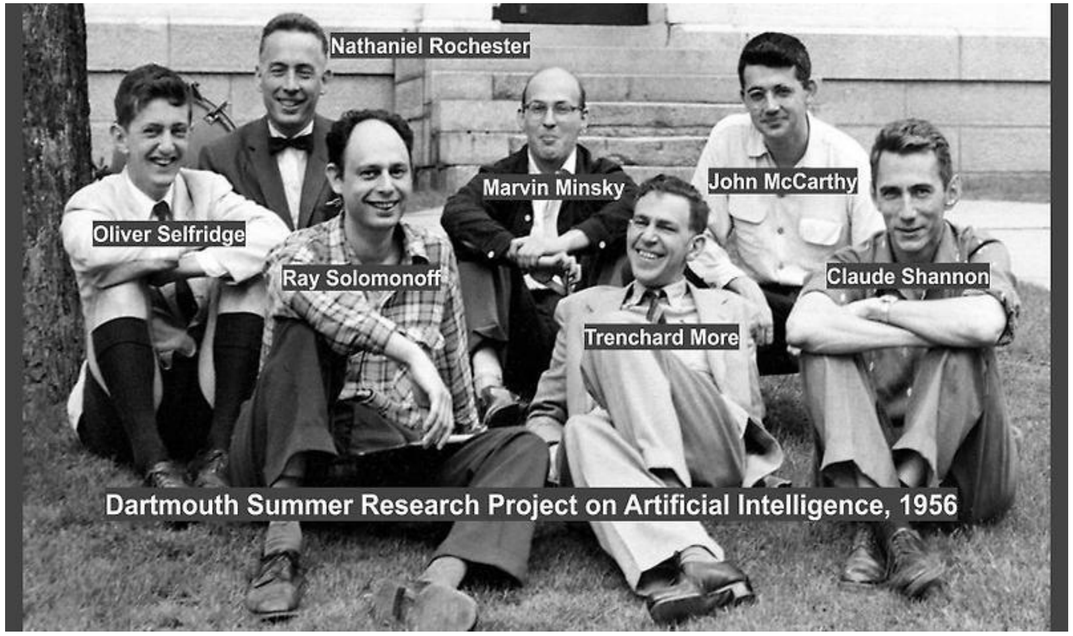
인공지능 (AI: Artificial Intelligence)

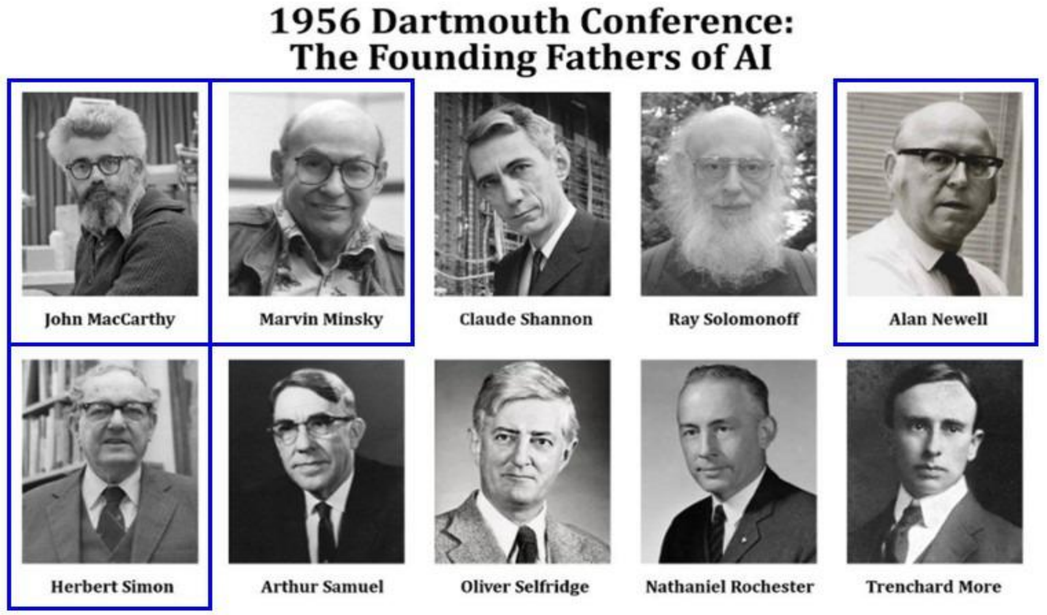
인공지능의 시작

1956년 다트머스 컨퍼런스

'인공 지능'이라는 개념을 처음으로 도입하여 지능형 기계를 만드는 과학과 공학으로 정의

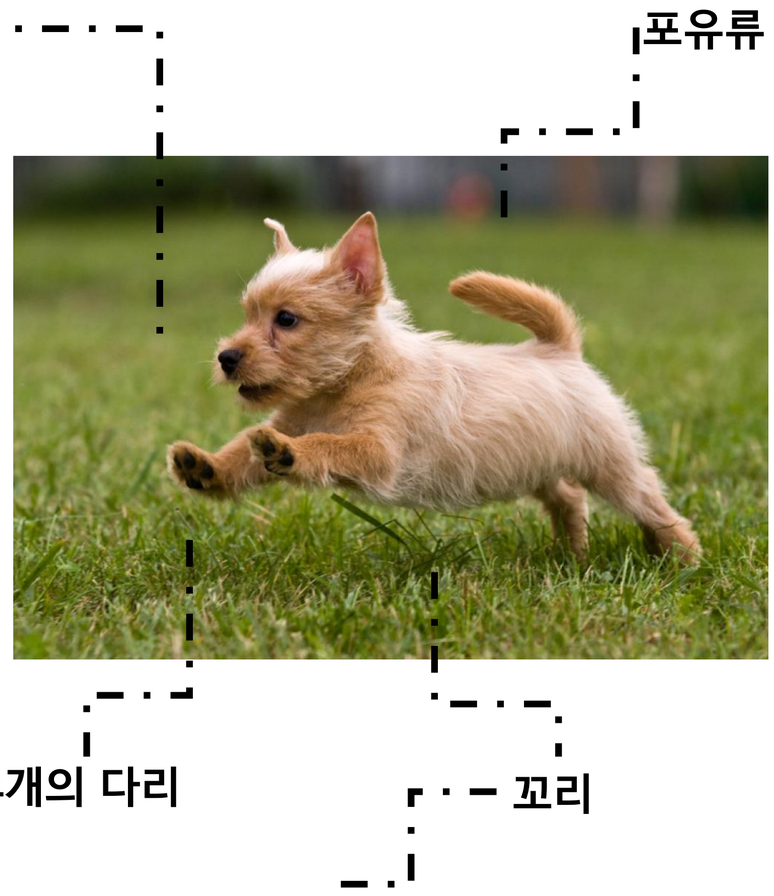
- 사물을 인식하는 기술 : 냉전 시대 적과 아군의 무기를 자동 구분하는 기술
- 언어를 이해하는 기술 : 냉전 시대 상대국의 문서를 빠르게 번역하는 기술



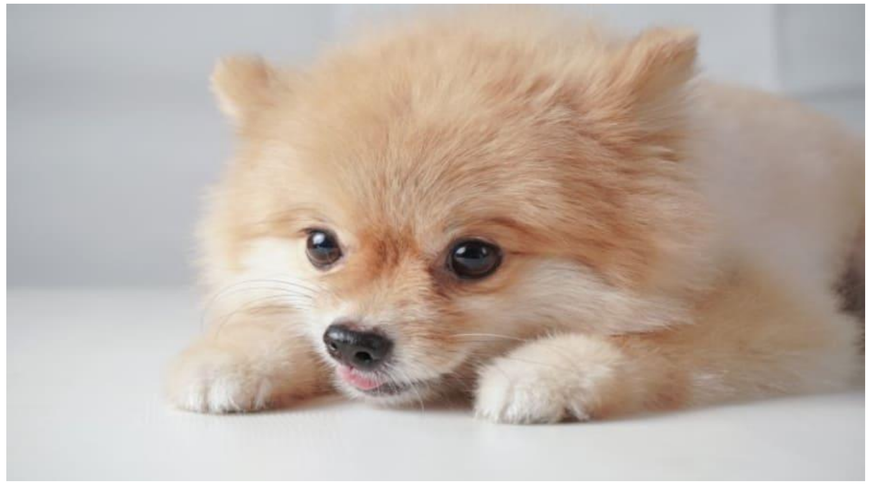


<출처 : Google>

규칙 기반 인공지능



강아지 학습 완료



<출처 : Google>

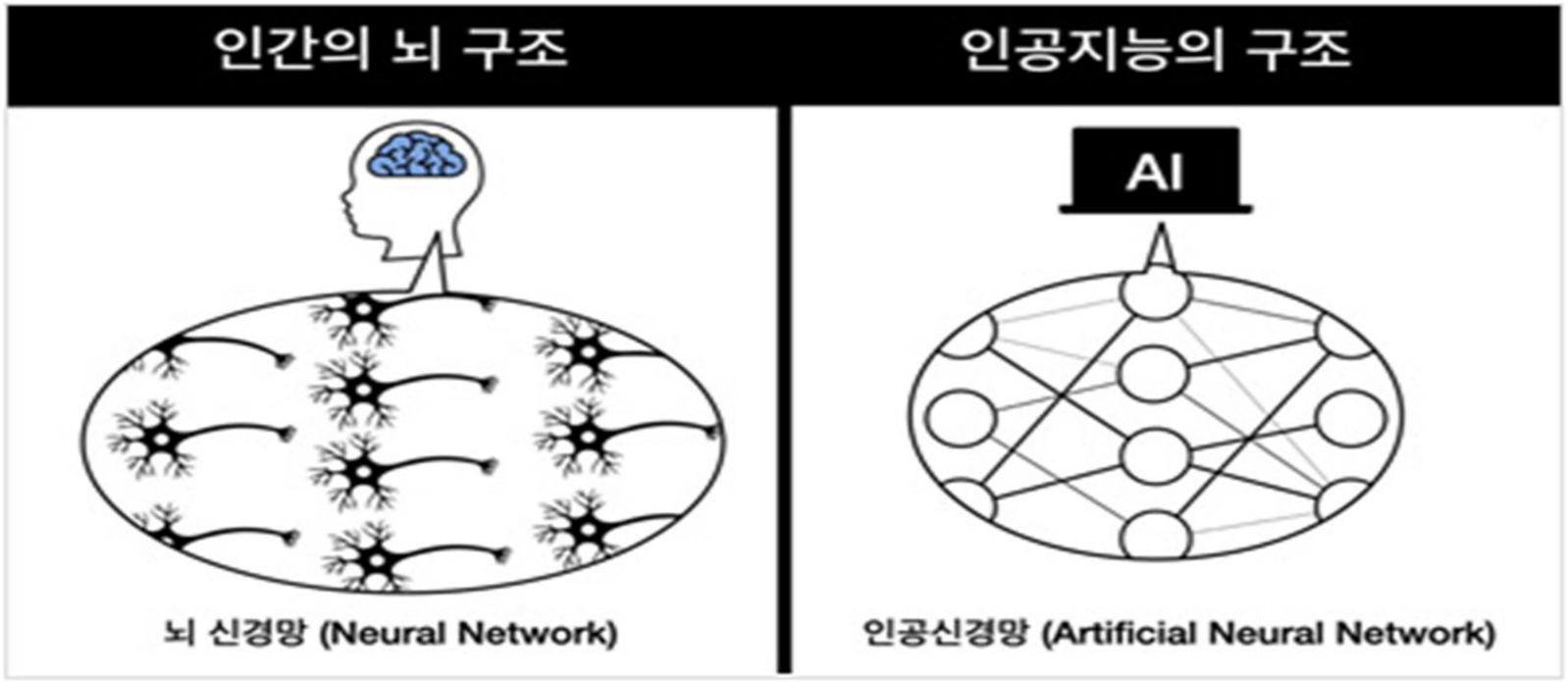
인공지능 신경망

인간의 뇌 신경망을 모방한 인공신경망 알고리즘

1980년대 ~ 1990년대 초

100조 개의 신경 세포 + 신경세포의 연결고리 (경험에 의한 가중치 설정)

* 가중치 (Weight) : 특정 요소가 차지하는 중요성을 나타내거나, 입력값의 중요도를 조절하는데 사용되는 수치



<출처 : 재료과학이야기>

규칙 기반 인공지능

규칙 기반 vs. 학습 기반 인공지능

규칙 기반 인공지능

→ 다리가 4개
꼬리가 있다
털이 있다 → 강아지
고양이

학습 기반 인공지능

= 강아지
고양이 → 다리가 4개
꼬리가 있다
털이 있다 → 강아지
고양이

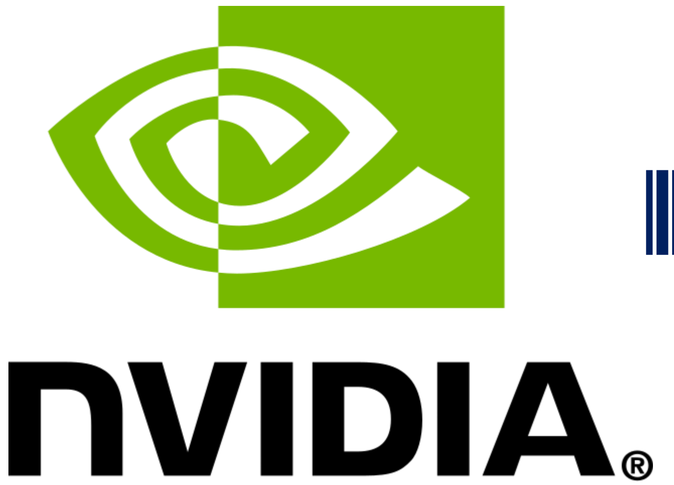
- University of Toronto
- 제프리 에버리스트 힌턴 교수
- 2024년 노벨 물리학상 수상

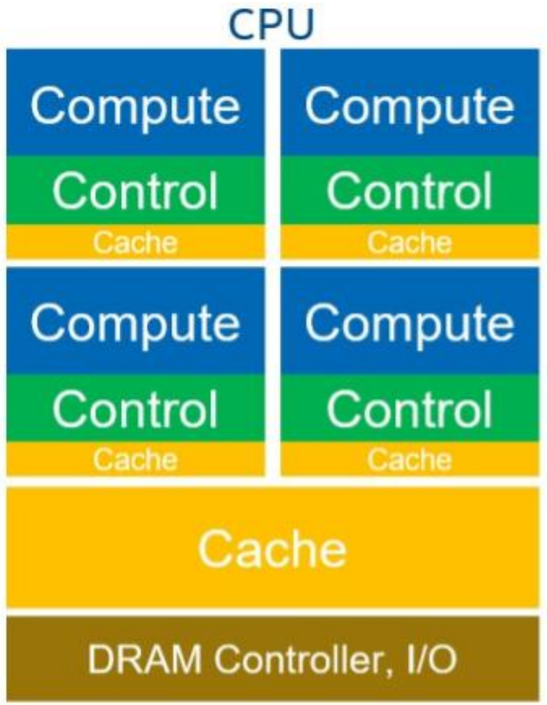
<출처 : Google>

CPU vs. GPU

CPU (Central Processing Unit)
복잡하고 다양한 연산을 **순차적**으로 수행

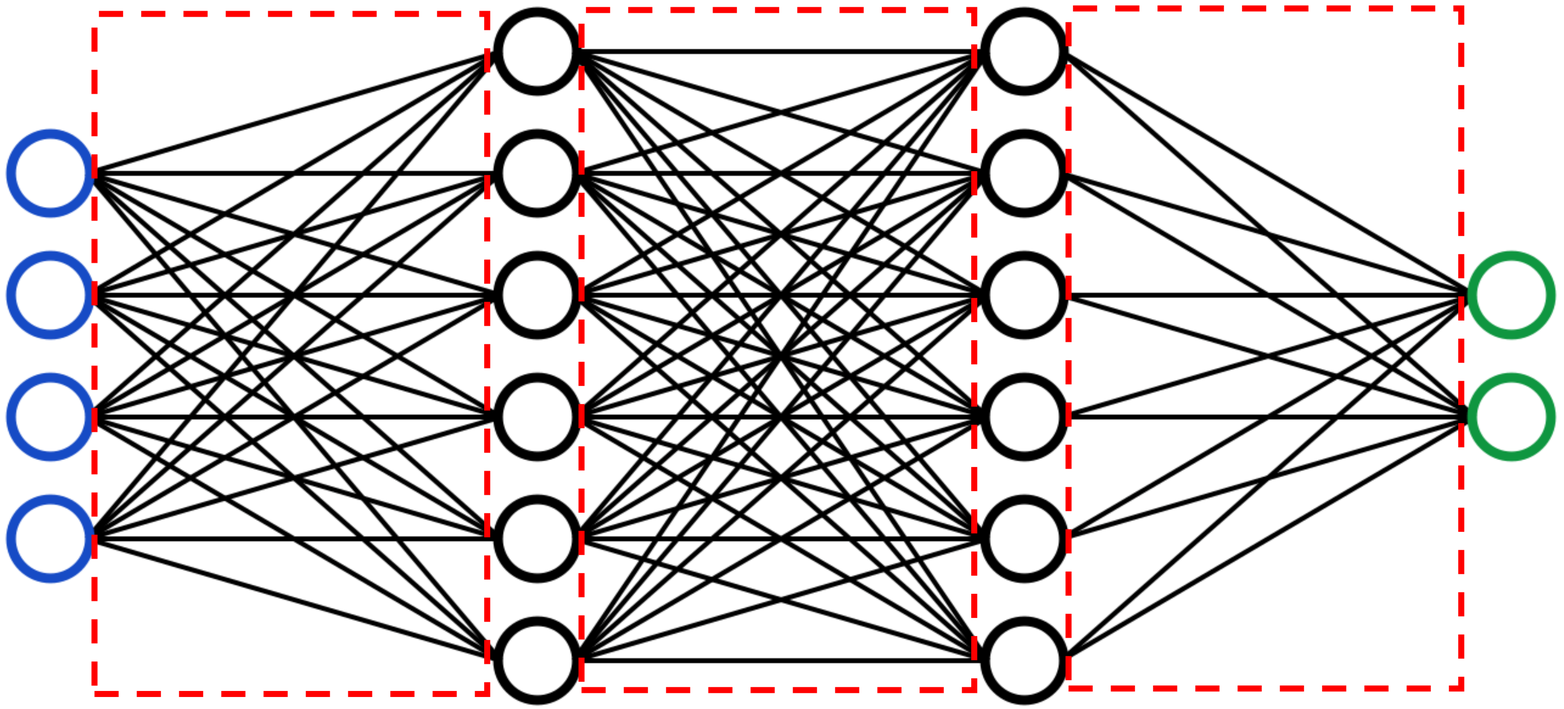
GPU (Graphical Processing Unit)
상대적으로 단순한 연산을 동시에 **병렬적**으로 수행





