



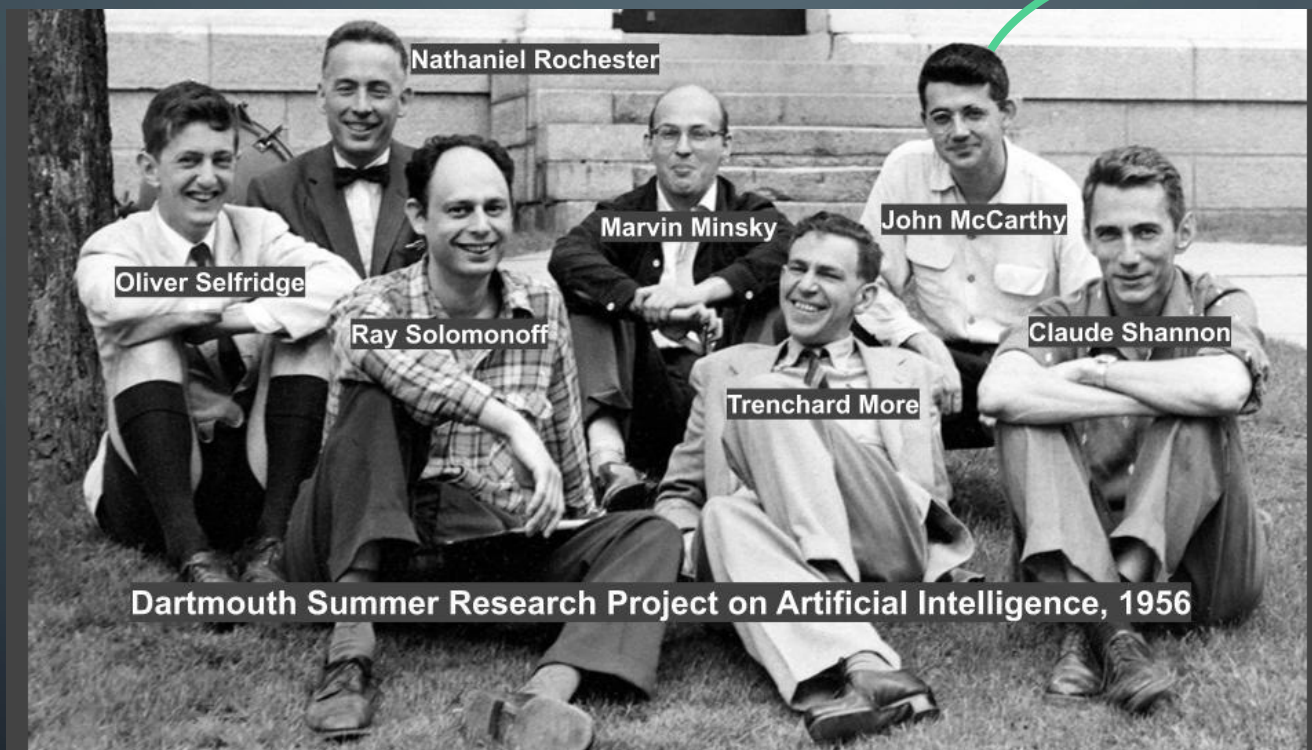
인공지능이 실현하는 4차산업혁명

전남대학교 소프트웨어중심대학사업단

유재명

인공지능 연구

- 1956년 여름, 미국 다트머스 컨퍼런스
- 의의: "AI가 연구 분야로 공식 출범하는 계기 마련 "



‘인공지능’ 개념 도입하여
지능형 기계를 만드는
과학과 공학 정의

LISP 언어 개발 – AI 연구
기본 언어로 채택

"우리는 1956년 여름에 다트머스 대학에서 2개월 동안
10명으로 구성된 인공 지능 연구를 수행할 것을 제안합니다..."

4차 산업혁명

- **초연결**과 **초지능** 환경에서 **초융합 서비스** 탄생
- 빅데이터 + 고도화된 디지털 기술과 역량
- 데이터 과학 ▶ 혁신을 추구하는 과정에서 4차 산업 혁명 실현

모든 것이 연결되고 보다 지능적인 사회로의 진화

- 다보스 포럼, 2016



모든 것이 연결되고 보다 지능적인 사회로의 진화

구분	1차 산업혁명	2차 산업혁명	3차 산업혁명	4차 산업혁명
시 기	18세기 후반	20세기 초반	20세기 후반	2020년 이후
혁신 부문	증기의 동력화	전기, 노동 분업	컴퓨터, ICT 혁명	IT 융합, 데이터 처리
커뮤니케이션 방식	책, 신문 등	전화기, TV 등	인터넷, SNS 등	사물인터넷(IoT), IoE
생산 방식	생산 기계화	대량생산	부분 자동화	시뮬레이션을 통한 자동생산
인간능력 대체	근육 (힘)	신경 (에너지)	두뇌 (일부)	두뇌 (일부) + 오감

출처: 전주대 이근호 교수 "인공지능 기초와 활용" 강연, 2023.7.4

데이터 과학

- IoT, 4G/5G 확산 => 빅데이터 생성, 전달, 저장 => AI => 초연결성 강화
- IoT(Internet of Things)
 - 센서, 기기 등에 인터넷 가능한 센서 부착
- 4G/5G : 촘촘한 무선네트워크
- AI(Artificial Intelligence) : 초지능
 - 인공지능으로 인지된 자극을 처리하고 분석하여 명령을 내리는 두뇌 역할

데이터 과학의 중요 사건

- [2007년] Smart Phone 혁명, 스티브잡스(1955 ~ 2011)
- [2016년] AlphaGo 혁명

알파고 리 vs. 이세돌 (바둑 경우의 수: 2.082×10^{170} 또는 361!, 우주 원자의 개수: 10^{82})

2017 알파고 제로 등장 (인간의 기보를 전혀 학습하지 않고 바둑 규칙 등 기본사항 학습한 후 스스로 '게임의 법칙' 터득, 독학 후 36시간만에 알파고 리 격파, 72시간 후 100전 100승)
그 배경에 [빅데이터 + 인공지능(머신러닝,딥러닝) + GPU]

- [2022년] ChatGPT 혁명

GPT-3 : 1750억개 매개변수, 단기 메모리는 약 8,000단어

GPT-4 : 100조개 매개변수, 단기 메모리는 약 64,000단어, 영어 외에 25개 언어 동시 작동

LLM(Large Language Model)

다중 모달 모델 : 텍스트와 이미지 데이터를 모두 처리할 수 있음 한국 네이버 하이퍼클로바 X(2023년)

> 네이버 뉴스(50년치), 블로그(9년치) 데이터 학습, 챗GPT대비 한국어 학습 데이터량 6,500배, 네이버 내부 서비스에 적용

글로벌 기술 전문가의 인공지능(AI) 평가



“챗GPT 같은 AI는 PC나 인터넷의 등장
만큼 세상에 영향을 줄 것”
빌 게이츠(MS 창업자)



“챗GPT는 무섭도록 좋다. 위험할
정도로 강력한 AI가 머지 않았다”
일론 머스크(테슬라 CEO)