<출처 : Naver>

GPU의 활용



가중치 값만 동시에 계산

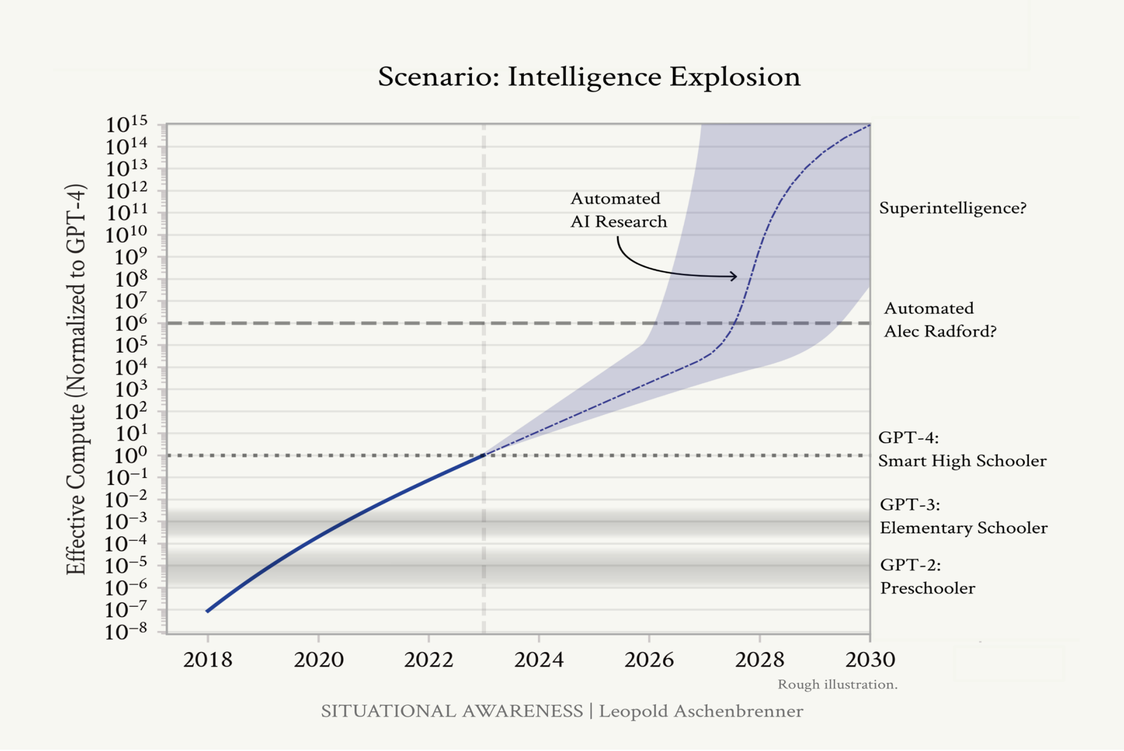
딥 러닝의 탄생

인간의 뇌 신경망을 모방한 인공신경망 알고리즘

인공신경망 연구

알고리즘 개선

<u>컴퓨팅 능력과 데이터 양의 폭발적인 증가</u>



<출처 : Situational awareness.ai>

인공지능 (AI: Artificial Intelligence)

기계가 인간처럼 사고하고, 판단하고, 행동하도록 만드는 기술 전반을 의미

1950년대부터 시작된 개념

머신러닝(ML)과 딥러닝(DL)은 AI의 하위 분야

예) 체스 두기, 로봇 제어, 음성 인식, 자동 번역 등

<출처 : FreePiK]>

<출처 : 인공지능신문>

<출처 : LG CNS>

머신러닝 (ML: Machine Learning)

AI의 하위 분야로, 기계가 명시적인 규칙 없이 데이터로부터 스스로 학습하는 기술

프로그래밍하지 않아도 예제를 통해 규칙을 학습

지도학습 (Supervised), 비지도학습 (Unsupervised), 강화학습 (Reinforcement) 등

예) 이메일 스팸 분류기, 주택 가격 및 주식 시세 예측 모델

<출처 : 비건뉴스>

<출처 : Medium>

<출처 : Medium>

딥러닝 (DL: Deep Learning)

머신러닝의 하위 분야로, 인공신경망(Neural Networks)을 기반으로 한 복잡한 문제를 처리하는 기술

사람의 뇌 구조를 모방한 다층 신경망 (Deep Neural Netowkrs) 사용

대용량 데이터와 높은 계산 성능이 필요

예) 자율주행차, 음성 비서, 얼굴 인식, ChatGPT 등

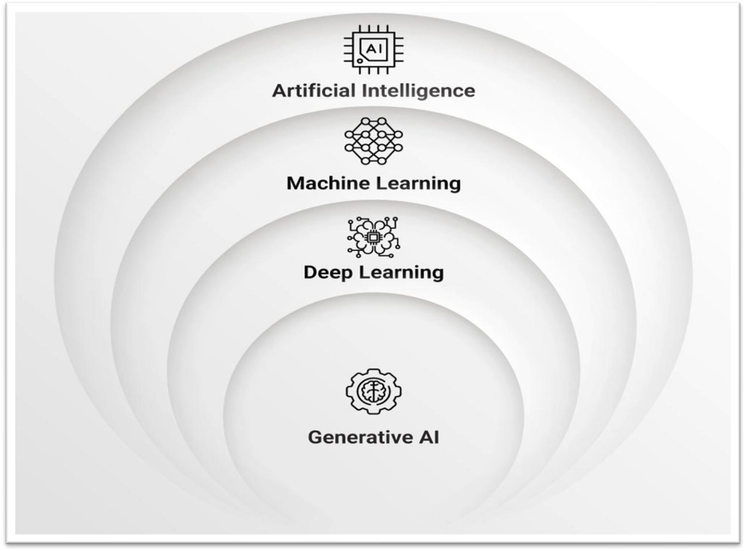
<출처 : Data Science Central>

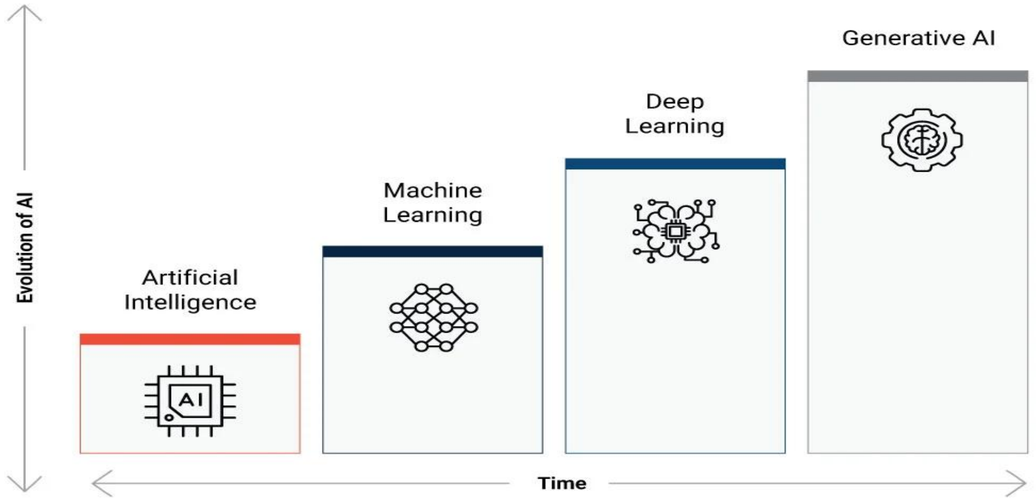
<출처 : 인공지능신문>

<출처 : itweb>

인공지능 vs. 머신러닝 vs. 딥러닝

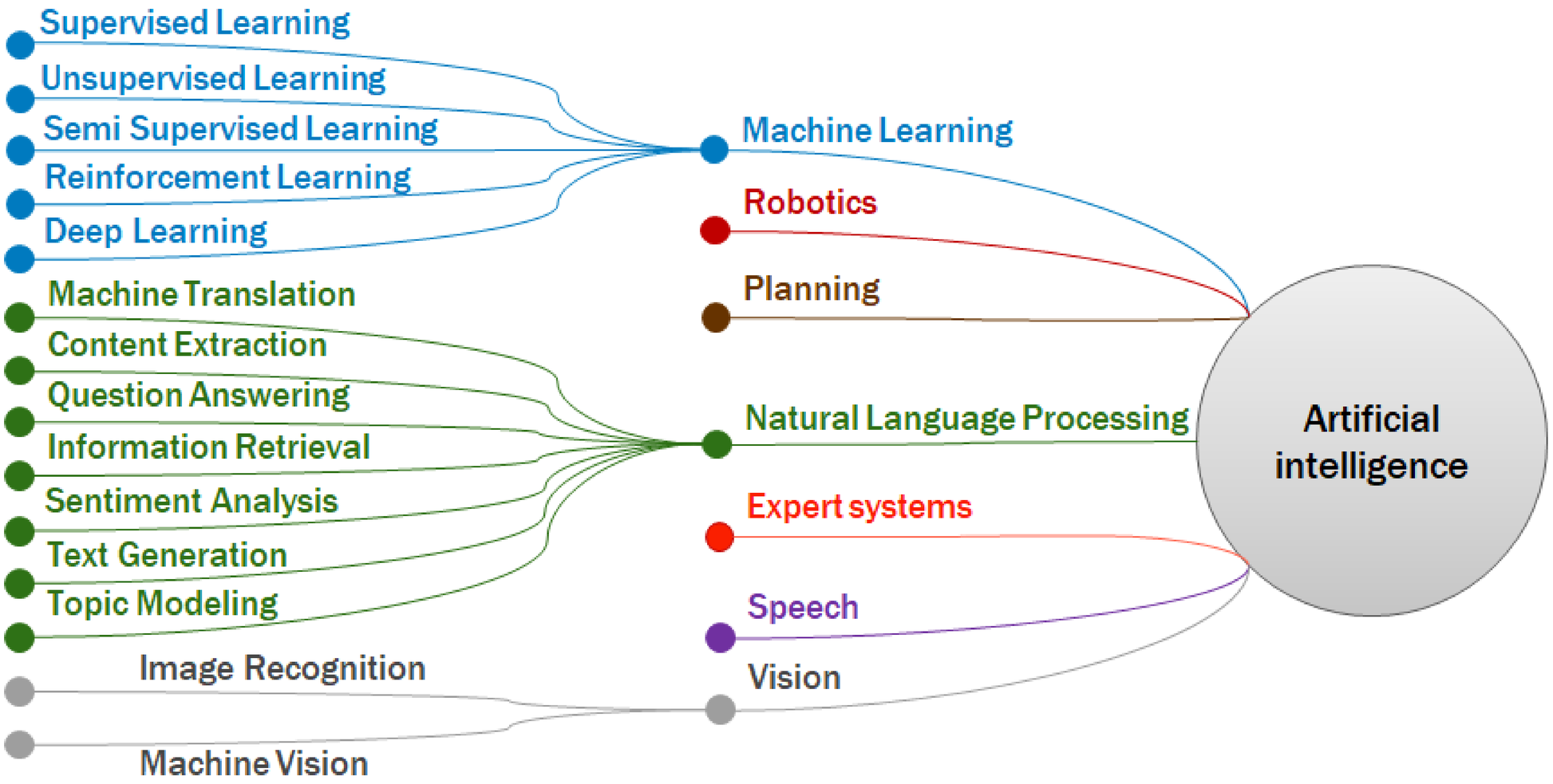
구분	인공지능(AI)	머신러닝(ML)	딥러닝(DL)
정의	인간처럼 판단하고 행동하는 기계	데이터에서 스스로 학습하는 알고리즘	신경망 기반의 고도화된 ML
특징	광범위한 개념	수학/통계 기반 학습	대용량 데이터+복잡한 모델 학습
학습방식	규칙 기반 또는 학습 기반	지도/비지도/강화 학습	대규모 인공신경망 사용
예시	로봇 청소기, 자율주행	이메일 분류기, 영화 추천	음성 비서, 얼굴 인식, 번역, GPT 등





<출처 : synoptek.com>

인공지능(AI) 하위 섹션



<출처 : Mukhamediev, Ravil I., et al. 2021>

AI 모델 공개 방식

구분	오픈 소스 (Open Source)	오픈 웨이트(Open Weight)	폐쇄형
모델구조	• 공개	• 공개	• 비공개/일부 공개
가중치 (파라미터)	• 공개	• 공개	• 비공개
훈련 코드	• 공개	• 비공개	• 비공개
훈련 데이터	• 공개	• 비공개	• 비공개
라이선스	• 개방	• 제한적 개방	• 사용 제한
사용자 수정	• 자유롭게 가능	• 조건부 가능	• 불가능
예시 모델	• 미스트랄 7B	• LLaMA2, DeepSeek	• GPT5, Gemini 1.5

<출처 : 조선일보>

파운데이션 모델 (Foundation Model)
특정 업무용으로 한정된 "전문 모델"이 아니라 다양한 인공지능 서비스를 만드는 **공통의 기반이 되는 초대형 범용 모델**
예) GPT, LLaMA, Gemini

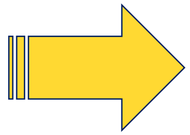
RAG (검색 증강 생성)

RAG (Retrieval-Augmented Generation)

대규모 언어 모델 (LLM)이 응답을 생성하기 전에 외부 지식 베이스에서 관련 정보를 검색하고, 이를 바탕으로 생성하는 기술

LLM이 학습 데이터에 없는 최신 정보나 특정 비공개 정보를 접근할 수 있게 하여 응답의 정확도를 높이고 환각 (Hallucination)을 줄임

인터넷 지식 기반의
AI 프로그램
(**LLM, 오픈 웨이트**)



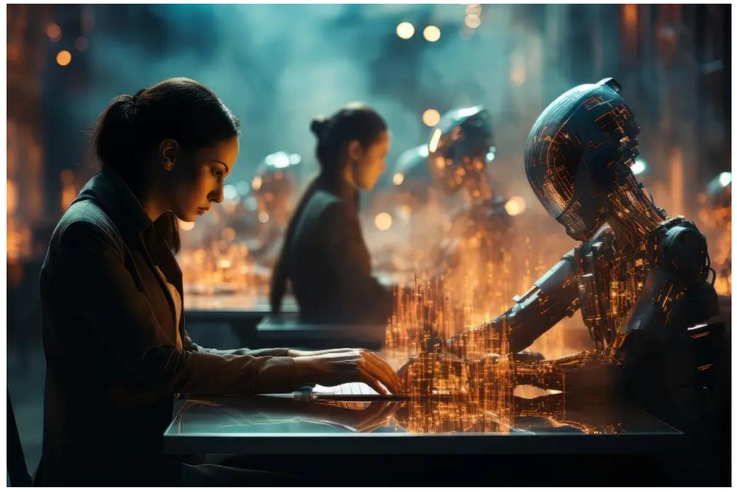
전문적인 분야의
AI로 활용

외부 데이터 소스를 통해
전문 지식을 학습

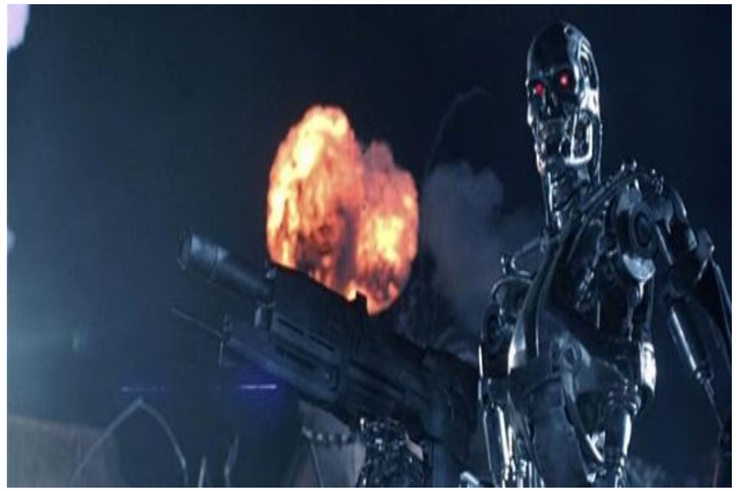
범용 인공지능
(AGI, Artificial General Intelligence)

범용 인공지능(AGI)

구분	특정 목적 인공지능, ANI (Artificial Narrow Intelligence)	범용 인공지능, AGI (Artificial General Intelligence)	초인공지능, ASI (Artificial Super Intelligence)
정의	단일 작업에 특화된 인공지능	인간과 비슷한 수준의 지능을 가진 인공지능	인간 지능을 뛰어넘는 인공지능
능력범위	한 가지 작업에 최적화	다양한 문제를 스스로 해결 가능	인간이 이해하기 어려운 수준의 문제 해결 및 감정 이해 가능
학습능력	특정 작업에 제한된 데이터를 학습하며, **인간의 개입** 필수	다양한 분야에 걸쳐 **인간 수준으로 스스로** 학습 목표를 정하고 판단	인간이 모르는 영역도 **인간 도움 없이 스스로** 학습하고 발전 가능
실현단계	대중화	개발 중	개념 단계
위험성	낮음 (통제 가능)	오작동 가능성	높음 (통제 불가능)



<출처 : FreePiK]>



10년 안에 나올 스마트폰 이상의 혁명

스마트 글래스의 장점

소비자의 **1인칭 시선**을 인공지능이 같이 볼 수 있음

- (현재) **검색** → 추천 알고리즘 → 인간
- (미래) **인공지능** → 추천 알고리즘 → 인간

<출처 : Apple>

<출처 : Google Glass>

<출처 : Ray-Ban | Meta>

AGI 개발 이유

인류가 직면한 거대 문제(Meta Problem)를 해결하는 핵심 열쇠

에너지 효율 최적화 (→ 무한한 에너지 생성)

감염병 예측 및 대응 (→ 인간의 질병 문제 해결)

과학 혁명의 가속기(Accelerator) 역할 (→ 양자역학과 상대성 이론)

AGI → **로봇공학**

무한으로 증가하는
제조업 생산성

대부분 제품 가격이 0원

노동의 가치 ↓

AGI가 대체 **데이터 센터 증설**

노동 투입량 x 자본 투입량 = 사회 생산성

AGI가 만든 새로운 자본주의

자본의 가치 ↑

인공지능이 일자리에 미치는 영향

챗GPT 출시 이후, 일자리 대체 예측

- 단순 노동의 소멸
- 나이 많은 노동자들의 소멸

일자리 데이터 분석 결과,

22세에서 25세 사이의 근로자는 2022년 말부터 2025년 7월까지 고용이 6% 감소했습니다.
반면, 고령 근로자는 6~9% 증가하였습니다.

Our second key fact is that overall employment continues to grow robustly, but employment growth for young workers in particular has been stagnant since late 2022. In jobs less exposed to AI young workers have experienced comparable employment growth to older workers. In contrast, workers aged 22 to 25 have experienced a 6% decline in employment from late 2022 to July 2025 in the most AI-exposed occupations, compared to a 6-9% increase for older workers. These results suggest that declining employment AI-exposed jobs is driving tepid overall employment growth for 22- to 25- year-olds as employment for older workers continues to grow.

<출처 : Standford Digital Economy>

문과생 프로그래머 열풍

개발자 모시기 경쟁에 문과생들도 코딩 열풍

| 졸업 후 IT 실무 역량 쌓는 문과생 多...비전공 개발자 증가세

인터넷 | 입력 :2021/04/06 17:06 수정: 2021/04/06 17:10

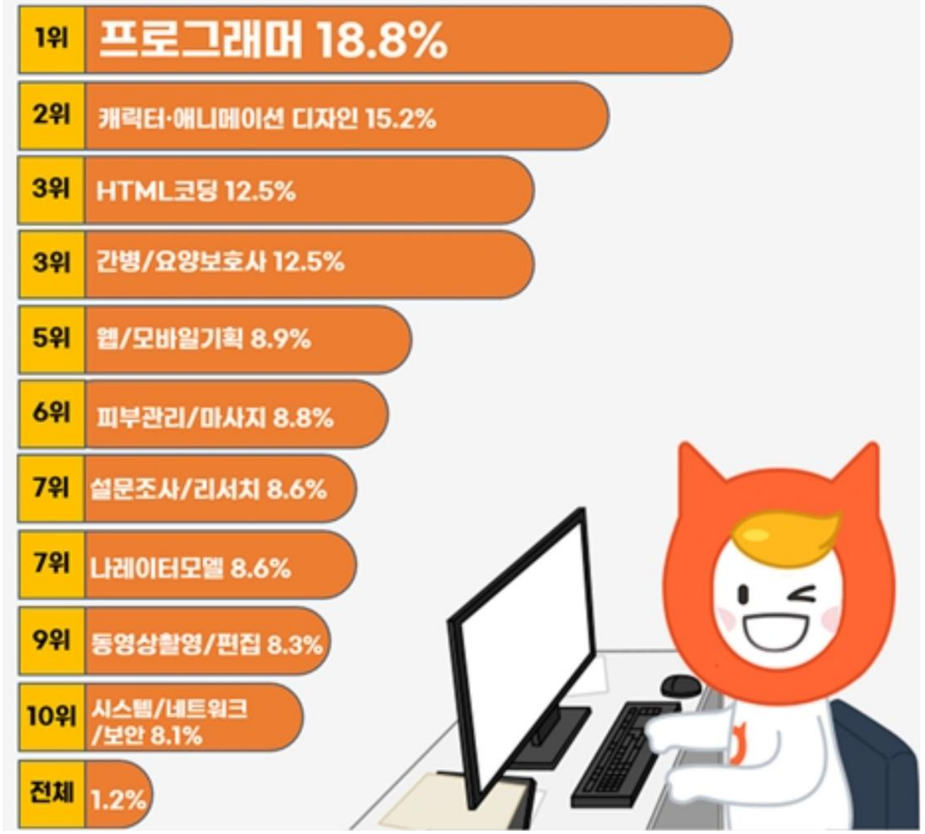
취업난 속 '문송 탈출' 러시... "요즘 코딩 교육생 절반은 문과 출신"

최연진 기자 입력 2021.07.28 16:13 수정 2021.07.28 21:19 | 1면 2 0

<출처 : 알바몬>

직종별 알바 시급 인상률 TOP10

| 자료제공: 알바몬(올해 1분기 아르바이트 시급 데이터 분석 결과)



<출처 : 알바몬>

'AI, 일자리 위협? 파트너?' 부산 직장인 세대별 인식차

등록 2025.10.16 10:15:37 | 수정 2025.10.16 11:20:24

20대 "일자리 뺏긴다" , 50대 "업무 효율 높이는 도구"

AI 활용률 74.4%, 2년 새 급등…교육 수요도 77% 달해

실제가 없는 직업
소프트웨어는 실체가 없기 때문에 없애는 비용이 들지 않는다
소프트웨어, 컨텐츠 등

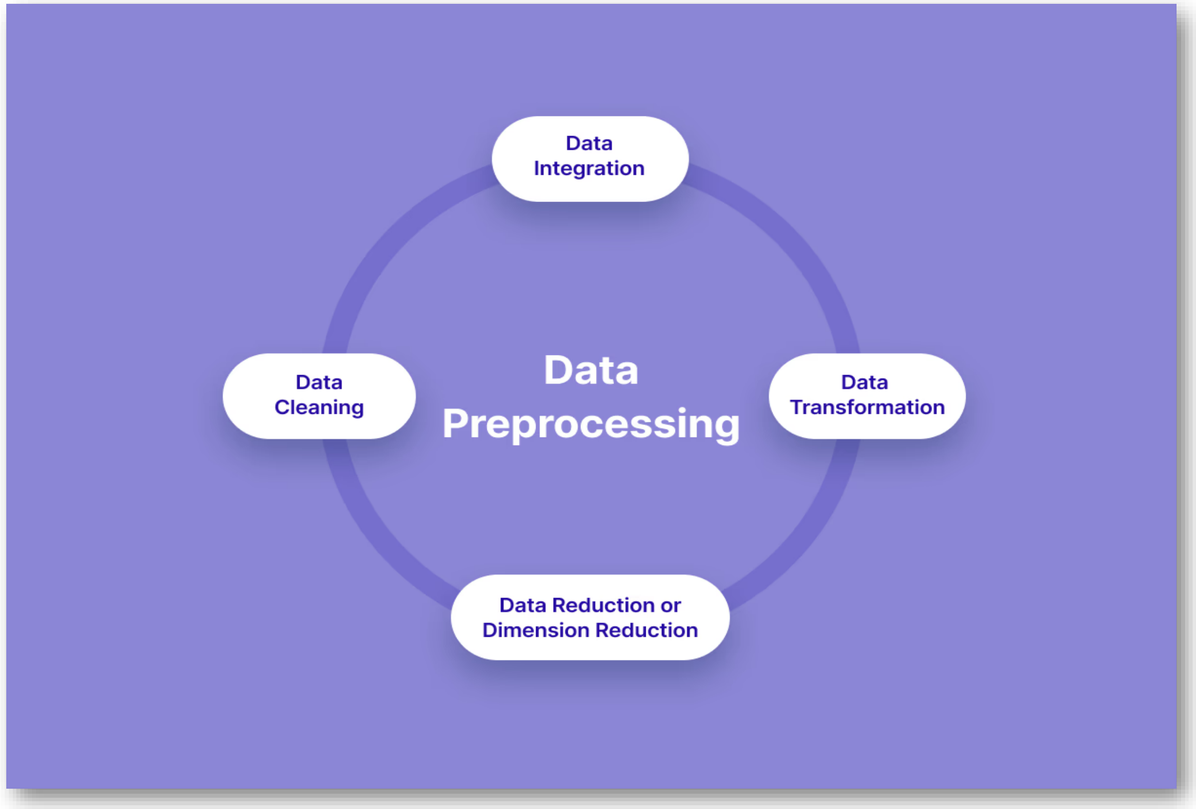
→

실제가 있는 직업
공장, 제조업

→

철학적으로 바꾸기 어려운 직업들
종교인, 판사, 의사

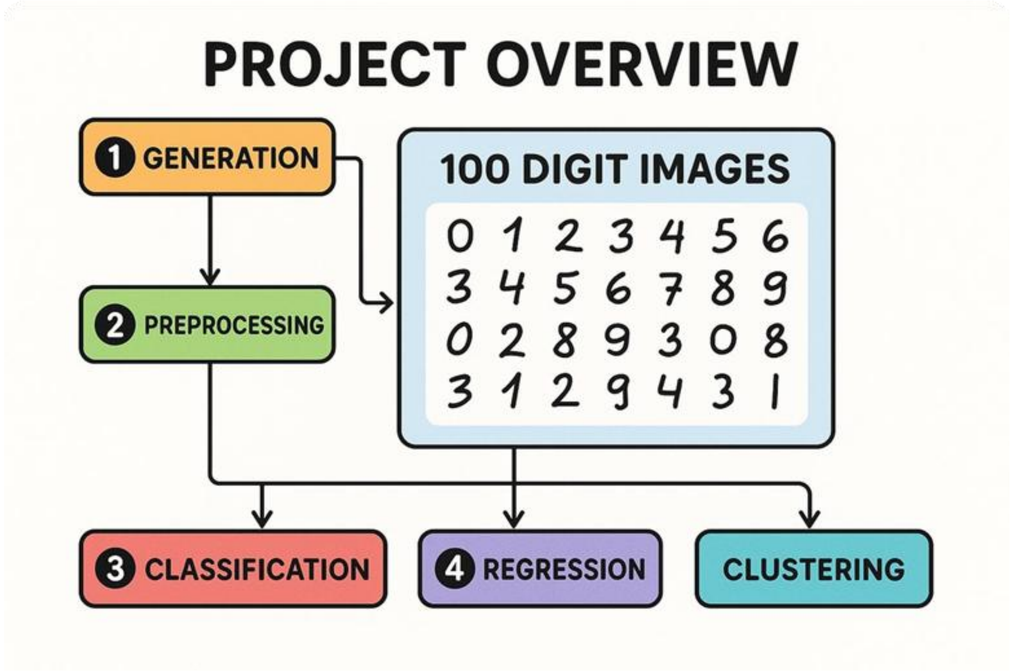
실습 프로젝트 개요


Data
Integration
Data
Cleaning
Data
Preprocessing
Data
Transformation
Data Reduction or
Dimension Reduction

실습 프로젝트 개요

100개 숫자 이미지 데이터셋으로 전체 머신러닝 워크플로우 체험하기

단일 데이터셋을 직접 생성하여 다양한 머신러닝 알고리즘을 적용하고 비교분석 함으로써 각 기법의 특징과 차이점을 명확하게 이해합니다.



100개 숫자 이미지 데이터셋 워크플로우 - 데이터 생성, 전처리, 분류, 회귀, 군집화 단계

실습 세부 내용

- **데이터 생성:** NumPy를 활용하여 0-9까지 다양한 스타일의 숫 자 이미지 100개 생성 (각 숫자당 10개)
- **분류 실습:** KNN 활용 이미지 숫자 분류
- **회귀 실습:** 다양한 회귀 모델 적용 이미지 숫자 예측
- **군집화 실습:** K-Means을 활용한 군집화

핵심 포인트: 동일한 데이터셋에 다양한 기법을 적용함으로써, 각 알고리즘의 강점과 약점을 직접 체험하고 비교할 수 있습 니다.

프로젝트 진행 단계

데이터 생성 & 전처리 | 분류 알고리즘 | 회귀 알고리즘 | 군집화 알고리즘

머신러닝(ML) 기본 개념

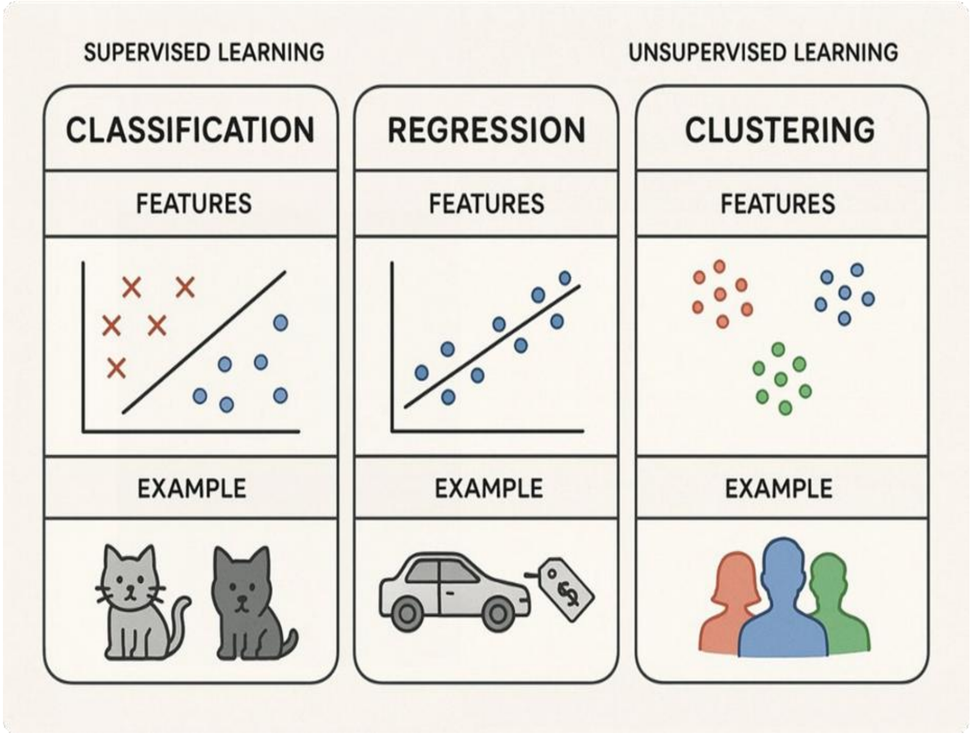
머신러닝 학습 유형

지도학습: 레이블이 있는 데이터로 학습하며, 입력(X)과 출력(Y) 사이의 관계를 모델링합니다.

비지도학습: 레이블이 없는 데이터에서 패턴을 발견하고, 데이터의 내재된 구조를 학습합니다.

주요 머신러닝 알고리즘 유형

유형	학습 방식	특징 및 예시
지도학습 **분류**	데이터를 사전 정의된 범주로 구분	- 이메일 스팸 필터링 - 숫자 이미지 인식
지도학습 **회귀**	연속적인 수치 값 예측	- 주택 가격 예측 - 판매량 예측
비지도학습 **군집화**	유사한 데이터를 그룹화	- 고객 세분화 - 유사한 숫자 이미지 그룹화



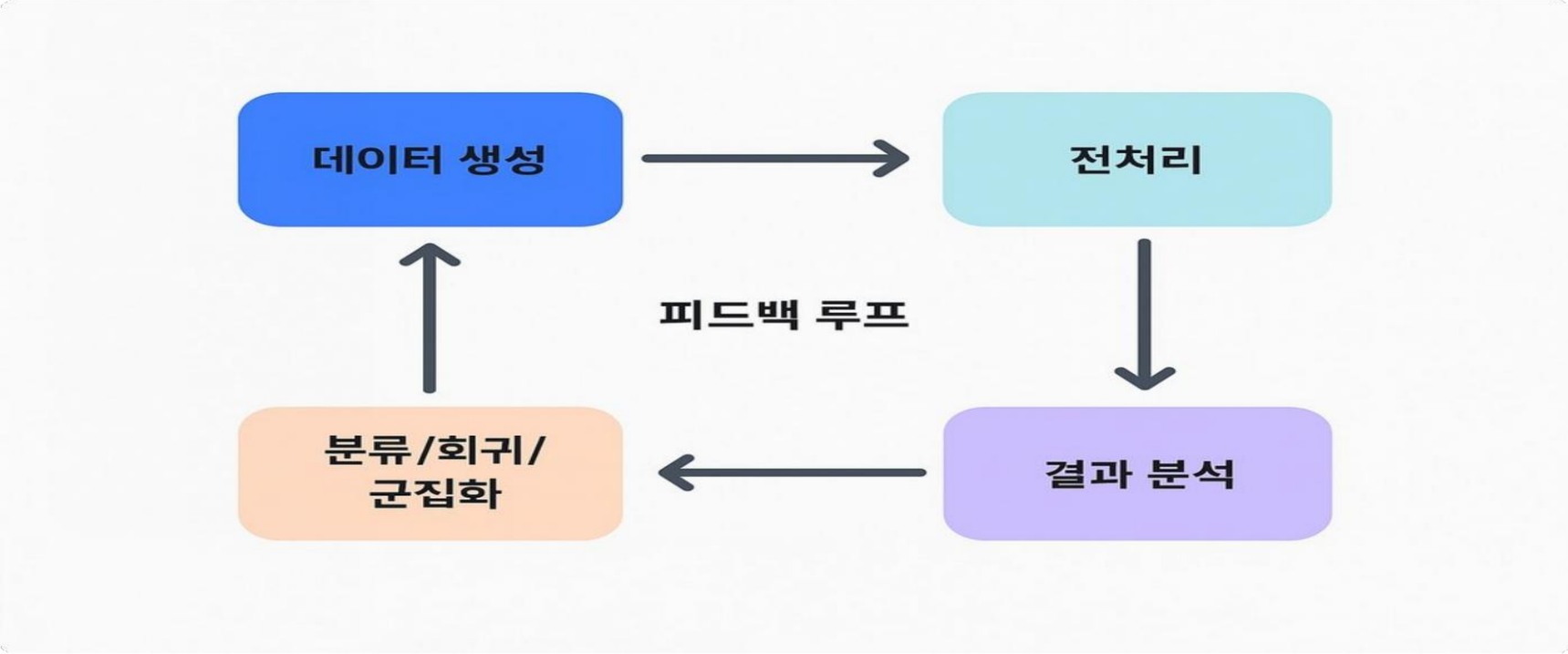
학습 프로세스 차이

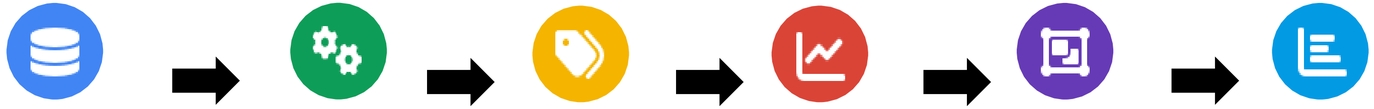
지도학습은 "정답"을 통해 학습하고, 비지도학습은 데이터 자체의 구조를 탐색합니다.

머신러닝(ML) 실습 전체 워크플로우

머신러닝 프로젝트의 완전한 순환 구조

본 실습에서는 데이터 생성부터 분석까지 전체 머신러닝 워크플로우의 모든 단계를 경험합니다.