“우리는 이제 AI 여행을 시작하고
있고 아직 정점에 이르지 않았다”
순다르 피차이(구글 CEO)



“AI는 우리 시대의 가장 강력한
기술이며 우리가 즐기는 모든
것이 어떤 식으로든 자동화할 것”
젠슨 황(엔비디아 CEO)



“정보 기술 용량의 대부분은
기계 지능에 사용될 것”
마이클 델(델 테크놀로지스 CEO)

AI의 문제



빌게이트

우리의 세상을 바꿀 것

- 일하고, 배우고, 여행하고, 건강관리하고, 소통하는 모든 일상이 AI를 중심으로 바뀔 것이다.
- 그러나 AI가 통제 불능의 상태가 될 위험성이 있기에 이를 멈추게 할 방법을 마련해야 한다



샘 올트먼 (오픈AI CEO)

"허위정보 유포와 사이버 공격 등이
염려되고, 특히 오답을 정답처럼
알려주는 환각(Hallucination) 이슈에
주의할 것"

AI의 문제



일론 머스크(테슬라 CEO)

우리는 모든 인공지능(AI) 연구소가 즉각 GPT-4를 능가하는 AI 시스템의 개발을 최소 6개월간 중단해 달라고 요청합니다.

- AI시스템이 인간과 경쟁
- 시스템이 사회와 문명에 잠재적 위험
- AI개발은 효과가 긍정적 및 위험관리 가능할 때 진행해야

유발하라리(히브리대 교수),스티브 워즈니악 등 1100여명 공동 서한 제출

2023.3.30일자 <https://v.daum.net/v/qDRoOFcppz>

비영리단체 인공지능디지털정책센터(CAIDP)는 연방거래위원회(FTC)에 오픈AI를 고발

CAIDP는

- "오픈AI의 GPT-4와 같은 대규모 언어 모델은 FTC 법을 위반한다"
- "GPT-4는 편향적이고 기만적이며 개인정보와 공공 안전에 위험을 가한다"고 주장

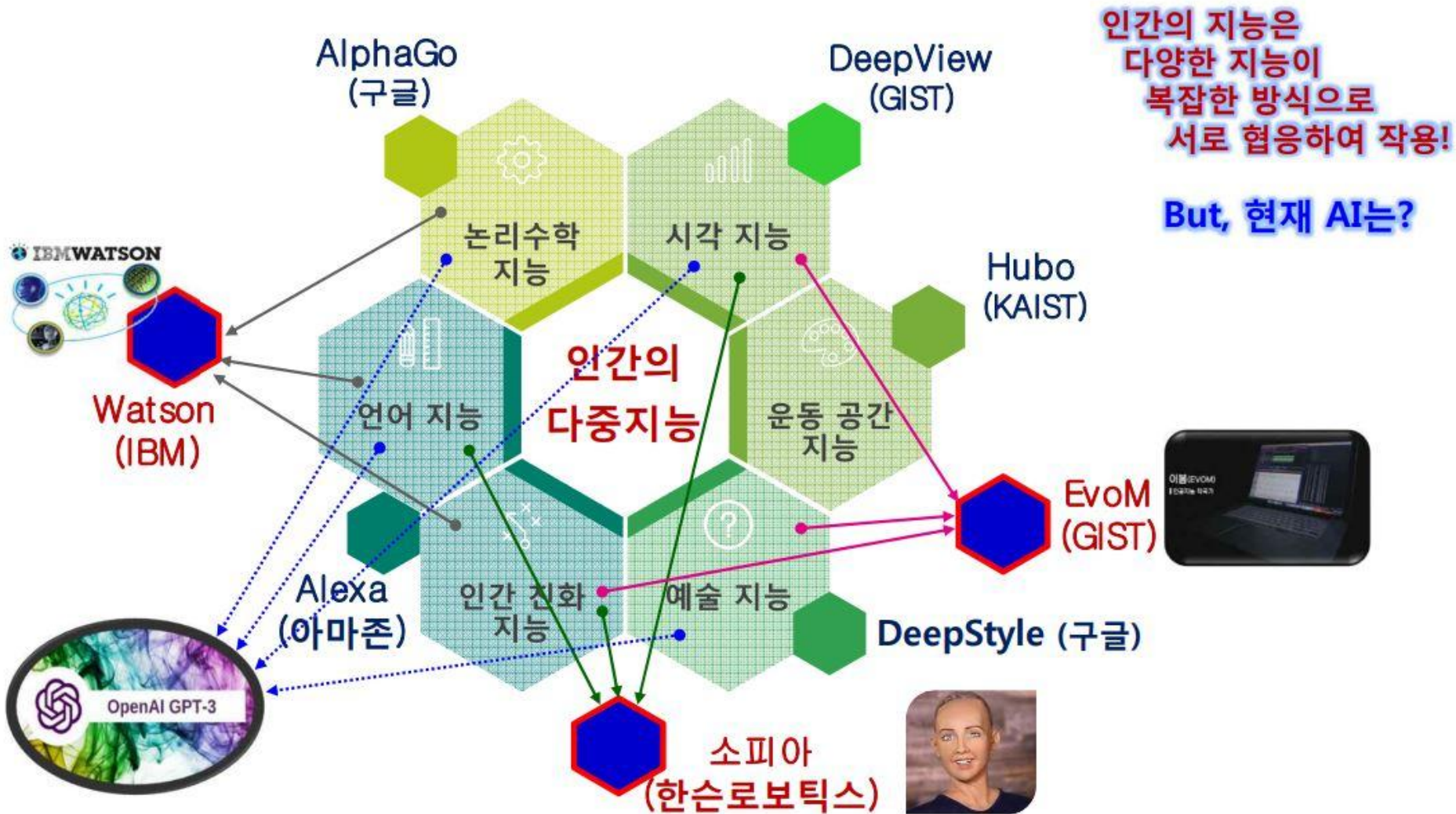
출처 : 디지털투데이 (DigitalToday)(<https://www.digitaltoday.co.kr>)

AI 어디까지 왔나, 앞으로 어떻게 될까

박태웅(녹서포럼 의장)



Howard Gardner의 인간의 다중지능 이론 관점에서...





챗GPT 등 AI가 대체할 10가지 직업

기술직(컴퓨터 프로그래머, 코드 작성자,
소프트웨어 엔지니어, 데이터 분석가)

미디어 직종(저널리즘, 광고, 콘텐츠 창작)

법률업 직종(준법률가, 법률 사무직원)

시장 리서치 분석가

교사

금융업(금융 분석가, 보험 설계사 등 개인 자산 자문가)

주식 트레이더

그래픽 디자이너

회계사

고객상담사

[그래픽=홍종현 미술기자]

 **NEWSPIM**

그래서 우리는 어떤 능력을 키워야 할까요?