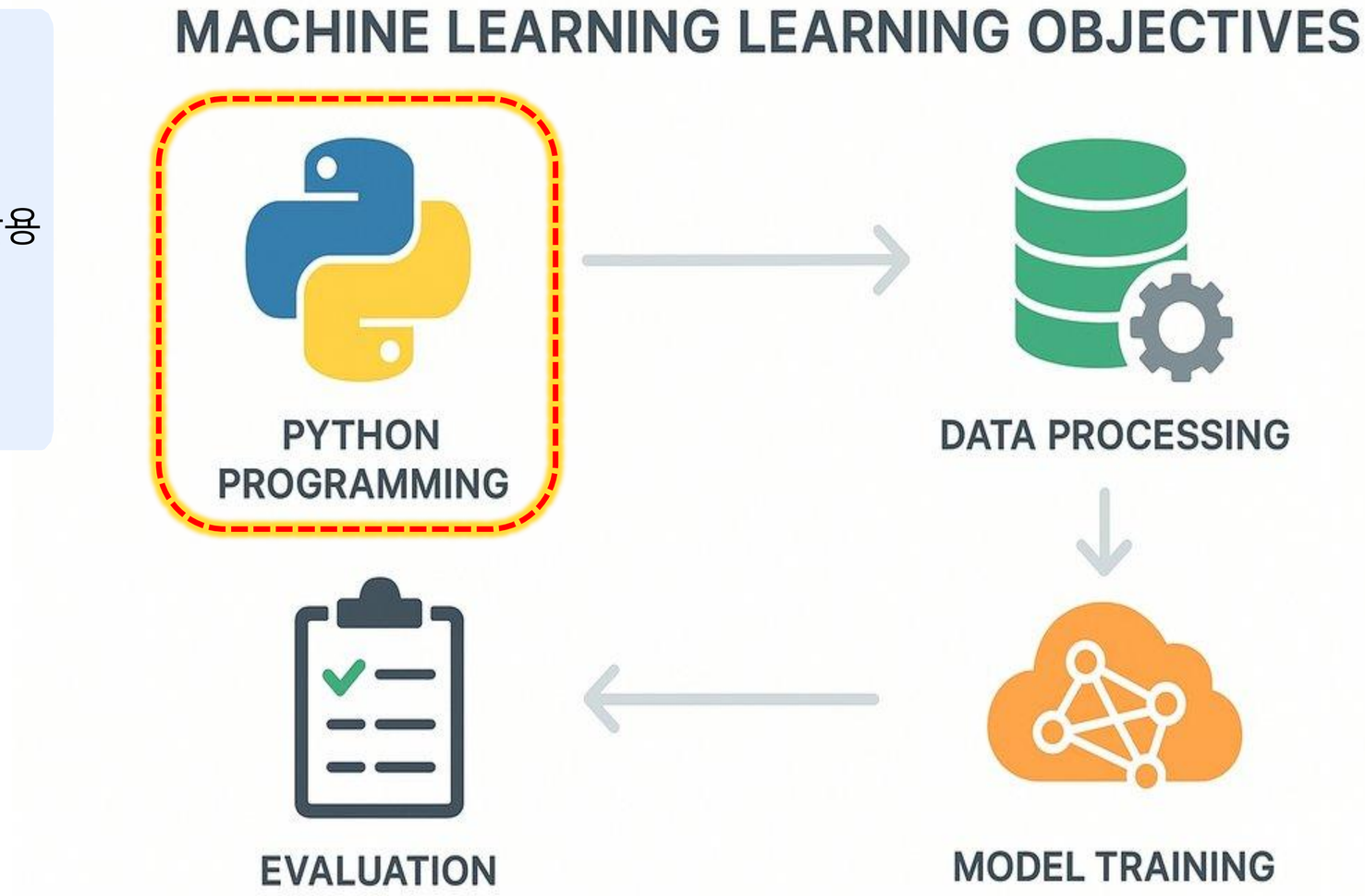




머신러닝(ML) 학습 목표

습득 가능 핵심 역량

1. Python 및 데이터 분석 라이브러리 활용
2. 데이터 생성 및 전처리
3. 머신러닝 알고리즘 구현 및 최적화
4. 모델 평가 및 결과 시각화 능력

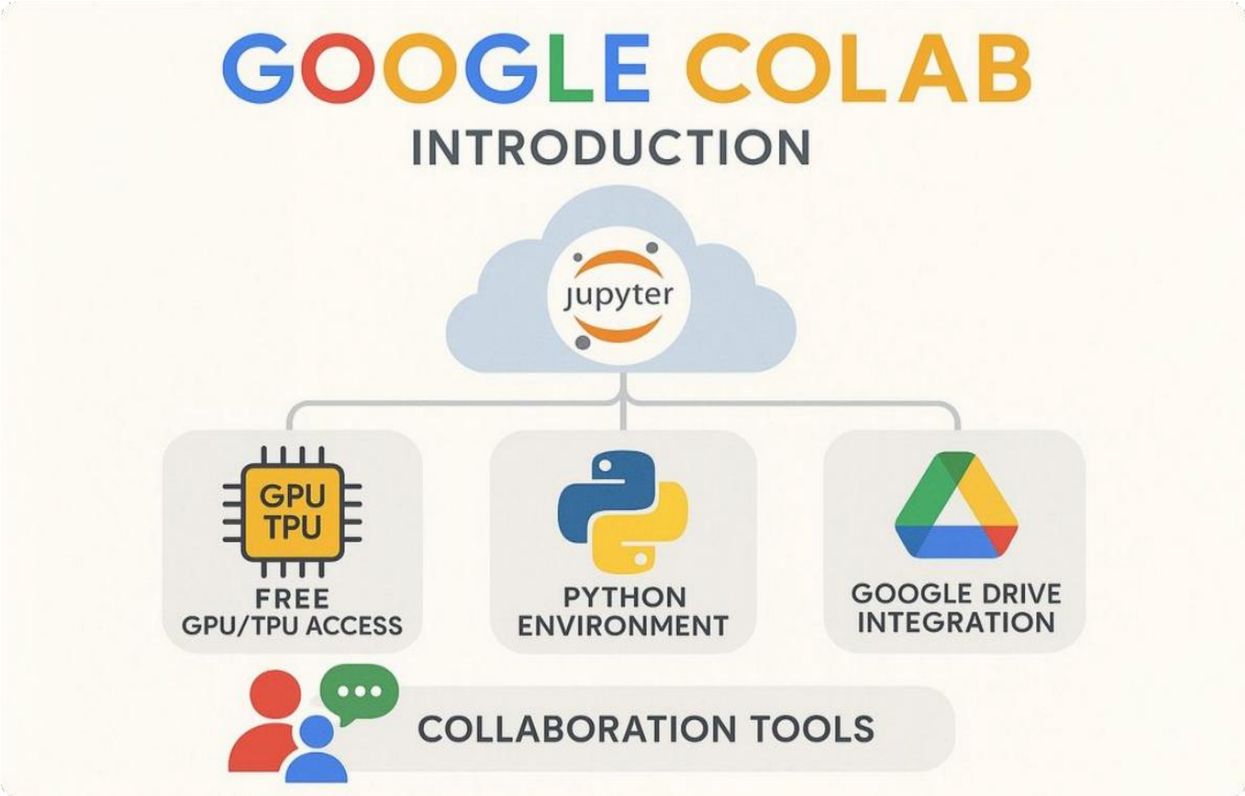


Colab 활용법

Google Colab이란?

Google의 클라우드 기반 Jupyter Notebook 환경

코랩(Colab)은 코드 작성, 실행 및 공유가 가능한 클라우드 기반의 주피터 노트북 환경으로, 브라우저만으로 Python 코드를 실행할 수 있습니다.



Jupyter Notebook과의 차이점

특징	Google Colab	Jupyter Notebook
실행 환경	클라우드 기반	로컬 환경
설치 필요	불필요(브라우저만)	로컬 설치 필요
컴퓨팅 리소스	Google 제공(무료)	로컬 리소스 사용
협업 기능	실시간 공동 작업	제한적

Colab의 주요 장점

- 설치없이 브라우저에서 바로 사용
- 무료 GPU/TPU 사용 가능
- GitHub 연동 및 공유 기능 (실시간 협업 지원)
- Google Drive 저장소 연결 가능
- 사전 설치된 라이브러리

Colab 환경 준비

실습 시작을 위한 Colab 환경 설정

Google Colab은 브라우저에서 바로 Python 코드를 작성하고 실행할 수 있는 환경으로, 머신러닝 실습에 최적화되어 있습니다.

1 Google 계정 생성

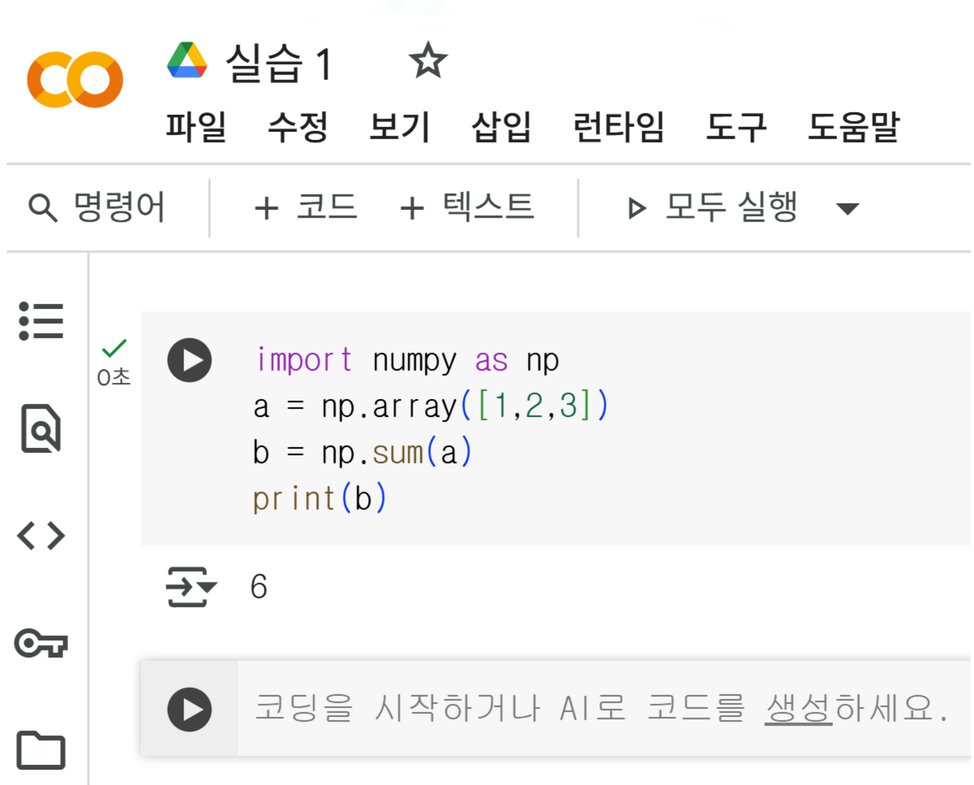
Google 계정이 없다면 accounts.google.com에서 새로운 계정을 만듭니다.

2 Google Drive 접속

Google Drive(drive.google.com)에 접속하여 로그인합니다.

3 Colab 노트북 생성

왼쪽 상단 '새로만들기' 버튼 클릭 → '더보기' → 'Google Colaboratory' 선택



Google Colab 노트북 인터페이스 – 코드 셀과 텍스트 셀이 표시된 화면

필수 라이브러리 설치 및 환경설정

머신러닝 실습을 위한 핵심 라이브러리

Google Colab은 대부분의 머신러닝 라이브러리가 사전 설치되어 있지만, 최신 버전으로 업데이트하거나 추가 라이브러리 설치가 필요할 수 있습니다.

NumPy

수치 계산을 위한 핵심 라이브러리로 다차원 배열과 행렬 연산을 지원합니다.

Matplotlib

데이터 시각화를 위한 파이썬 라이브러리로 그래프, 차트 등 다양한 시각화 기능을 제공합니다.

scikit-learn

머신러닝 알고리즘을 쉽게 구현할 수 있는 종합 라이브러리로 분류, 회귀, 군집화 등을 지원합니다.

Version 확인 명령어

✓ 1초

```
import numpy as np
import matplotlib as mpl
import sklearn
print(f'Numpy: {np.__version__}, Matplotlib: {mpl.__version__}, scikit-learn: {sklearn.__version__}')
```

```
Numpy: 2.0.2, Matplotlib: 3.10.0, scikit-learn: 1.6.1
```

Colab 사용법 기초

Google Colab 기본 인터페이스 활용하기

Colab은 코드셀과 마크다운셀로 구성된 노트북 형태로, 코드 실행과 문서화를 동시에 할 수 있는 대화형 환경입니다.

셀 타입과 기능

- **코드 셀 :** Python 코드를 작성하고 실행
- **텍스트 셀 :** 서식있는 텍스트, 이미지, 수식

기본 작업 방법

- **셀 실행 :** 셀 왼쪽 실행 버튼 또는 Ctrl+Enter
- **새 셀 :** '+코드'/'+텍스트 ' 버튼
- **셀 타입 변경 :** '코드' 또는 '텍스트 ' 선택

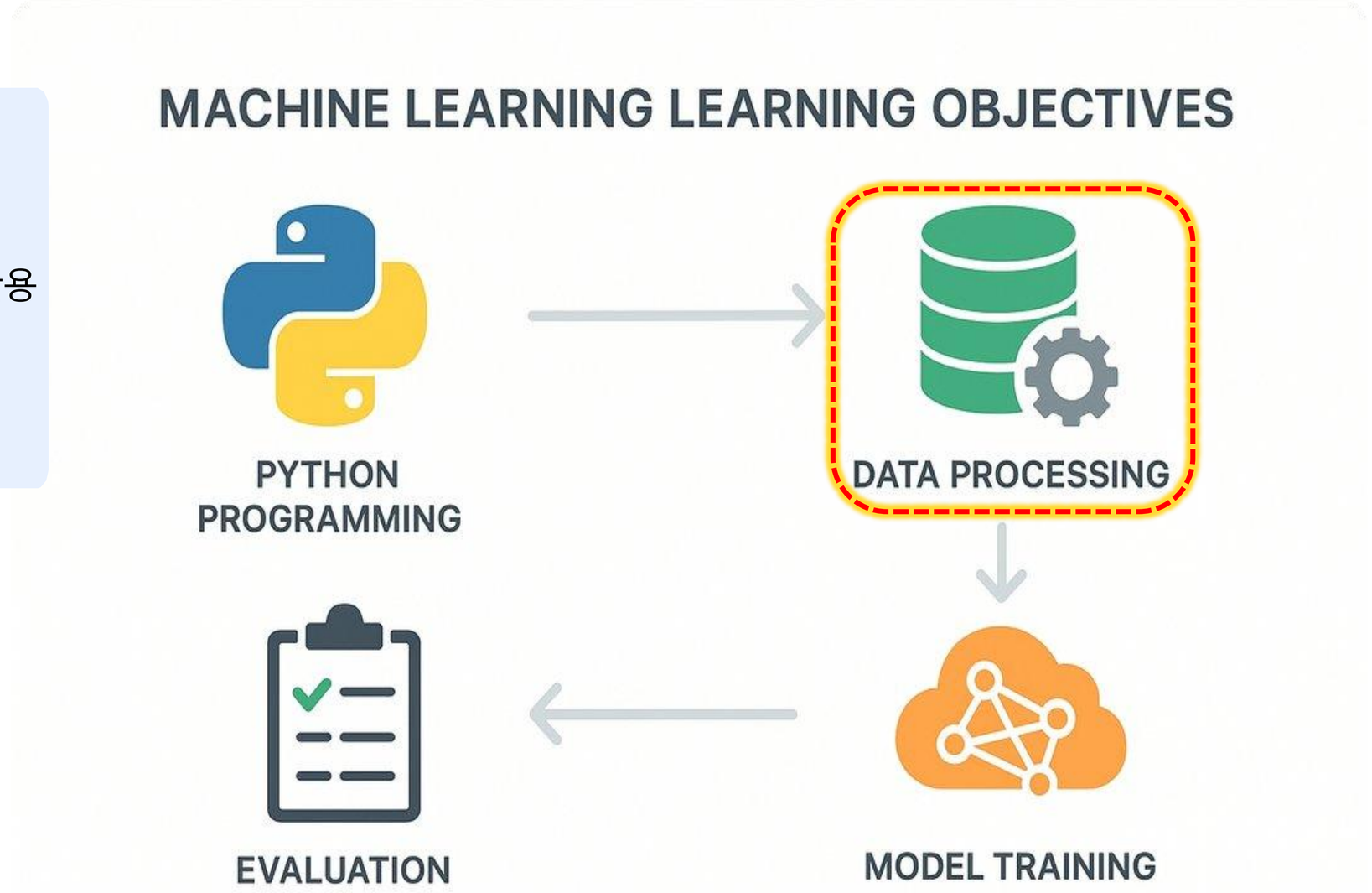
파일 관리와 드라이브 연동

- **파일 업로드 :** 파일 탐색창 또는 files.upload() 함수
- **파일 다운로드 :** files.download('파일명') 함수
- **드라이브 연동 :** drive.mount('/content/dirve')

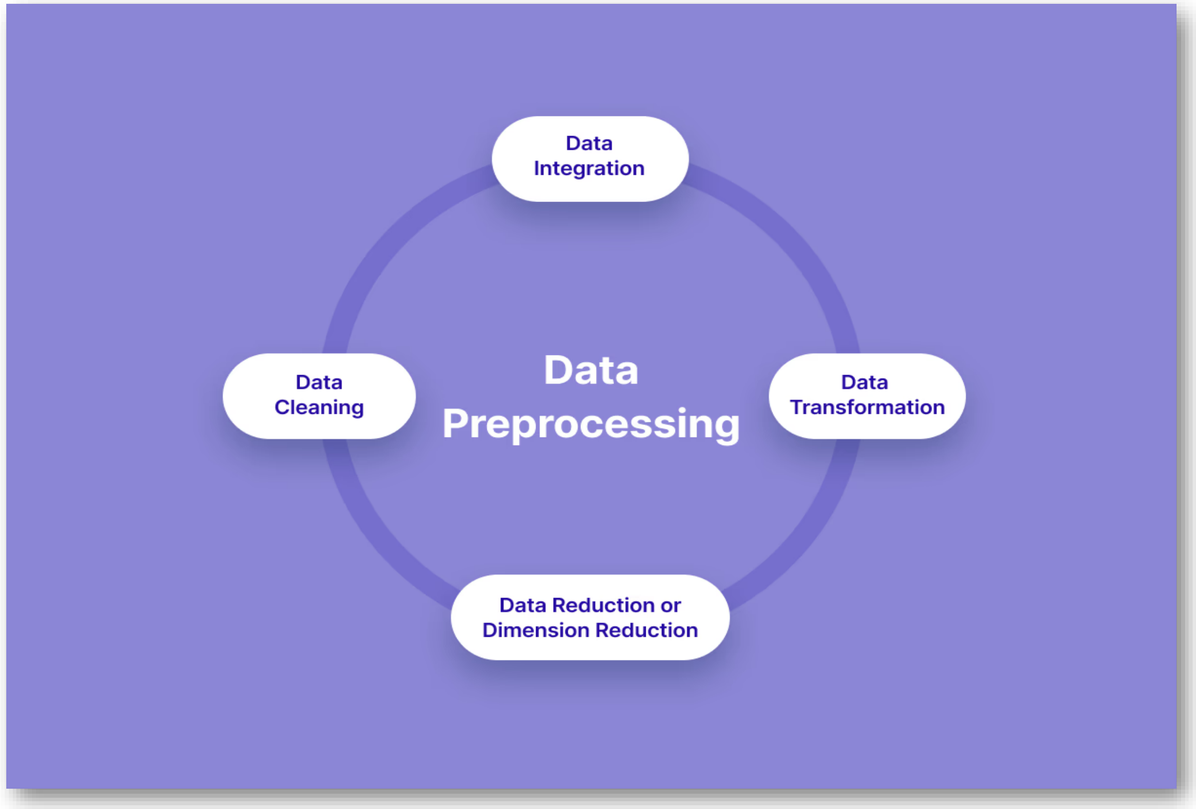
머신러닝(ML) 학습 목표

습득 가능 핵심 역량

1. Python 및 데이터 분석 라이브러리 활용
2. 데이터 생성 및 전처리
3. 머신러닝 알고리즘 구현 및 최적화
4. 모델 평가 및 결과 시각화 능력



데이터 생성 및 전처리


Data
Integration
Data
Cleaning
Data
Preprocessing
Data
Transformation
Data Reduction or
Dimension Reduction

숫자 이미지 생성

Numpy와 Matplotlib을 활용한 28×28 숫자 이미지 합성
숫자 이미지 생성은 모델 학습에 필요한 데이터를 생성하고 데이터 특성을 이해하는 기초가 됩니다.