시키는 것만 수행하는 능력



상상하고 기획하고 명령하는 능력



개발환경 만들기

TENSORFLOW

PANDAS

NUMPY

MATPLOTLIB

TENSORFLOW 지원

참조: <https://tensorflow.org/install?hl=ko>

The screenshot shows the TensorFlow installation page in Korean. The browser address bar displays 'tensorflow.org/install?hl=ko'. The page header includes the TensorFlow logo and navigation links: '설치' (Installation), '학습' (Learning), 'API', '리소스' (Resources), '커뮤니티' (Community), and 'TensorFlow를 사용해야 하는 이유' (Why use TensorFlow). The main content area is titled 'TensorFlow 2 설치' (TensorFlow 2 Installation). It states that TensorFlow is tested and supported on the following 64-bit systems:

- Python 3.6~3.9 (highlighted with a red dashed box)
- macOS 10.12.6(Sierra) 이상(GPU 지원 없음)
- Ubuntu 16.04 이상
- Windows 7 이상(C++ 재배포 가능 패키지)

The left sidebar contains a search filter and a list of links: 'TensorFlow 설치', '패키지' (Packages), '추가 설정' (Additional settings), 'GPU 지원' (GPU support), and '문제' (Issues).

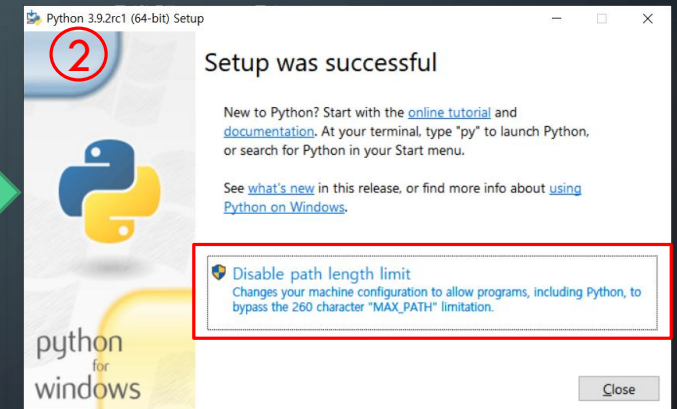
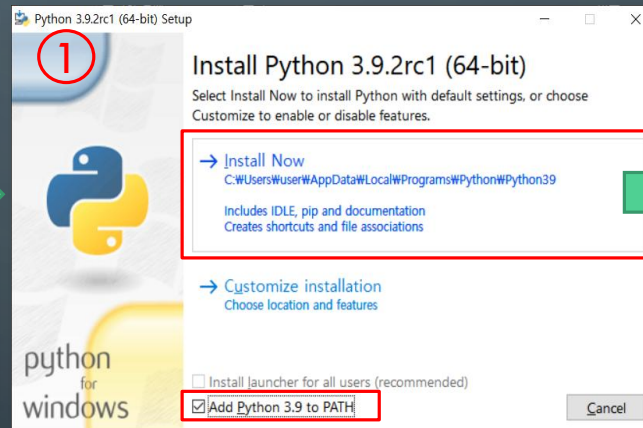
파이썬 설치(~ PYTHON-3.10.X)

- 버전: python-3.9.2rc1-amd64.exe
- 다운로드: <https://www.python.org/downloads/windows/>

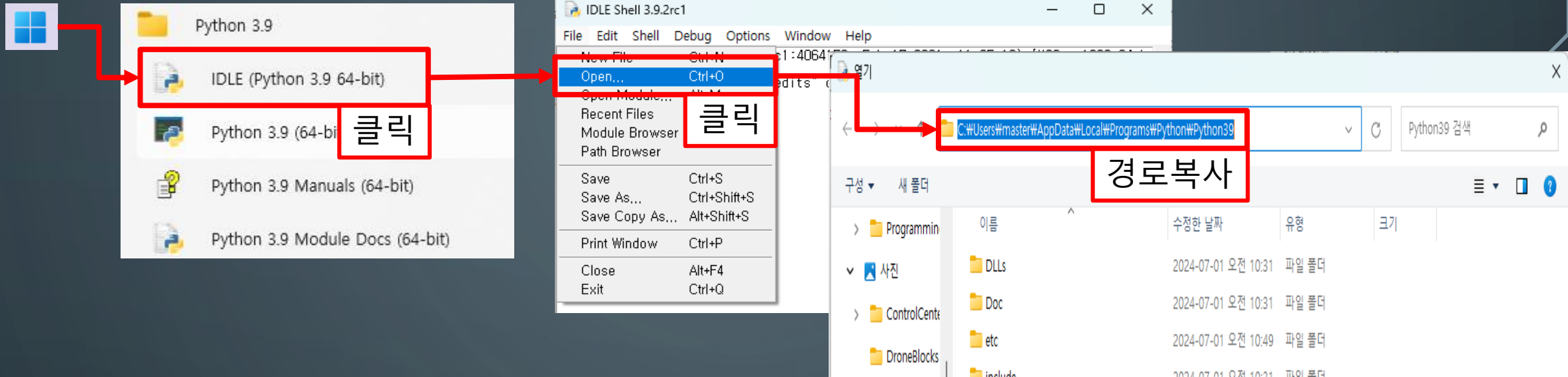
Python 3.9.2rc1 - Feb. 16, 2021

- Download [Windows installer \(64-bit\)](#)
- Download [Windows installer \(32-bit\)](#)
- Download [Windows help file](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows embeddable package \(32-bit\)](#)


다운로드 및 설치

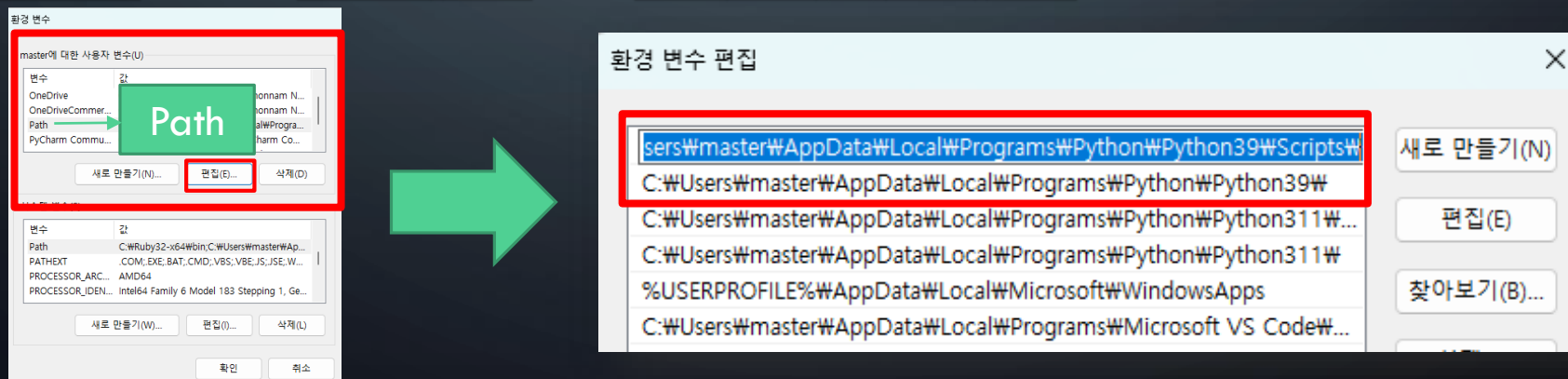


환경구성




위에서 복사한 경로를 시스템 환경변수에 등록하는 방법

- ① 실행창( + R)을 열고 **control system** 입력후 엔터 > **고급 시스템 설정** 클릭 > **환경변수** 클릭 > **사용자 변수**에서 **Path** 선택 후 **편집 버튼** 클릭 > 아래와 같이 새로 만들기로 등록



환경구성 및 주피터노트북 실행

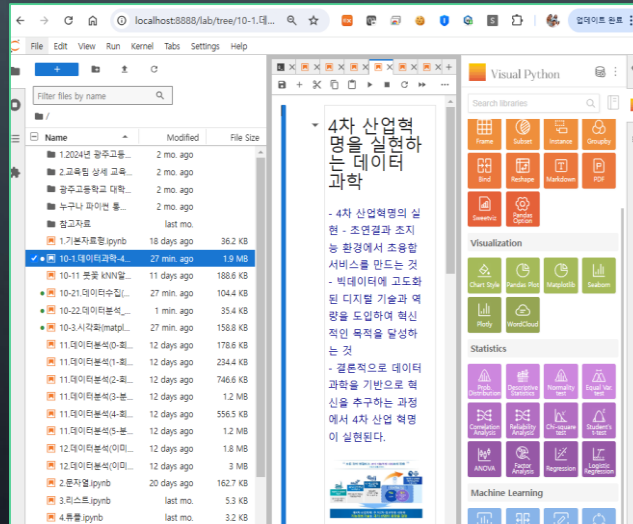
- 탐색기를 열어( + E) 작업폴더 만들기
 - 바탕 화면에 “인공지능” 폴더 생성 후 클릭
- 탐색기 주소창에 “**cmd**” 입력 후 엔터 > 명령창 실행

필수 패키지 설치

```
python -m pip install --upgrade pip
pip install tensorflow
pip install pandas
pip install scikit-learn
pip install scipy
pip install statsmodels
pip install jupyterlab
pip install jupyterlab-visualpython
```

주피터랩 실행

- [1] 탐색기로 실행폴더로 이동
- [2] jupyter lab

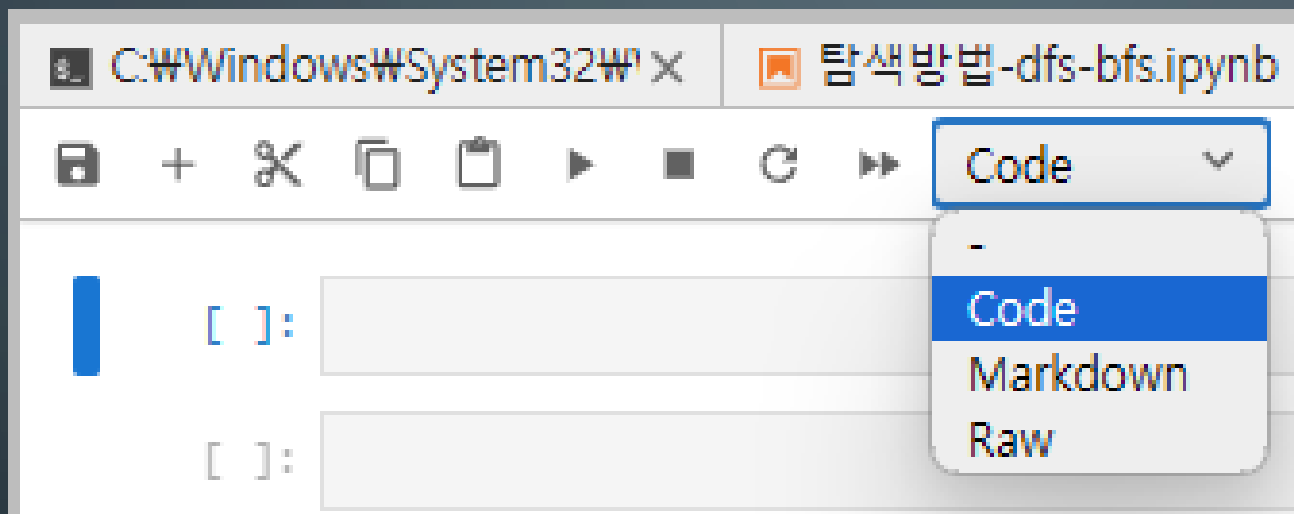


주피터랩 종료

```
PS C:\Users\master> netstat -anop tcp | findstr 8888
TCP    127.0.0.1:8888      0.0.0.0:0           LISTENING      14568
PS C:\Users\master> taskkill /PID 14568 /F
```

```
> netstat -anop tcp | findstr 8888
> taskkill /PID 14568 /F
```

주피터 사용법



마크다운 명령어(1)

마크다운
단축키: **ESC + M**

큰 제목(H1)

부제목(H2)

본문(H3)

글자크기

글자크기

글자크기

큰 제목

부 제목

본문(###)

글자 크기(####)

글자 크기(#####)

글자 크기(#####)

>블럭인용문자

블럭인용문자

마크다운 명령어(2)