숫자 이미지 생성의 기본 원리

- **클래스 균형 :** 0부터 0까지의 숫자가 균등하게 분포되도록 설계 (각 숫자당 10개씩, 총 100개)
- **다양한 형태 :** 굵기, 기울기, 크기 등의 다양한 변형을 포함하여 실제 손글씨의 변동성 반영
- **노이즈 포함 :** 랜덤 노이즈, 블러, 픽셀 변형 등을 적절히 추가하여 모델의 강건성 향상
- **배경 변화 :** 다양한 배경 밝기와 텍스처를 적용하여 실제 환경의 다양성 반영
- **위치 변화 :** 숫자의 중심 위치와 회전 각도를 조금씩 변형하여 다양한 배치 학습

숫자 변형 예시

100개 숫자 이미지 데이터셋 생성 및 저장

데이터셋 생성 프로세스

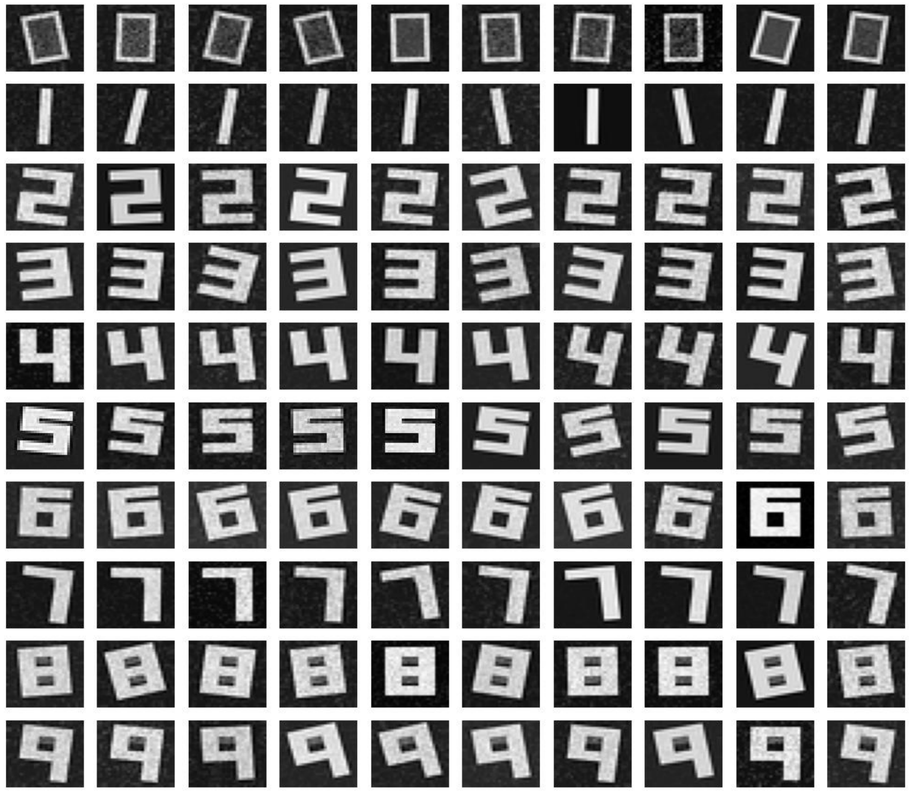
0부터 9까지의 숫자를 각 10개씩 변형하여 총 100개의 숫자 이미지를 생성하고 Numpy 배열로 저장합니다.
모델 학습을 위한 데이터셋을 직접 생성함으로써 데이터의 특성과 분포를 완전히 제어할 수 있습니다.

데이터셋 생성 단계

- 각 숫자별로 10개씩 변형된 이미지 생성
- 다양한 굵기, 회전, 노이즈 적용
- 28×28 픽셀 크기의 그레이스케일 이미지
- 생성된 이미지를 Numpy 배열에 저장

저장된 파일명

1. “digits_data.npy”
2. “digits_labels.npy”
3. “digits_data.csv”

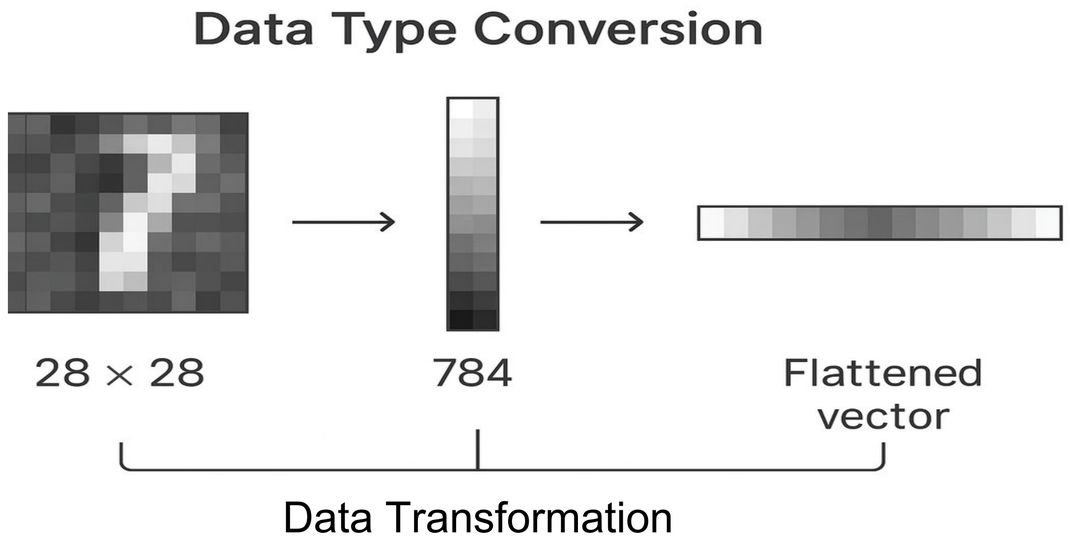


생성된 이미지 시각화

데이터셋 변환

데이터 변환 과정

1. 28x28 픽셀 2D 이미지 생성
2. 2D→ 1D 평탄화
3. ML 알고리즘 입력

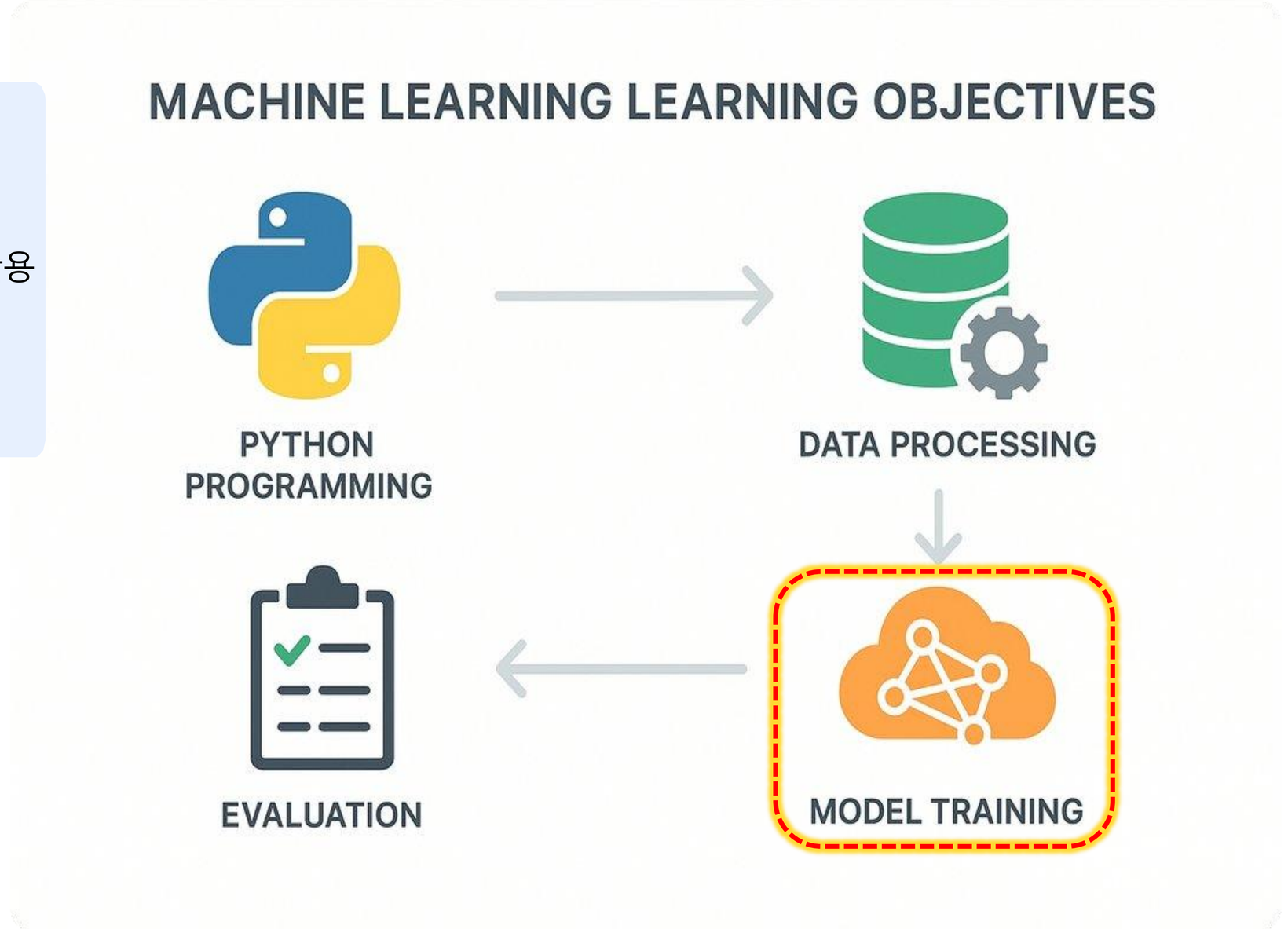


28x28 이미지 행렬에서 784 특성 벡터로의 데이터 변환 과정

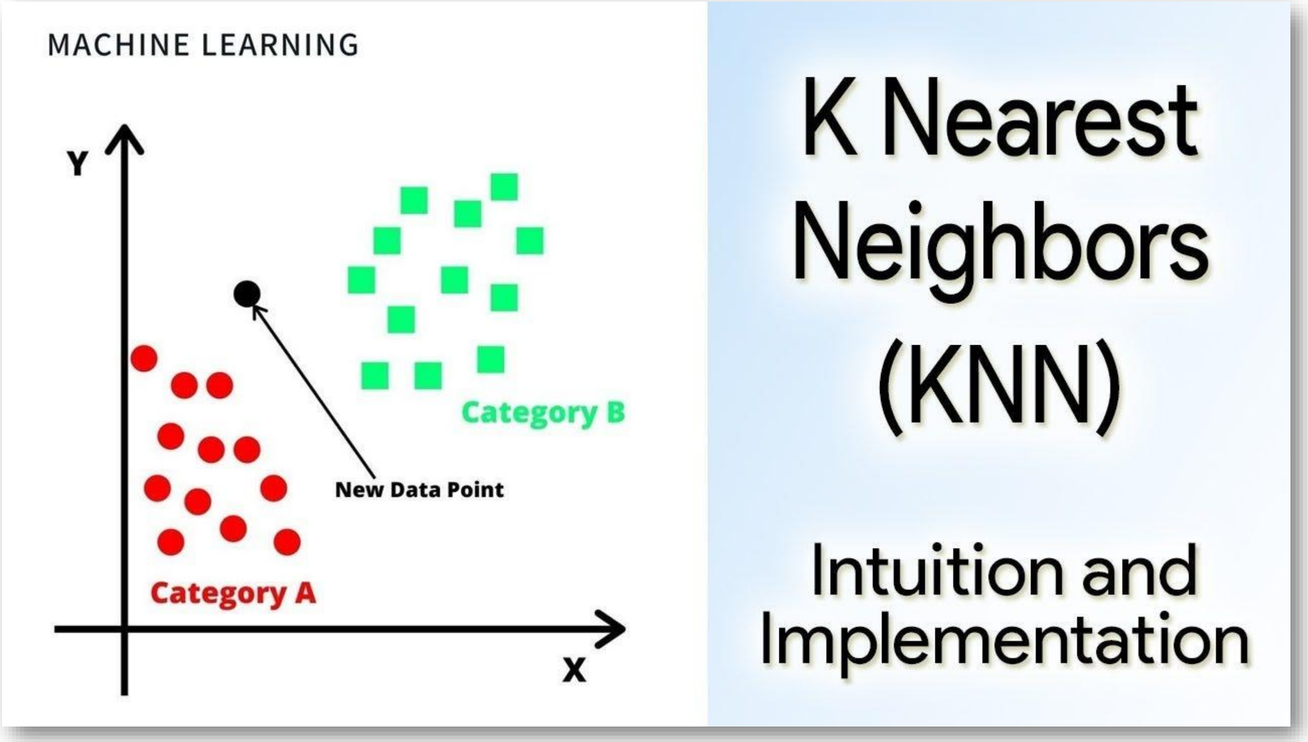
머신러닝(ML) 학습 목표

습득 가능 핵심 역량

1. Python 및 데이터 분석 라이브러리 활용
2. 데이터 생성 및 전처리
3. 머신러닝 알고리즘 구현 및 최적화
4. 모델 평가 및 결과 시각화 능력



KNN(K-Nearest Neighbors) 분류(Classification) 알고리즘



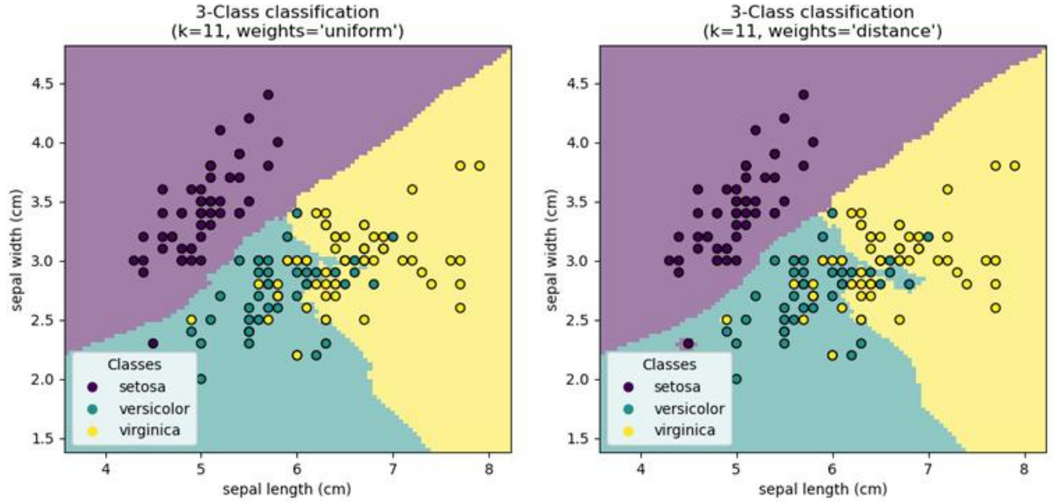
분류(Classification)란?

분류는 데이터를 미리 정의된 범주로 구분하는 지도학습 방법

입력 데이터의 특성(feature)을 기반으로 해당 데이터가 어떤 클래스(class)에 속하는지 예측하는 기법으로, 레이블이 있는 데이터를 사용하여 모델을 훈련합니다.

주요 응용 사례

- **이미지 인식 :** 손글씨 숫자 분류(MNIST), 객체 검출
- **텍스트 분류 :** 스팸 메일 필터링, 감성 분석
- **의료진단 :** 질병 진단, 의료 이미지 분석
- **금융 :** 신용 평가, 사기 거래 감지



분류의 종류: 이진 분류(Binary Classification)와 다중 분류(Multi-class Classification)로 나눌 수 있습니다.

※ MNIST : Modified National Institute of Standards and Technology, 손글씨 숫자 데이터베이스

Scikit-learn 소개 및 분류 알고리즘 개요

Scikit-learn은 파이썬 머신러닝의 표준 라이브러리

간결한 API와 풍부한 알고리즘, 통합된 생태계를 제공하여 데이터 전처리부터 모델 훈련, 평가까지 머신러닝 전체 워크플로우를 지원합니다.

Scikit-learn 패키지 구조

- **preprocessing** – 데이터 정규화, 스케일링
- **model_selection** – 교차검증, 튜닝
- **metrics** – 모델 성능 평가 지표
- **ensemble** – 앙상블 학습 알고리즘
- **neighbors, tree, svm** – 알고리즘 모듈
- **cluster** – 군집화 알고리즘

Scikit-learn의 일관된 API : 모든 알고리즘은 fit(), predict(), transform() 등의 공통 메소드를 제공하여 쉽게 교체하고 비교할 수 있습니다.

주요 분류 알고리즘

다양한 특성을 가진 분류 알고리즘들의 장단점

로지스틱 회귀
선형 모델 기반, 해석력 높음

K-최근접 이웃
거리 기반, 직관적 방법

결정 트리
규칙 기반, 해석 용이

SVM
마진 최대화, 고차원 강함

KNN을 이용한 숫자 분류

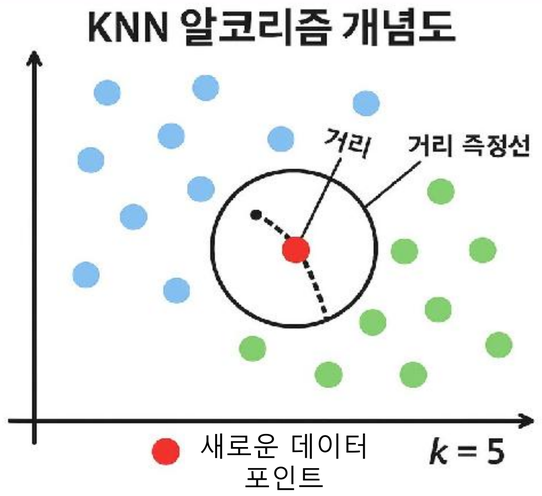
K-최근접 이웃(KNN, K-Nearest Neighbors) 알고리즘 개요

KNN은 새로운 데이터 포인트를 분류할 때 가장 가까운 k개의 이수 데이터를 기반으로 클래스를 결정하는 직관적인 분류 알고리즘입니다.