프로그램 코드 보이기

```
```c
#include <stdio.h>

int main(){
 printf("hello\n");
 return 0;
}
```
```

...
```[언어] 영역에 java , python, c++, ...

또는

<pre>

<code>

```
#include <stdio.h>

int main(){
 printf("hello\n");
 return 0;
}
```

</code>

</pre>

```
#include <stdio.h>
int main(){
 printf("hello\n");
 return 0;
}
```

```
#include
int main(){
 printf("hello\n");
 return 0;
}
```

## 수평선 만들기

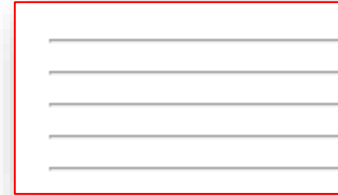
\* \* \*

\*\*\*

<hr/>

- - -

-----



## 외부링크

[Title](link)

예)

[Google](<https://www.google.com> “구글”)

## 자동링크

예1) 네이버: <https://www.naver.com>

# 마크다운 명령어(3)

## 강조

\*single asterisks\*

\_single underscores\_

\*\*double asterisks\*\*

\_\_double underscores\_\_

~~cancelline~~

*single asterisks*

*single underscores*

**double asterisks**

**double underscores**

~~cancelline~~

## 이미지

![text](URL)

예) 캡춰이미지 붙여넣기

![image.png](attachment:fae7e725-307a-486e-8f7a-832a95b76ffd.png)





>

## 줄바꿈

문장끝에 띄어쓰기 2칸이상 후 엔터 또는

\\

## 들여쓰기

level1

[탭]or [띄어쓰기] \* level2

또는

1. 첫번째

[탭]\* level2

2. 두번째

[탭]\* level2

[탭][탭]\* level3

## 폰트 글자, 색, 굵기, 기울기, 글꼴 바꾸기

<span style="color:red;font-size:15px;line-height:1em;font-style:italic;font-weight:bold;">글자크기 색깔 글간격 </span>

## Tex 문법 활용하여 넣기

\$문장\$ 또는

\$\$

$F(x) = ax^2 + bx + c$

$G(x) = Ax^4$

원의면적 =  $\pi r^2$

\$\$

예)

$f(x) = 3x^2 + 2x - 5$

$F(x) = ax^2 + bx + c$

$G(x) = Ax^4$

원의면적 =  $\pi r^2$

$f(x) = 3x^2 + 2x - 5$

Tex 참조사이트 바로가기

Tex문법 참조

\$\$

$F(x) = ax^2 + bx + c$

$G(x) = Ax^4$

$y = a \times x + b$

$y_i = x_i^3 + x_{i-1}^2 +$

$x_{i-2}$

\$\$

$F(x) = ax^2 + bx + c$

$G(x) = Ax^4$

$y = a \times x + b$

$y_i = x_i^3 + x_{i-1}^2 + x_{i-2}$

\$\$

$\log_{10}(x+1)$

$\log_{10}(x+1)$

\$\$

$|x|$

$\log_{10}(x+1)$

$\sqrt[3]{x^3 + y^3}$

$\sqrt[3]{\frac{x^3 + y^3}{2}}$



# 마크다운 명령어(4)

## 테이블

- |로 뼈대를 잡아주고 칼럼명과 내용은 --로 구분
- 예)
- | 헤더1 | 헤더2 | 헤더3 |
- | ---- | ---- | ---- |
- | row1 | row1-2 | row1-3 |
- | 하나의 내용만 보이기 |||
- | row1 | row1-2 | row1-3 |

## 또는 table 태그 활용

```
<table width=800px border=1 align=left>
```

```
<caption> 문자열 format 형식</caption>
```

```
<thead>
```

```
<tr><th>형식지정</th><th>의미</th></tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr><td align=center>{
}</td><td>전체10칸을 차지하며 공백을 앞에 붙임, 오른쪽 정렬</td></tr>
```

```
<tr><td align=center>{
}</td><td>전체10칸을 차지하며 공백을 앞에 붙임, 왼쪽쪽 정렬</td></tr>
```

```
<tr><td align=center>{
}</td><td>전체10칸을 차지하며 공백을 앞에 붙임, 가운데 정렬</td></tr>
```

```
<tr><td align=center>{}</td><td>부동소수점의 소수점 아래 5자리까지 표시</td></tr>
```

```
<tr><td align=center>{}</td><td>천단위 쉼표 표시</td></tr>
```

```
</tbody>
```

```
</table>
```

헤더1

헤더2

헤더3

row1

row1-2

row1-3

하나의 내용만 보이기

row2

row2-2

row2-3

## 문자열 format 형식

형식지정

의미

{>10}

전체10칸을 차지하며 공백을 앞에 붙임, 오른쪽 정렬

{<10}

전체10칸을 차지하며 공백을 앞에 붙임, 왼쪽쪽 정렬

{^10}

전체10칸을 차지하며 공백을 앞에 붙임, 가운데 정렬

{.5f}

부동소수점의 소수점 아래 5자리까지 표시

{,}

천단위 쉼표 표시

# 마크다운 명령어(5) – 테이블 또는 수식 정렬

<span style="display: inline-block;text-align:left;font-size:20px;margin-left:5em;">

테이블 또는 수식

</span>

# 4비트의 2진수에 대한 10진수 표현

<span style="display: inline-block;text-align:left;font-size:20px;margin-left:5em;">

4비트의 2진수 | 10진수 변환 | 10진수(16진수표현)

4비트의 2진수	10진수 변환	10진수(16진수표현)
0000	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$	0(0x0)
0001	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	1(0x1)
0010	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	2(0x2)
0011	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	3(0x3)
0100	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	4(0x4)
0101	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	5(0x5)
0110	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	6(0x6)
0111	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	7(0x7)
1000	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	8(0x8)
1001	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	9(0x9)
1010	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	10 = A(0xA)
1011	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	11 = B(0xB)
1100	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	12 = C(0xC)
1101	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	13 = D(0xD)
1110	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	14 = E(0xE)
1111	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	15 = F(0xF)

</span>

4비트의 2진수에 대한 10진수 표현

4비트의 2진수	10진수 변환	10진수(16진수표현)
0000	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$	0(0x0)
0001	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	1(0x1)
0010	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	2(0x2)
0011	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	3(0x3)