KNN의 작동 원리

- 새로운 샘플과 가장 가까운 k개의 이웃을 찾습니다.
- k개 이웃들의 다수결 투표로 클래스를 결정합니다.
- k값에 따라 결과가 크게 달라질 수 있습니다.

KNN 알고리즘 시각화

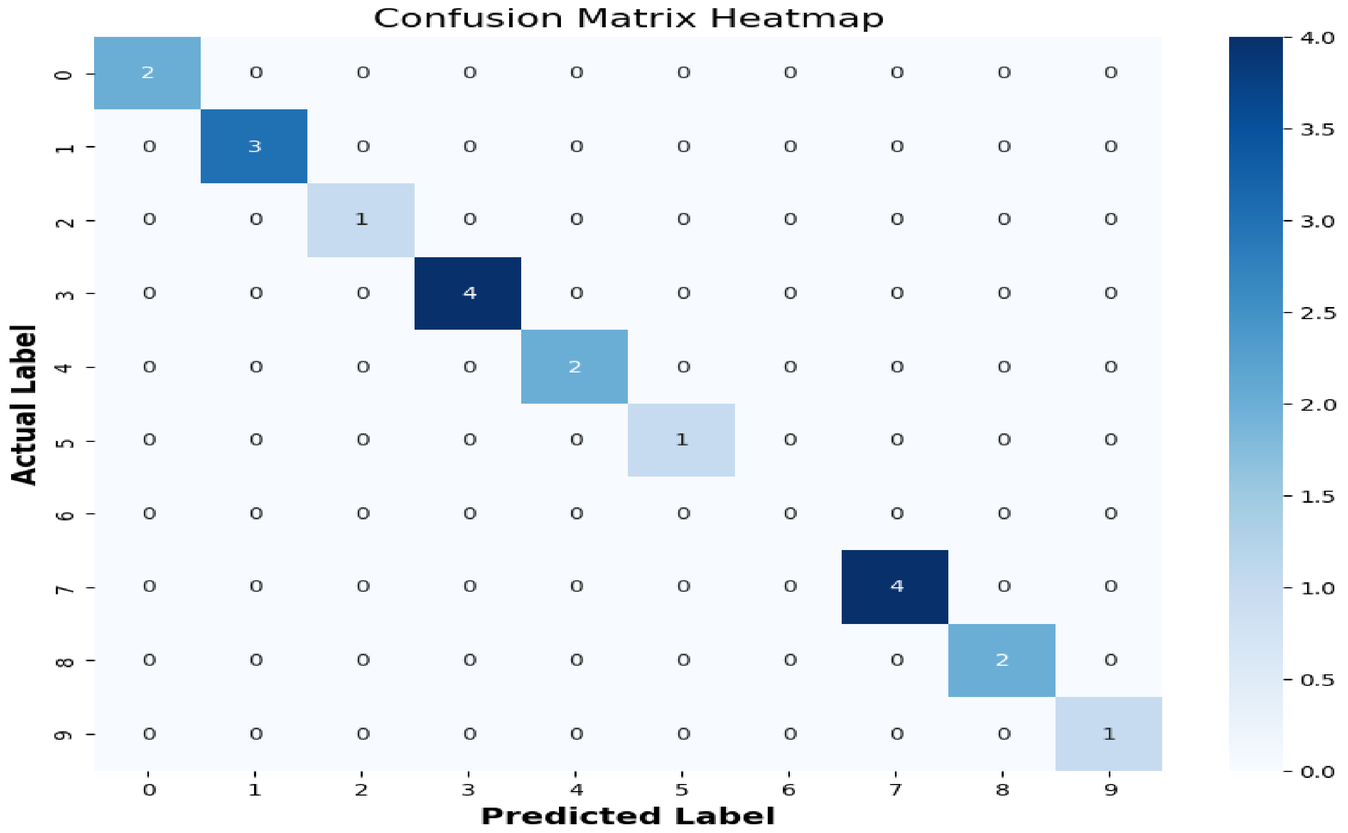


Google Colab ‘실습_KNN’ 참조

분류 모델 평가 및 Confusion Matrix

20X20 Confusion Matrix

10개 숫자(0-9) 분류의 예측 결과를 히트맵으로 시각화.
대각선은 올바를 예측(진한 파란색), 그 외는 오분류(연한 색상)를 나타냅니다.



높은 정확도(대각선) – 올바르게 분류된 숫자

낮은 오분류율 – 거의 혼동되지 않음

높은 오분류율 – 자주 혼동되는 숫자 쌍

분류 모델 평가 및 Confusion Matrix

K값을 5로 설정하였을 때 성능 비교

	정확도 (Accuracy)	정밀도 (Precision)	재현율 (Recall)	F1 Score
K=5	100.0%	100.0%	100.0%	100.0%

숫자 이미지 분류를 위한 다중 클래스 평가

0~9까지 10개 클래스의 분류 성능을 평가하기 위해 다양한 지표와 Confusion Matrix를 활용합니다. Confusion Matrix는 각 숫자별 오분류 패턴을 시각적으로 확인할 수 있습니다.

정확도 (Accuracy)

전체 예측 중 올바르게 분류된 숫자의 비율

Accuracy = 올바르게 분류된 숫자 / 전체 숫자 수

정밀도 (Precision)

특정 숫자로 예측한 것 중 실제로 그 숫자인 비율

Precision = TP / (TP + FP)

재현율 (Recall)

실제 측정 숫자 중 올바르게 예측된 비율

Recall = TP / (TP + FN)

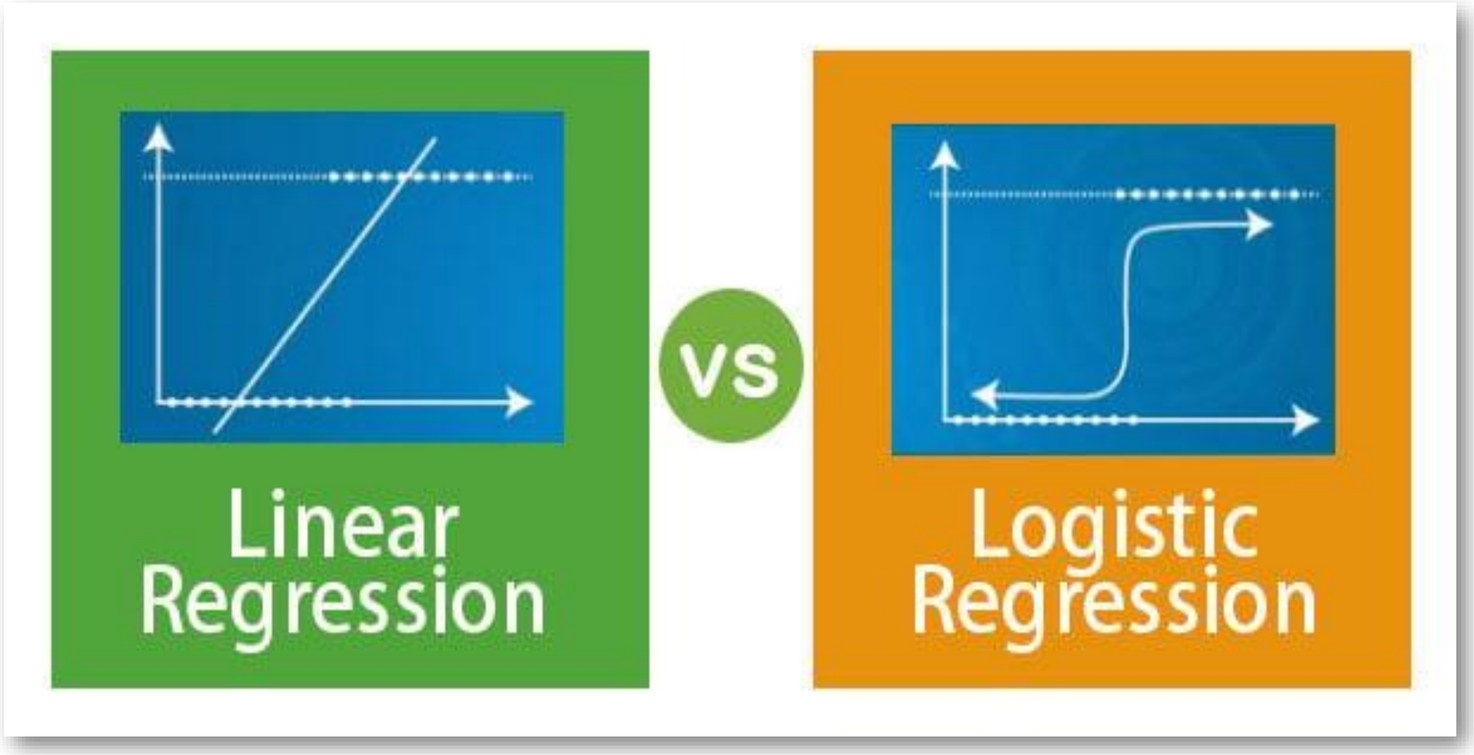
F1 점수 (F1 Score)

정밀도와 재현율의 조화평균

F1 = 2 * (Precision * Recall) / (Precision + Recall)

K값 선택의 중요성: 너무 작은 k는 노이즈에 민감하고, 너무 큰 k는 분류 경계를 흐리게 만듭니다. 최적의 k값은 교차 검증을 통해 결정합니다.

선형 회귀(Regression) 알고리즘



회귀(Regression)란?

회귀는 연속적인 값을 예측하는 지도학습 방법

입력 특성(feature)과 출력 값(target) 사이의 관계를 모델링하여 새로운 입력에 대한 출력 값을 예측합니다.

숫자 이미지에서의 회귀 예시

- **픽셀 강도 합 :** 이미지 픽셀 값 합계 예측
- **필기체 숫자 기울기 :** 숫자의 각도 예측
- **숫자의 두께 :** 필기체 선 두께 측정
- **숫자 가치 예측 :** 숫자 값 기반 가치 예측

항목	분류(Classification)	회귀(Regression)	군집화(Clustering)
출력	이산 클래스(범주)	연속 값(실수)	그룹 레이블
학습	지도 학습	지도 학습	비지도 학습
예시	숫자가 '3'인지 판별	픽셀로 어떤 숫자인지 예측	비슷한 형태 그룹화

회귀 모델의 수학적 표현

일반적인 선형 회귀 모델

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$$

여기서 β_i는 가중치, X_i는 특성, ε는 오차항

회귀 모델 종류

- 선형 회귀
- 다항 회귀
- 릿지/라쏘 회귀
- 결정트리 회귀
- 랜덤포레스트 회귀

선형 회귀 기초 이론

선형 회귀(Linear Regression)의 기본 개념

선형 회귀는 종속 변수 y와 독립 변수 X 간의 선형 관계를 모델링하는 기법으로, 머신러닝에서 가장 기본적인 회귀 알고리즘입니다.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$$

- y: 종속변수
- $X_1, X_2, \cdots, X_n$: 독립 변수들
- $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$: 회귀 계수
- ε: 오차항

머신러닝에서의 활용

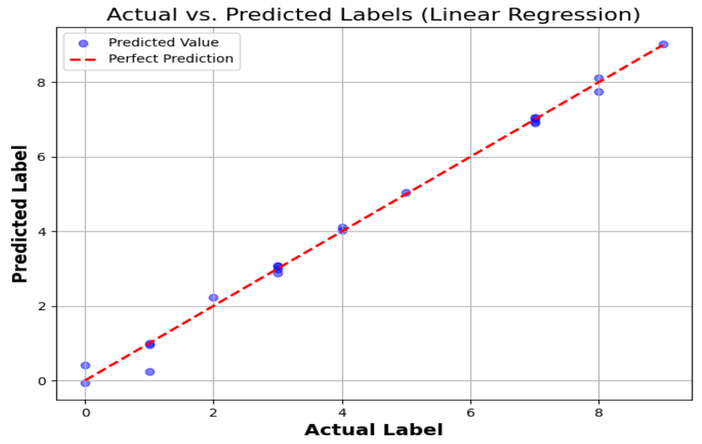
- 숫자 이미지의 픽셀 값으로 특정 속성 예측
- 선형 관계가 있는 데이터에 대한 예측 모델 구축
- 특성 중요도(가중치)를 해석하여 데이터 이해

Scikit-learn 구현

Scikit-learn에서는 선형 회귀를 쉽게 구현할 수 있는 다양한 클래스를 제공합니다.

```
# scikit-learn 선형회귀 예제
from sklearn.linear_model import LinearRegression

def train_model(X, y): model = LinearRegression()
model.fit(X, y)
# 학습된 계수 확인
intercept = model.intercept_
coefficients = model.coef_
```



선형 회귀 실습: 숫자 이미지의 합/특징 예측

숫자 이미지 데이터에 선형 회귀(Linear Regression) 적용하기

숫자 이미지의 픽셀 값을 특징으로 사용하여 타겟 변수(예 : intensity 합계, 숫자 굵기 등)를 예측하는 회귀 모델 구현

타겟 변수 선정

- 전체 intensity 합 : 픽셀 값 총합
- 평균 픽셀 값 : 숫자 영역의 평균 밝기
- 숫자 영역 비율 : 숫자 면적 비율

구현 과정

- 이미지 데이터를 1차원 벡터로 변환
- 타겟 변수 계산 (픽셀 합계 등)
- 데이터셋 분할 (학습/테스트)
- 선형 회귀 모델 학습 및 예측

핵심 포인트 : 숫자 이미지에 선형 회귀를 적용하면 숫자의 특성과 픽셀 분포 사이의 선형 관계를 파악할 수 있습니다.

모델 선택 TIP:
모델 선택 시 성능뿐만 아니라 복잡도, 학습 시간, 해석 가능성 등 다양한 측면을 종합적으로 고려해야 합니다.

Google Colab ‘실습_Regression’ 참조

회귀 모델 평가 및 시각화

회귀 모델의 성능 평가와 비교 분석

다양한 회귀 모델의 성능을 체계적으로 비교 평가하고, 이를 시각화하여 각 모델의 강점과 약점을 분석할 수 있습니다.

주요 평가 지표

MSE (평균 제곱 오차)

예측값과 실제값 차이의 제곱 평균으로, 오차의 크기를 강조합니다. 값이 작을수록 좋습니다.

$$MSE = \frac{1}{n}\sum(y_{실제} - y_{예측})^2$$

R^2 (결정계수)

모델이 설명하는 분산의 비율로, 1에 가까울수록 예측 성능이 우수합니다.

$R^2 = 1 - (오차\ 제곱합 / 총\ 제곱합)$

RMSE (평균 제곱근 오차)

MSE의 제곱근으로, 원본 단위의 오차를 나타내어 해석이 용이합니다.

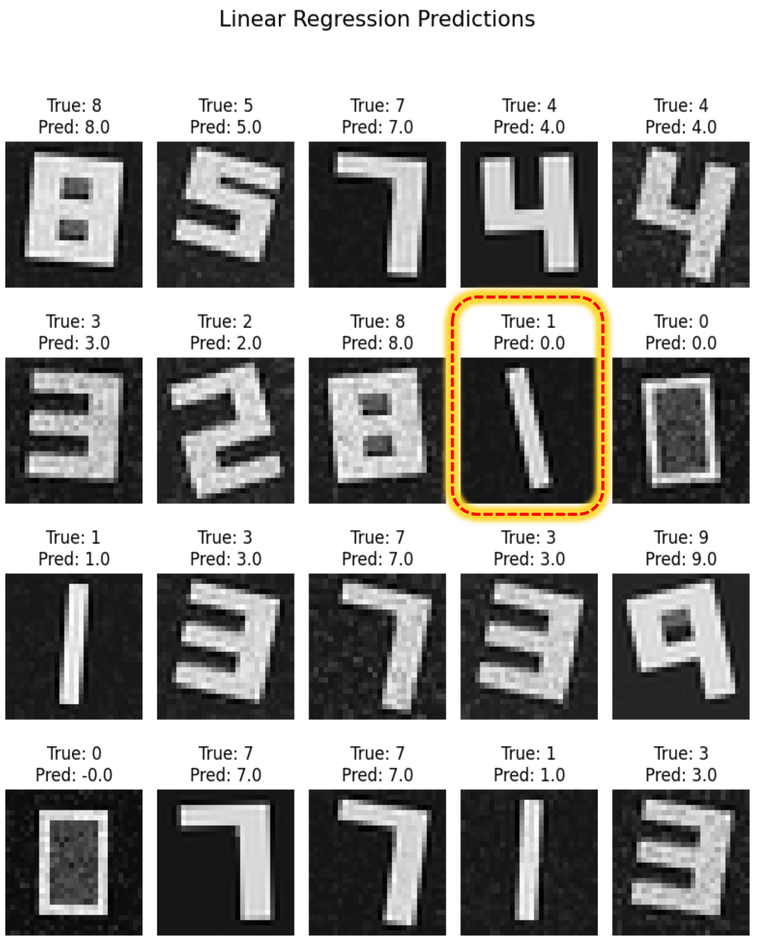
MAE (평균 절대 오차)

예측값과 실제값 차이의 절대값 평균으로, 이상치에 덜 민감합니다.

MSE: 0.05 | R² 점수: 0.99 | RMSE: 0.22 | MAE: 0.13

모델 해석

R^2값이 0.99로 모델이 타겟 변수 변동의 99%를 설명합니다. 데이터 포인트들이 $y = X$ 직선 주변에 밀집되어 예측 정확도가 높음을 보여줍니다.