0100	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	4(0x4)
0101	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	5(0x5)
0110	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	6(0x6)
0111	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	7(0x7)
1000	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	8(0x8)
1001	$0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$	9(0x9)

# 인공지능(머신러닝) 개발 개요

## (사례1) 자율주행 : 인지 - 판단 - 제어

### 자율주행 5단계

레벨0: 자동화 기능의 미적용 상태

레벨1: 운전자 보조주행: 운전자가 속도 또는 방향을 통제

레벨2: 부분적 자율주행: 차간 거리 및 속도 유지 등이 가능하지만 운전자가 주행에 적극 개입해야 하는 상태\*

레벨3: 조건부 자율주행: 자율주행 시스템을 운행하지만 비상시 몇 초 안에 운전자가 개입해야 하는 상태

레벨4: 고수준 자율주행: 비상시 차량이 일정 시간은 자체 대응하는 상태로, 운전자가 차량 내에서 책을 읽어도 되는 수준

레벨5: 완전 자율주행: 어떠한 도로 환경에서도 무인 자율주행이 가능한 상태

## (사례2) 커넥티드카 connected car

- 정보통신기술과 자동차를 연결시킨 것으로 양방향 인터넷 및 모바일 서비스가 가능한 차량
- 커넥티드카는 차량과 도시의 모든 곳이 연결되어 스스로 위험을 감지하고 다른 자동차와의 거리나 속도를 제어하며 운전
- 차량 고장에 대해 스스로 필요한 조치를 취할 뿐아니라 영화, 날씨, 뉴스검색, SNS 등 다양한 운전자 맞춤형 서비스를 제공

## (사례3) 스마트시티, 스마트 헬스케어

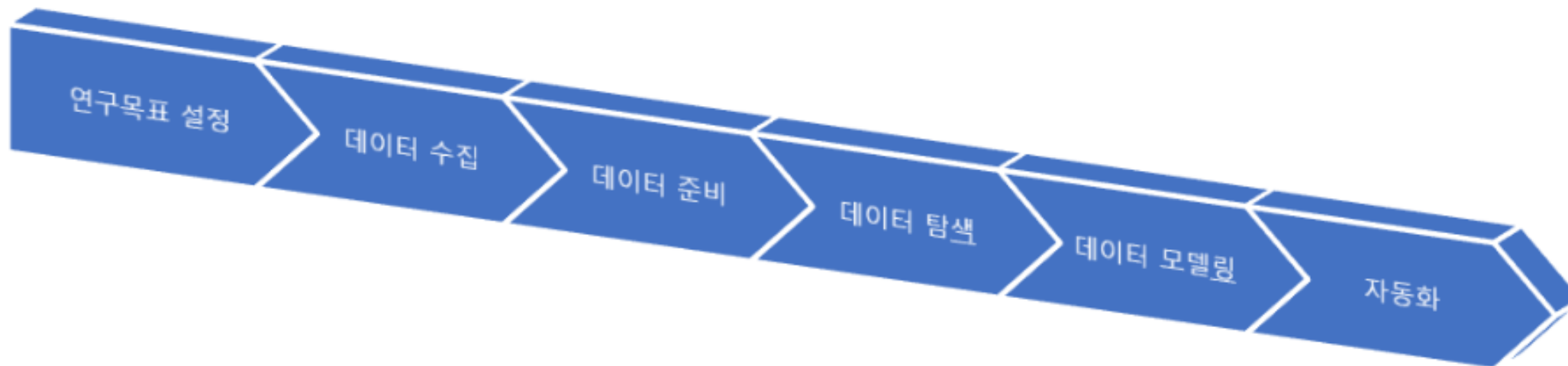
## 빅데이터란

1. 빅데이터 분류: 정형(DB, 스프레드시트 등), 반정형(API/XML, HTML, JSON, 웹로그 등), 비정형(소셜데이터, 텍스트문서, 이미지/동영상/음성 등)
2. 빅데이터 특징: 규모(Volume), 다양성(Variety), 속도(Velocity) + 정확성(Veracity), 가치(Value)

## 빅데이터 처리 단계

1. 데이터소스
2. 수집: 크롤링(Crawling), 추출(Extraction), 변환(Transformation), 적재>Loading)
3. 저장: NoSQL(비정형데이터 저장), 저장소(Storage)
4. 처리: 맵리듀스(MapReduce), 다중처리작업
5. 분석: 신경망, 머신러닝, 직렬화(Serialization)
6. 표현: 시각화(Visualization), 데이터 획득 및 재해석

## 빅데이터 분석을 위한 데이터 과학 방법론





## 개방 데이터를 제공하는 사이트

사이트	설명
<a href="http://data.go.kr">http://data.go.kr</a>	한국 정보에서 제공하는 공공데이터
<a href="http://kostat.go.kr">http://kostat.go.kr</a>	한국 통계청 공개 데이터
<a href="http://opendata.hira.or.kr">http://opendata.hira.or.kr</a>	한국 보건 의료 빅데이터 개방 시스템
<a href="https://kosis.kr">https://kosis.kr</a>	국가 통계 포털
<a href="https://data.kma.go.kr">https://data.kma.go.kr</a>	기상 자료 개발 포털
<a href="http://www.airkorea.or.kr">http://www.airkorea.or.kr</a>	대기 오염 정보
<a href="http://data.ex.co.kr">http://data.ex.co.kr</a>	고속도로 공공 데이터 포털
<a href="http://www.localdata.kr">http://www.localdata.kr</a>	한국 지방행정 인허가 데이터
<a href="https://www.mcst.go.kr">https://www.mcst.go.kr</a>	한국 문화체육관광부 문화 데이터
<a href="http://data.seoul.go.kr">http://data.seoul.go.kr</a>	서울시 열린데이터 광장