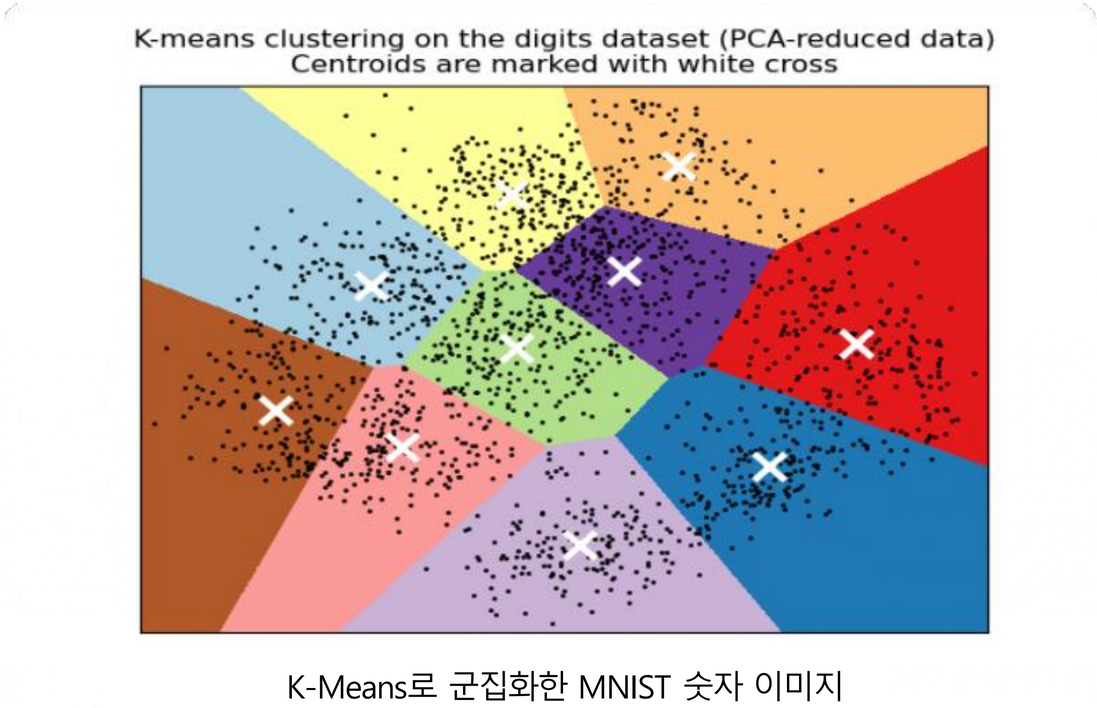


K-평균(K-Means) 군집화(Clustering) 알고리즘

군집화(Clustering)란?

군집화는 비지도학습의 대표적인 기법

레이블이나 정답 없이 데이터의 패턴과 유사성을 기반으로 비슷한 특성을 가진 데이터를 그룹화하는 기법으로, 데이터 내부의 구조를 발견하는데 활용됩니다.



K-Means로 군집화한 MNIST 숫자 이미지

군집화 vs. 분류 : 분류는 미리 정의된 클래스에 데이터를 할당하는 지도학습이지만, 군집화는 데이터 자체의 유사성만으로 그룹을 형성하는 비지도 학습입니다.

군집화의 주요 특징

- **비지도 학습:** 정답(레이블)이 없는 상태에서 데이터 간의 유사성만으로 군집 형성
- **자동 그룹화:** 데이터의 내재적 특성에 따라 자동으로 그룹을 형성
- **탐색적 분석:** 정데이터에 숨겨진 패턴과 구조를 발견하는데 유용
- **차원 축소:** 복잡한 데이터를 의미 있는 그룹으로 단순화

숫자 이미지 군집화의 활용

- 레이블 없이 숫자 이미지의 유사성을 분석하여 자동으로 0~9 그룹으로 분류
- 비슷한 필기체 스타일 또는 특성을 가진 숫자들을 묶어 패턴 발견
- 이상치(특이한 필기체) 탐지를 통한 데이터 품질 관리

K-Means 군집화 알고리즘

주요 군집화 알고리즘

- **K-Means :** 중심 기반 군집화
- **계층적 군집화 :** 상향식/하향식 병합
- **DBSCAN :** 밀도 기반 군집화
- **GMM :** 확률 분포 기반 군집화

K-Means: 가장 널리 사용되는 분할 군집화 알고리즘

데이터를 K개의 클러스터로 나누어 각 클러스터의 중심(Centroid)과 데이터 포인트 간의 거리 제곱 합을 최소화하는 알고리즘입니다.

K-Means 알고리즘 작동 원리

1 초기화

K개의 초기 중심점을 무작위로 선택

2 할당단계

각 데이터 포인트를 가장 가까운 centroid의 클러스터에 할당

3 업데이터 단계

각 클러스터 내 데이터의 평균으로 centroid 재계산

4 수렴

Centroid가 더 이상 크게 변하지 않을 때까지 2~3단계 반복

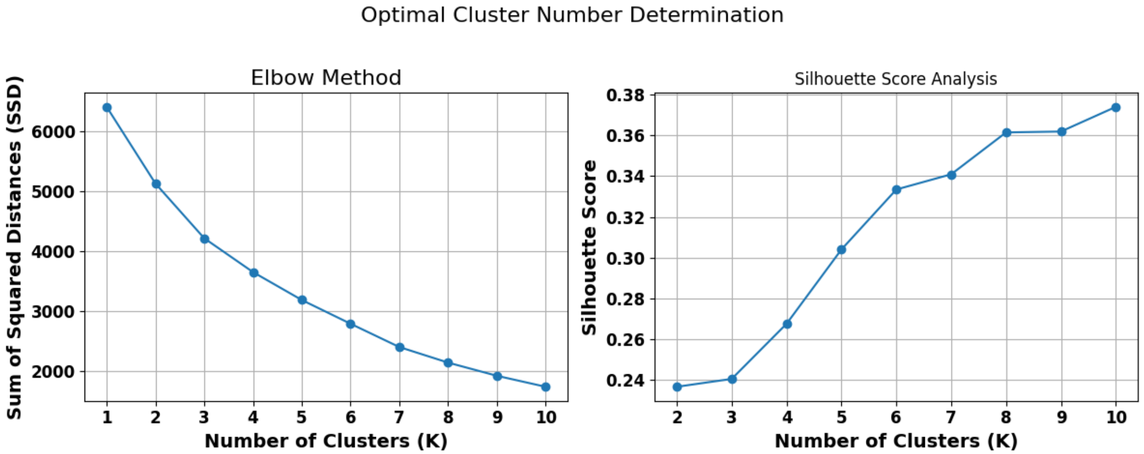
※ DBSCAN : Density-Based Spatial Clustering of Applications with Noise
GMM : Gaussian Mixture Model

K-Means 군집화 알고리즘

최적의 클러스터 수(K) 선정 방법

Elbow Method: WCSS 그래프에서 "팔꿈치" 지점 찾기

실루엣 분석: 클러스터 내 응집도와 분리도 측정



최적 클러스터 수 결정을 위한 실루엣 분석

$$\textbf{Sum of Squared Distances} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - u_k\|^2$$

- x_i : i번째 데이터 포인트
- u_k : k번째 클러스터의 기준점

$$\textbf{Silhouette Coefficient } (S_i) = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- a_i는 점 i에서 같은 클러스터 내 다른 모든 점까지의 평균 거리
- b_i는 점 i에서 가장 가까운 이웃 클러스터 내 모든 점까지의 평균 거리

K-Means의 한계 : 초기 중심점에 민감하고 구형 클러스터만 잘 찾으며, 클러스터 크기가 다양한 경우 성능이 저하됩니다.

Google Colab ‘실습_Kmeans’ 참조

K-Means 군집화 알고리즘 실습

K-Means로 숫자 데이터 자동 그룹화 실습

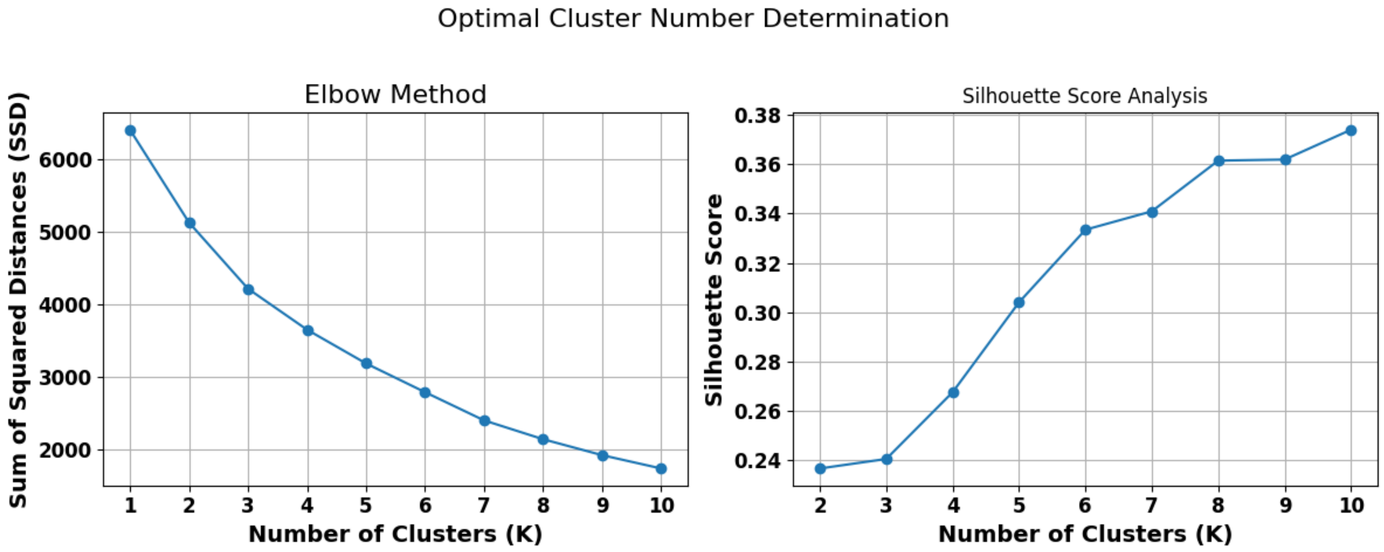
생성된 100개의 숫자 이미지 데이터를 K-Means 알고리즘으로
군집화하여 유사한 이미지들이 자동으로 그룹화되는 과정을 구현합니다.

Scikit-learn 구현

Scikit-learn을 사용하면 몇 줄의 코드만으로 K-Means 군집화를 수행할 수 있습니다.

군집화 성능 평가 지표

분류와 달리 군집화는 명확한 정답이 없어 다양한 내부/외부 지표로 평가합니다.



최적 클러스터 수 결정을 위한 실루엣 분석

군집화 결과 종합 시각화

군집화 알고리즘 결과 시각화

28 x 28 픽셀(784차원) 숫자 이미지 데이터를 데이터 평탄화를 통해 1차원으로 변환하여 K-Means 알고리즘의 군집화 결과를 분석합니다.

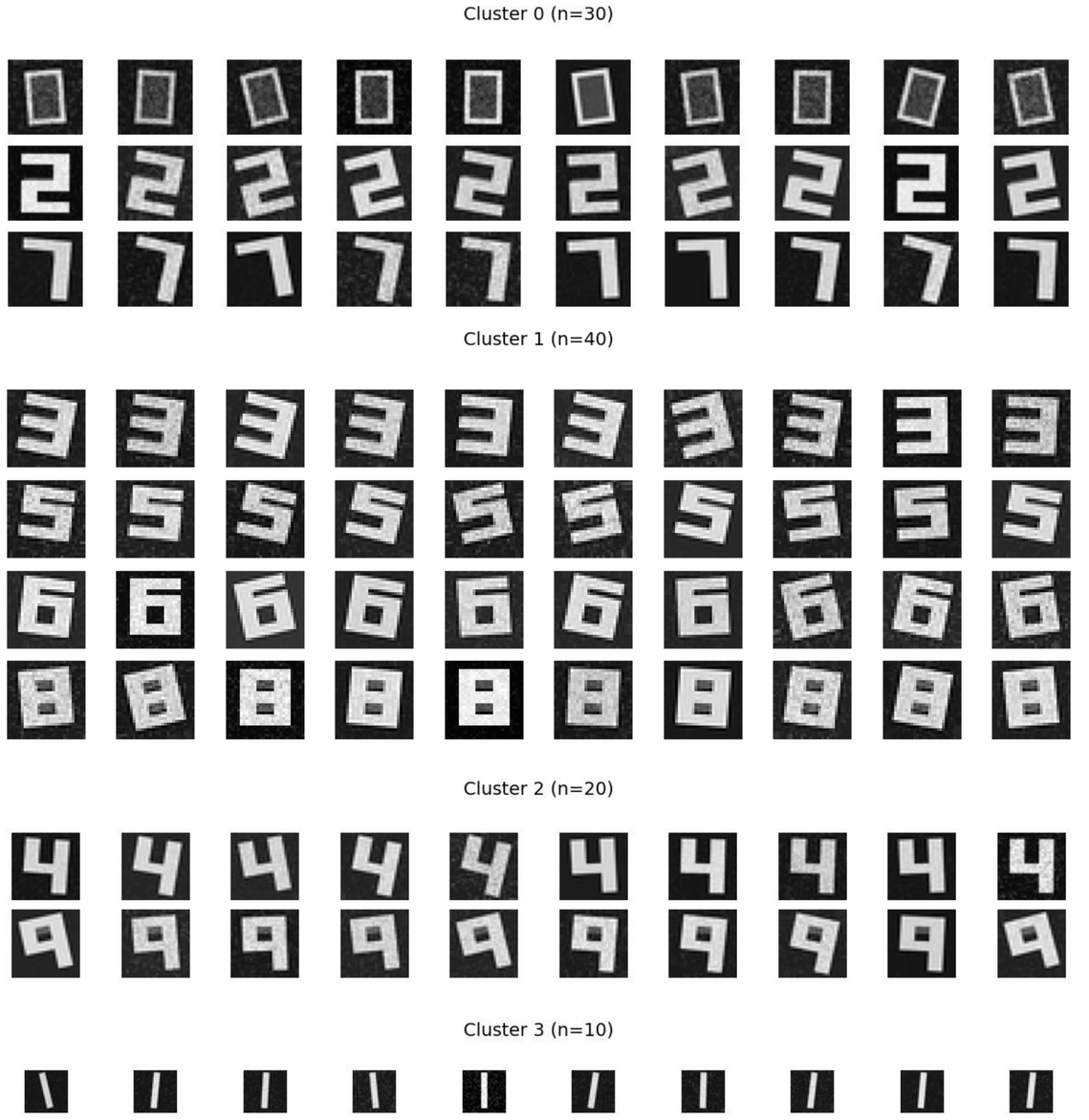
K-Means

중심점 기반 군집화로 클러스터가 원형 모형을 형성.

중심점이 클러스터내 데이터 중심(★)에 위치.

클러스터별 특성 요약

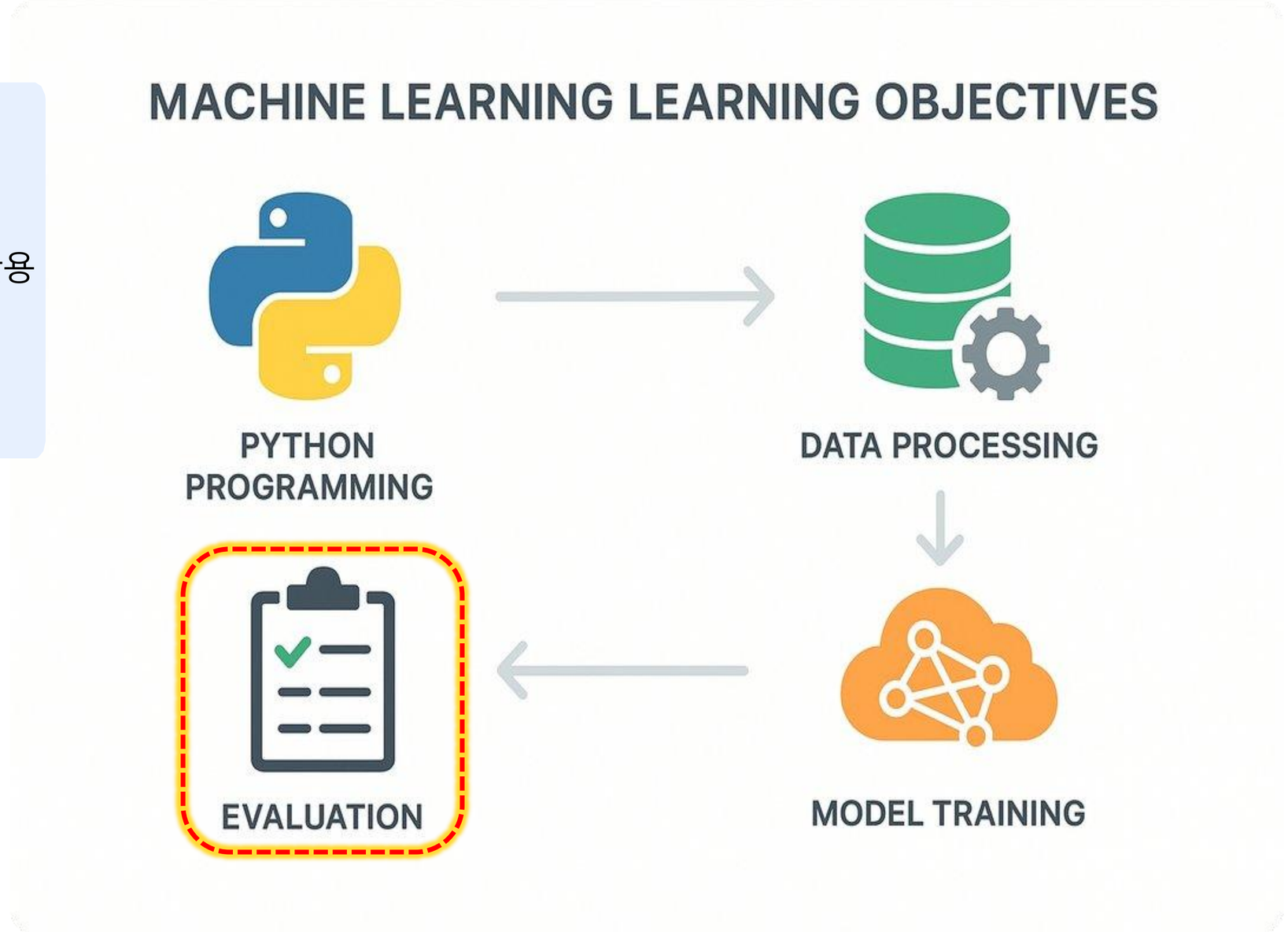
클러스터	샘플 수	주요 숫자	특징
클러스터 0	30	0, 2, 8	둥근 형태의 숫자들
클러스터 1	40	3, 5 ,6 ,8	굴곡이 많은 숫자들
클러스터 2	20	4, 9	위쪽이 열린 형태의 숫자들
클러스터 3	10	1	직선 획이 많은 숫자들



머신러닝(ML) 학습 목표

습득 가능 핵심 역량

1. Python 및 데이터 분석 라이브러리 활용
2. 데이터 생성 및 전처리
3. 머신러닝 알고리즘 구현 및 최적화
4. 모델 평가 및 결과 시각화 능력



생성형 AI를 활용한 보고서 초안 작성

좋은 프롬프트 작성 원칙

1	명확성 (Clarity)	무엇을 원하는지 명확히
2	구체성 (Specificity)	형식·분량·톤을 제시
3	맥락 (Context)	대상, 목적, 배경 포함
4	역할 (Role)	“너는 전문가/교사/데이터 분석가야” 등 지정
5	반복 개선 (Iteration)	피드백을 통해 점진적 수정

프롬프트 실습 예시

목적	나쁜 예시	좋은 예시
요약	“이 글 요약해줘”	“대학생 수준으로 5문장 내 요약해줘”
코드	“파이썬 코드 짜줘”	“sklearn 사용하여 선형회귀 예제 만들어줘”
이미지	“고양이 그림 그려줘”	“파란 배경에 앉아 있는 고양이 사진 스타일로 그려줘”

과제 개요

과제 목표

본 과제는 실습을 통해 학습한 지도학습(Supervised Learning)과 비지도학습(Unsupervised Learning) 알고리즘의 원리를 깊이 이해하고, 실제 데이터에 적용한 결과를 분석하여 보고서로 정리하는 것을 목표로 합니다. 특히, **생성형 AI** (예: ChatGPT, Google-Gemini, Genspark 등) 도구를 활용하여 기술적인 내용을 효과적으로 문서화하는 경험을 습득합니다.

과제 내용

강의에서 진행한 100개의 숫자 이미지(0-9까지 각 10개) 생성 및 실습 결과를 바탕으로 아래의 내용을 포함한 최종 보고서를 제출해야 합니다.

제출물

생성형 AI (예: ChatGPT, Google-Gemini, Genspark 등)를 활용하여 실습 결과를 정리한 최종 보고서 초안을 작성하세요.

- 보고서 형식: **PDF 파일 (총 5장 이내)**

보고서 초안 작성

생성형 AI를 활용하여 보고서 초안 작성

강의에서 진행한 100개의 숫자 이미지(0-9까지 각 10개) 생성 및 실습 결과를 바탕으로 최종 보고서 초안을 작성해보세요.

No.	항목	설명	본 요구사항의 적용 예시
1	**명확성 (Clarity)**	AI가 해야 할 작업을 모호하지 않게, 핵심 목표를 명확히 제시	"숫자 0~9 각각을 28x28 픽셀 이미지로 생성하고, KNN, Linear Regression, K-means를 각각 적용하여 분류·회귀·군집화 수행 결과를 포함한 5쪽짜리 보고서 작성"처럼 **최종 산출물과 목적**을 명시
2	**구체성 (Specificity)**	출력 형식·분량·언어·구성 등을 구체적으로 지시	"PDF 형식으로 저장", "한글로 작성", "5장 분량", "결과 그래프 및 이론 설명 포함" 등 **형식과 범위 명시**
3	**맥락 (Context)**	작업 배경, 목적, 사용 도구 등 상황 정보를 함께 제공	"Colab에서 MNIST 유사 데이터(28x28)를 직접 생성 후 KNN·회귀·군집화 수행", "분류·회귀·비지도학습 개념을 함께 설명" 등 **보고서의 학습적 의도**를 부연
4	**역할 (Role)**	AI에게 수행할 역할을 부여해 산출물의 톤과 수준을 조정	"너는 머신러닝 기초 강의를 수강 중인 학생으로서, 일반인 학습자로서 이해한 내용을 바탕으로 쉽게 설명하는 보고서를 작성하라."
5	**반복 개선 (Iteration)**	초안 생성 후 수정 방향이나 평가 기준을 함께 제시	"초안을 생성한 후, 챕터별 내용 일관성 및 시각자료 배치 여부를 검토하여 수정 버전을 다시 생성하라." 또는 "결과 요약 부분을 더 짧게 만들어 달라."와 같은 **피드백 루프 설정**

사용가능한 생성형 AI 서비스

기업	대표 모델	특징
OpenAI	GPT-5	생성형 AI 선두주자
Google DeepMind	Gemini (구 Bard)	LLM과 강화학습 선도
Genspark	**Super Agent** **(Mixture-of-Agents 기반)**	다중 **LLM** 오케스트레이션 / 자율형 **AI** 에이전트 플랫폼

결과물 (GPT-5)

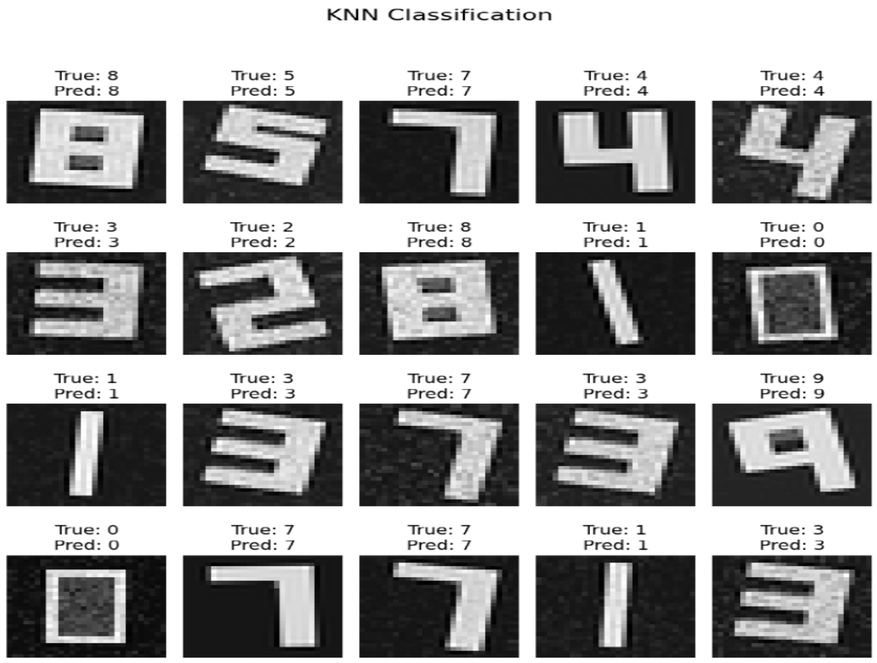
■■■■■ ■■■■■■ ■■ ■■■

■ 1 ■ ■■

■ ■■■■■ ■■ 0■■ 9■■■■ ■ ■■■■ 10■■, ■ 100■■ 28x28 ■■ ■■■ ■■■■■ ■■■■■ ■■■■■ ■■■ ■■■ ■■■ ■■■■ ■■■■■. ■ ■■■■■■ ■■■■■ ■■■■■ K-■■■■ ■■(K-Nearest Neighbors, KNN) ■■■ ■■ ■■(Linear Regression)■ ■■■■■■■, ■■■■■■ ■■■■■ K-■■(K-Means) ■■■■■ ■■■■■■■■. ■■ ■■ ■■■■■■ ■■■■■■■ ■■■ ■■■ ■■■■■, ■ ■■■ ■■■ ■■■ ■■■■■■ ■■■.

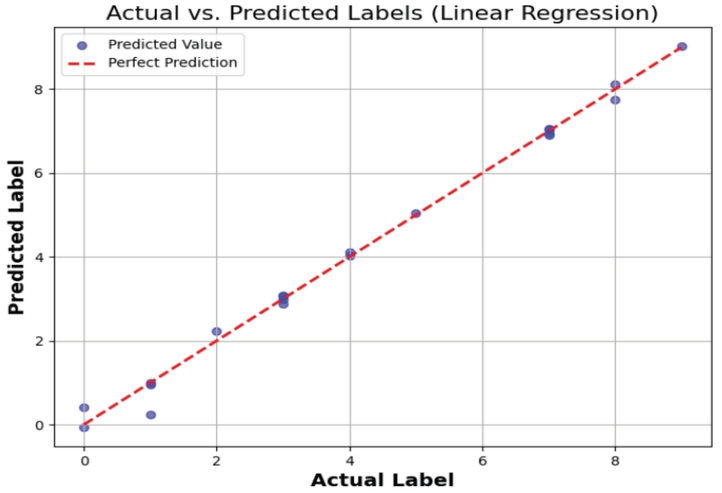
■ 2 ■ KNN ■■

KNN■ ■■■■■ ■■■■■ ■■■■■■■, ■■■ ■■■■■ ■■ ■■■ ■■■■■ ■■■ ■■ ■■■■■ ■■■■■■. ■ ■■■■■■ ■■ ■■■ ■■■■■ ■■■■■ KNN■ ■■■■■ ■■■ ■■ ■■■ ■■■■■■■■. ■■ ■■■ KNN■ ■■■ ■■ ■■■ ■■ ■■(Confusion Matrix)■ ■■ ■■■ ■■■■■■ ■■■ ■■■■■.



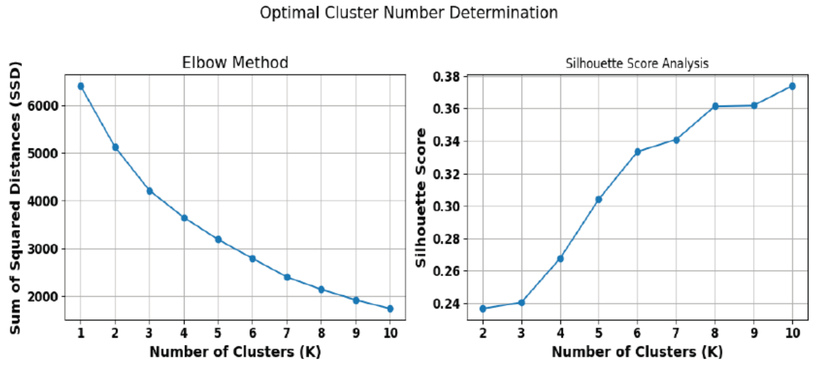
■ 3 ■ ■■ ■■

■■ ■■■■ ■■■■■ ■■ ■■■■■ ■■■■■ ■■■■■, ■■ ■■■■■ ■■ ■ ■■ ■■■■ ■■■■ ■■■■■■■. ■ ■■■■■■ ■■ ■■■■■ ■■■■ ■■■■■ ■■■■ ■■■■■■ ■■■■■ ■■■■ ■■■■■■■■■. ■■■■ ■■ ■■ ■■■■ ■■■■■■ ■■■■ ■■■■■■■ ■■ ■■■■ ■■ ■■■■■ ■■■■■ ■■■■ ■■■■■■. ■■ ■■■■ ■■ ■■ ■■ ■■ ■■■ ■■■■ ■■■■■ ■■■■ ■■ ■■■■ ■■■■■■.



■ 4 ■ K-Means ■■■

K-Means■ ■■■■■ ■■■■■■ ■■■■■■■, ■■■■■ ■■ ■■■■ ■■■■■ ■■ ■■■■ ■■ ■■■■■ ■■■■■ ■■■■ ■■■■■■. ■ ■■■■■■ Elbow Method■ ■■■■ ■■(Silhouette Score)■ ■■■■■ ■■■■ ■■ ■■ ■■■■■■■, ■ ■■ ■■ ■■■■■■ ■■■■■■ ■■■■ ■■■■ ■■ ■■■■■■ ■■ ■■■■ ■ ■■■■■■.



지도학습과 비지도학습: 손글씨 이미지 데이터 분석 사례를 중심으로

제 1 장: 서론 및 머신러닝 학습 패러다임의 이해

1.1 보고서 목적 및 개요

본 보고서는 숫자 0 부터 9 까지의 손글씨 이미지 데이터에 지도학습과 비지도학습의 대표 알고리즘인 KNN, 선형 회귀(Linear Regression), 그리고 K-평균(K-means)을 각각 적용한 결과를 심층적으로 분석합니다. 이 분석을 통해 각 알고리즘의 이론적 배경과 작동 원리를 명확히 하고, 실제 데이터에 대한 적용 적합성을 비판적으로 검토합니다. 특히, 범주형 분류 문제에 부적합한 회귀 모델의 적용 사례를 통해 올바른 문제 해결을 위한 모델 선택의 중요성을 강조합니다. 이 보고서는 단순한 결과 나열을 넘어, 독자들에게 머신러닝 모델의 본질적 특성에 대한 깊이 있는 통찰을 제공하는 데 중점을 둡니다.

1.2 머신러닝의 양대 산맥: 지도학습과 비지도학습

머신러닝은 크게 지도학습과 비지도학습이라는 두 가지 주요 패러다임으로 나뉩니다. 이 두 방식의 근본적인 차이는 모델 훈련에 사용되는 데이터의 특성에 있습니다.

지도학습(Supervised Learning)은 '정답' 또는 '라벨'이 포함된 데이터셋을 사용합니다. 모델은 입력 데이터와 그에 상응하는 정답 라벨 간의 관계를 학습하여, 새로운 데이터에 대한 예측을 수행하는 것을 목적으로 합니다. 이러한 예측은 두 가지 주요 기법으로 구분됩니다. 첫째, **분류(Classification)**는 이메일 스팸 여부 감지, 감정 분석, 이미지에 나타난 객체 분류와 같이 데이터를 사전에 정의된 이산적인 범주로 나누는 작업입니다. 둘째, **회귀(Regression)**는 주택 가격 예측, 날씨 예측, 수입-지출 관계 분석과 같이 연속적인 값을 예측하는 데 사용됩니다. 지도학습은 명확한 목표를 가지고 있기 때문에, 모델의 성능을 정량적으로 평가하기 용이하다는 장점이 있습니다. 그러나 모델을 훈련시키기 위해 대량의 데이터에 일일이 정답 라벨을 지정해야 하므로, 이 과정에 많은 시간, 비용, 그리고 전문 인력이 소요된다는 단점이 존재합니다.

비지도학습(Unsupervised Learning)은 이와 대조적으로, 정답 라벨이 없는 데이터셋을 활용합니다. 모델은 데이터 자체에 내재된 숨겨진 패턴, 구조, 그리고 관계를 스스로 발견하도록

4.2 핵심 통찰 및 결론

본 분석은 지도학습과 비지도학습의 근본적인 차이를 극명하게 보여줍니다. 첫째, 모델 선택의 중요성은 손글씨 숫자 분류에 선형 회귀를 적용한 사례에서 가장 명확하게 드러났습니다. 데이터에 정답이 있는지 여부는 단순히 학습 방식을 결정하는 것을 넘어, '예측'과 '탐색'이라는 모델의 궁극적인 목적을 결정합니다. 선형 회귀는 연속적인 값을 예측하는 목적에 충실했으나, 이산적인 범주를 다루는 손글씨 분류 문제에는 근본적으로 부적합했습니다. 이처럼 올바른 모델을 선택하는 것은 단순히 기술적인 정확도를 넘어, 문제 해결 자체의 타당성을 결정하는 핵심적인 과정입니다.

둘째, **두 개의 'K'**는 각 학습 패러다임이 데이터를 바라보는 관점의 차이를 상징합니다. KNN 의 K(이웃의 수)는 '정답을 아는' 상태에서 새로운 데이터가 속할 범주를 결정하기 위해 몇 개의 주변 참고 사례를 볼 것인지에 대한 매개변수입니다. 반면, K-평균의 K(군집의 수)는 '정답을 모르는' 상태에서 데이터의 내재된 구조가 몇 개의 그룹으로 이루어져 있을지 가설을 세우는 데 사용되는 매개변수입니다. 두 매개변수의 이름은 같지만, 그 의미와 역할은 각 학습 패러다임의 철학을 반영합니다.

셋째, 라벨의 가치입니다. 지도학습의 높은 정확도와 예측력은 데이터 라벨링이라는 값비싼 대가를 치러 얻은 결과입니다. 반면, K-평균은 라벨 없이도 데이터에 숨겨진 시각적 유사성 패턴을 발견하는 통찰을 제공하며, 이는 데이터 탐색의 첫 단계로 매우 유용합니다. 이 두 패러다임은 서로 상반된 방식으로 데이터의 가치를 극대화합니다.

4.3 실용적 제언 및 향후 연구 방향

본 분석 결과를 바탕으로 다음과 같은 실용적인 제언을 제시할 수 있습니다.

* 더 적합한 모델 활용: 손글씨 이미지와 같이 픽셀의 공간적 구조가 중요한 분류 문제에는 픽셀의 공간적 관계를 효과적으로 학습하는 **합성곱 신경망(Convolutional Neural Network, CNN)**과 같은 딥러닝 모델이 훨씬 더 뛰어난 성능을 보입니다.

* K-평균의 전략적 활용: K-평균은 단독적인 분류 모델로 사용하기보다, 데이터의 잠재적인 패턴을 파악하는 탐색적 데이터 분석 단계나, 정상 범주에서 벗어난 데이터를 찾아내는 이상치(Outlier) 탐지에 활용하는 것이 더욱 실용적입니다.

결과물 (Genspark)

제1장. 서론

1.1 연구 배경

머신러닝은 컴퓨터가 명시적으로 프로그래밍되지 않고도 데이터로부터 학습할 수 있게 하는 인공지능의 한 분야입니다. 특히 패턴 인식과 분류 문제에서 뛰어난 성능을 보여주며, 현대 사회의 다양한 분야에서 활용되고 있습니다.

1.2 연구 목적

본 연구는 지도학습과 비지도학습의 대표적인 알고리즘들을 손글씨 숫자 데이터셋을 이용하여 비교 분석하고, 각 알고리즘의 특성과 성능을 평가하는 것을 목적으로 합니다.

1.3 연구 방법

28×28 픽셀의 손글씨 숫자 이미지(0-9)를 생성하고, KNN(K-Nearest Neighbors), Linear Regression, K-means 클러스터링 알고리즘을 적용하여 분류, 회귀, 군집화 성능을 분석합니다.

1.4 기대효과

이 연구를 통해 지도학습과 비지도학습의 차이점을 명확히 이해하고, 실제 데이터에서 각 알고리즘의 적용 가능성을 평가할 수 있을 것입니다.

제4장. 실험 결과 및 분석

4.1 KNN 분류 결과

KNN 알고리즘의 성능을 다양한 K값에 대해 평가한 결과, K=1일 때 최고 정확도 0.550을 달성했습니다.

K값별 성능 분석:

- K=1: 0.550
- K=3: 0.350
- K=5: 0.350
- K=7: 0.250
- K=9: 0.250

K=1에서 가장 높은 성능을 보였으며, K값이 증가할수록 성능이 감소하는 경향을 보였습니다. 이는 작은 데이터셋에서 과적합이 오히려 도움이 되는 경우를 보여줍니다.

4.2 Linear Regression 결과

Linear Regression을 이용한 숫자 값 예측 결과:

- 평균제곱오차(MSE): 4.476
- 평균제곱근오차(RMSE): 2.116
- 결정계수(R^2): 0.458
- 분류 정확도(반올림 후): 0.400

선형 회귀는 연속값 예측 모델이지만, 예측값을 반올림하여 분류 성능으로도 평가했습니다. 비선형적인 이미지 데이터의 특성상 제한적인 성능을 보였습니다.

4.3 K-means 군집화 결과

K-means 클러스터링을 10개 클러스터로 수행한 결과:

- 최종 관성(Inertia): 52055.13
- 군집화 순도: 0.270

클러스터별 분석 결과, 일부 클러스터는 특정 숫자에 특화되어 있으나, 전반적으로는 숫자별 명확한 분리가 어려운 것으로 나타났습니다.

4.4 알고리즘 비교 분석

세 알고리즘의 성능 비교:

- KNN: 0.550 (분류 정확도)

감사합니다

hplee@hongik.ac.kr

Scan to access the homepage