<a href="https://data.gg.go.kr">https://data.gg.go.kr</a>	경기도 공공데이터 개방 포털
<a href="http://data.gov">http://data.gov</a>	미국정보의 공공데이터
<a href="http://data.worldbank.org">http://data.worldbank.org</a>	세계 은행에서 제공하는 개방 데이터
<a href="http://open.fda.gov">http://open.fda.gov</a>	미국 식약청의 개방 데이터

## 검색어를 활용한 데이터 수집

사이트	설명
<a href="https://trends.google.co.kr">https://trends.google.co.kr</a>	구글 트렌드
<a href="https://datalab.naver.com">https://datalab.naver.com</a>	네이버 데이터랩
<a href="https://datatrend.kakao.com">https://datatrend.kakao.com</a>	카카오 데이터트렌드

## 표

- (외계인의 질문) 인류가 만든 시각화 도구 중에서 가장 위대한 것은?
- (답) 표와 좌표평면
- 표는 데이터의 모음
- **데이터 세트**

## 행(row)

개체(instance)  
관측치(observed value)  
레코드(record)  
케이스(case)  
사례(example)

	student number	english	mathematics
1	1	42	65
2	2	69	80
3	3	56	63
4	4	41	63
5	5	57	76
6	6	48	60
7	7	65	81
8	8	49	66
9	9	65	78
10	10	58	82

## 열(column)

특성(feature)  
속성(attribute)  
변수(variable)  
필드(field)

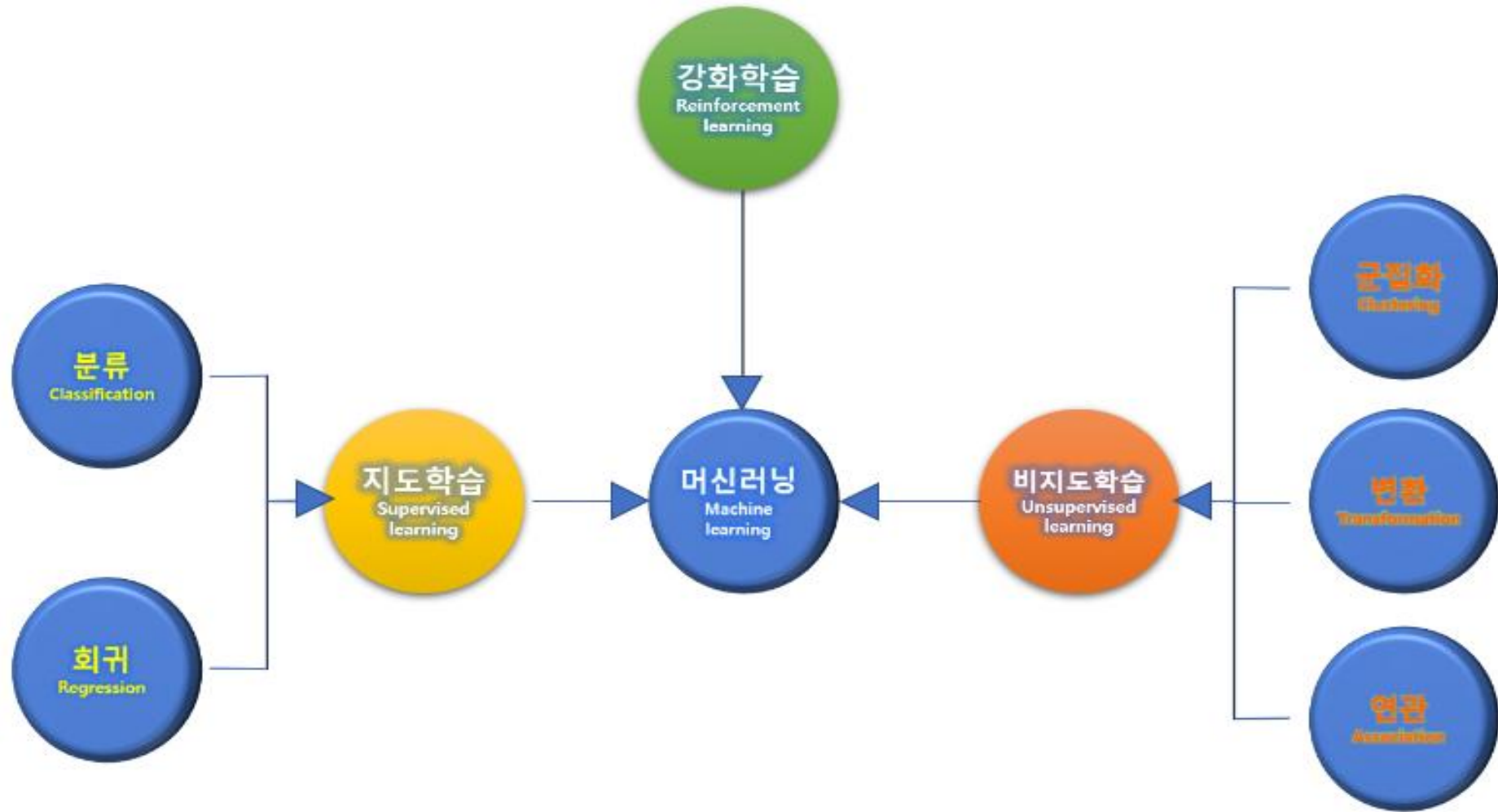
## 상관관계

날짜	요일	온도	판매량
2024.1.3	수	20	40
2024.1.4	목	21	42
2024.1.5	금	22	44

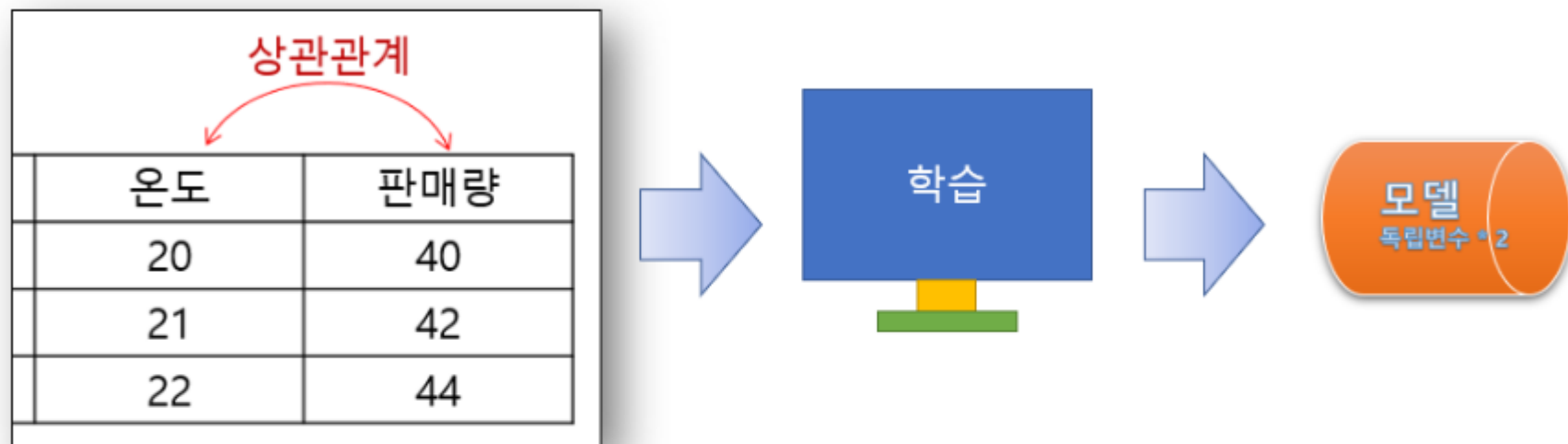
- 이 표는 분석을 해 볼 때, 온도가 20도일때 판매량이 40이었고, 계속 온도에 따라 판매량이 늘어나고 있음
- 이 때에 온도는 다른 외부요인에 상관없는 데이터이므로, **독립변수(Independent variable)**
- 판매량은 온도에 따라 영향을 받음으로 **종속변수(Dependent variable)**라고 한다.

# 머신러닝의 분류

- 지도학습: 문제와 정답을 비교하여 학습하는 형태
- 비지도학습: 정답을 알려주지않고, 대상에 대한 관찰을 통해 의미나 관계를 찾아내는 형태
- 강화학습: 게임 등에 적용되는 형태로 상 또는 벌점 등 보상을 이용하여 학습하는 형태



## 지도학습 Supervised Learning



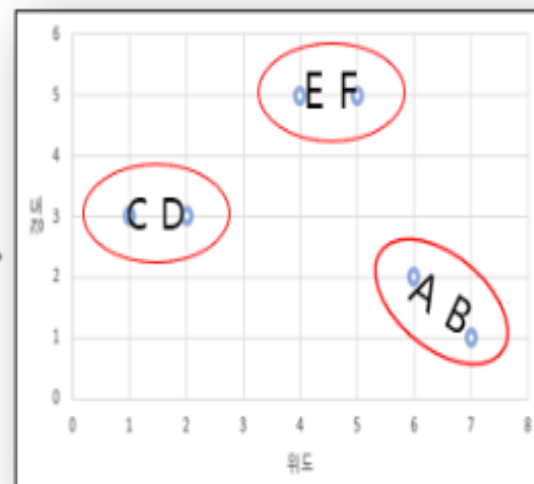
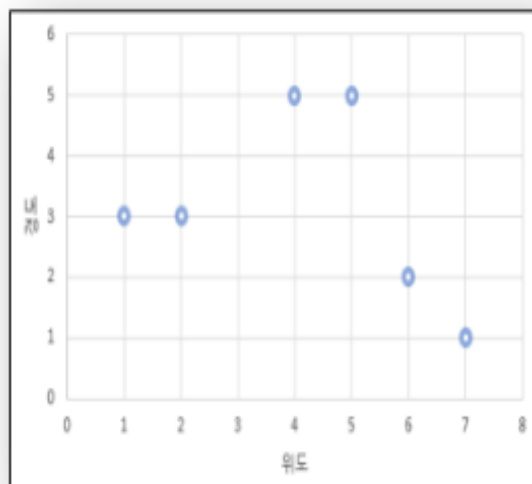
위 사례의 알고리즘: 회귀분석

독립변수	종속변수	학습방법
공부시간	시험점수	사람들의 공부시간을 입력받고 점수를 확인
온도	판매량	온도와 그날의 판매량을 기록
역세권, 조망 등	집값	집과 역까지의 거리, 수치화된 조망의 평점 등을 집값과 함께 기록
온실 기체량	기온 변화량	시간순으로 배출된 온실 기체량과 기온의 변화량을 기록
자동차 속도	충돌시 사망 확률	충돌시 속도와 사망자 기록
나이	키	학생들의 나이에 따른 키 기록



## 비지도학습 : 군집화(Clustering)

이름	위도	경도
A	7	1
B	6	2
C	2	3
D	1	3
E	5	5
F	4	5



군집화를 통해 3개 클러스터

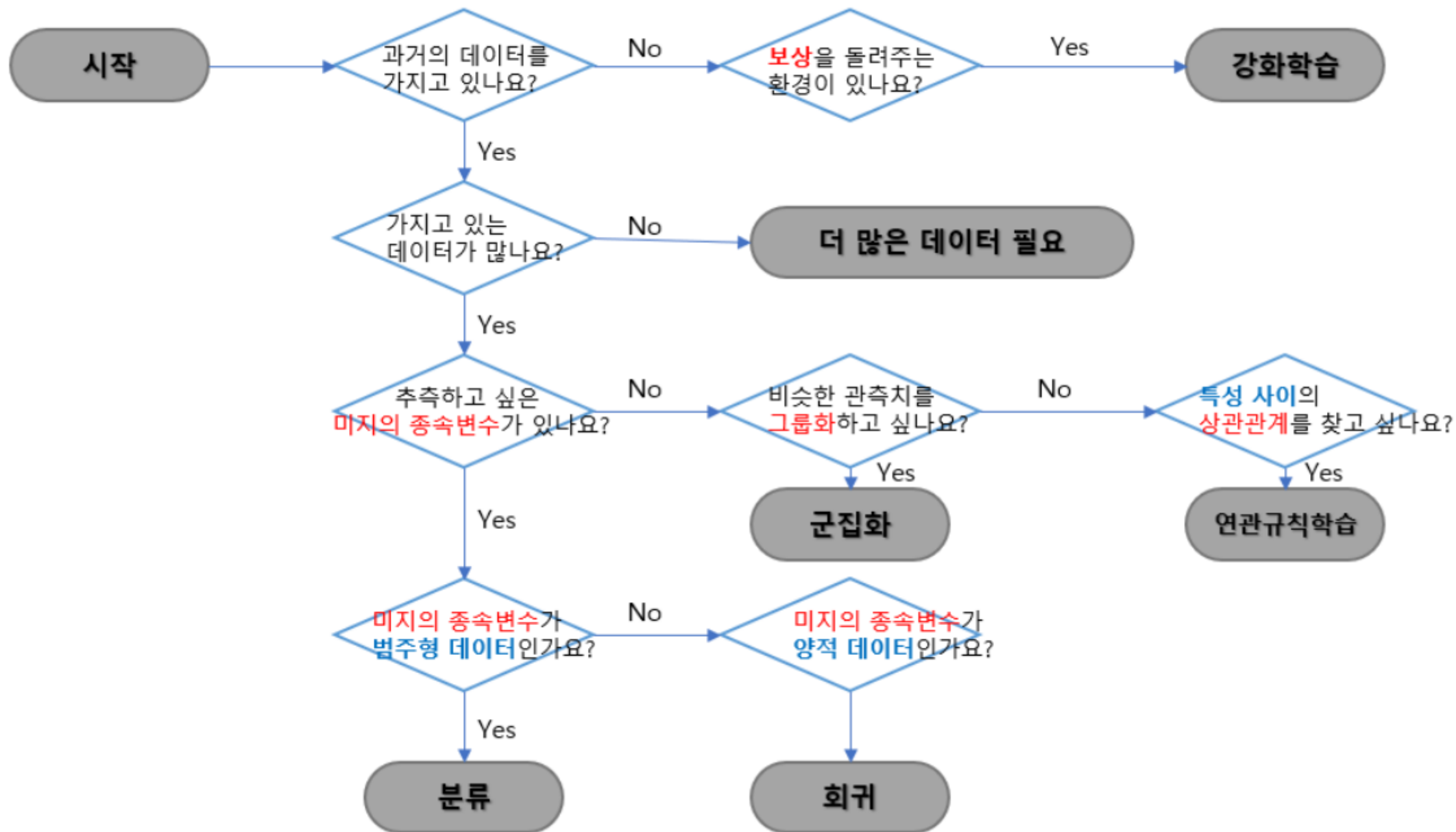
## 강화학습 Reinforcement Learning

- 자신이 한 행동에 대해 보상(Reward)를 받으며 학습하는 것
- 컴퓨터가 주어진 상태에 대해 최적의 행동을 선택하도록 학습하는 방법
- 강화학습을 이해하기 위해 알아야 할 개념들

에이전트(Agent)	주어진 문제 상황에서 행동하는 주체
상태(State)	현재 시점에서의 상황
행동(Action)	플레이어가 취할 수 있는 선택지
보상(Reward)	플레이어가 어떤 행동을 했을 때 따라오는 이득
환경(Environment)	문제 상황
관찰(Observation)	에이전트가 수집한 환경에 대한 정보



# 머신러닝 지도



# 간단한 데이터 분석 처리 과정

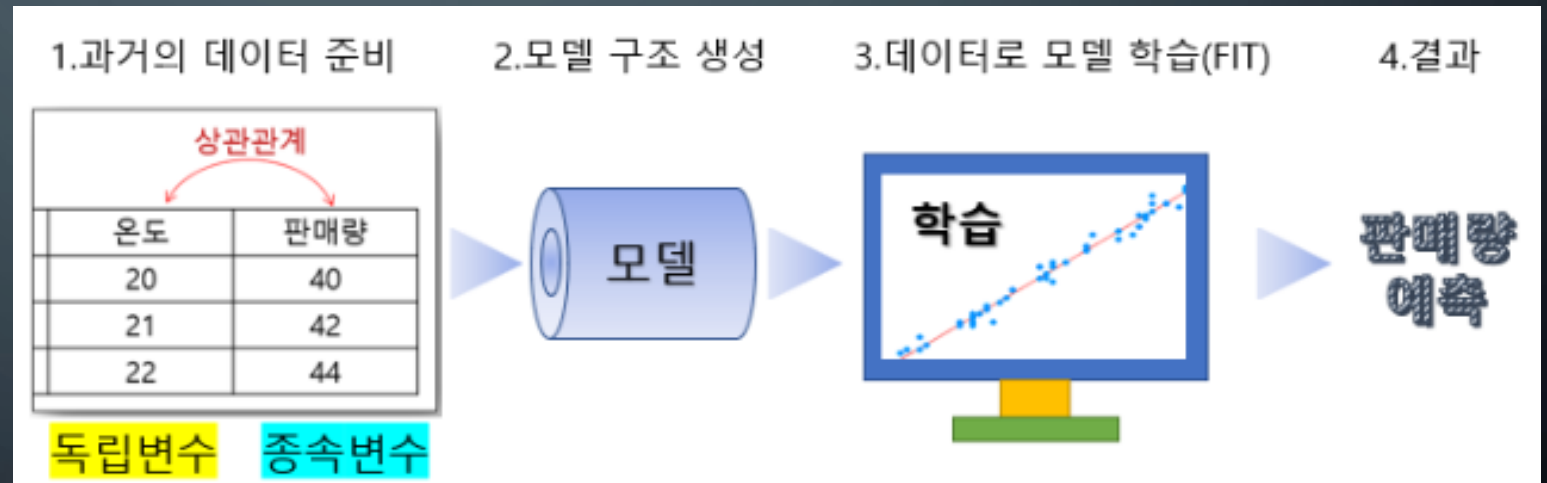
1.데이터 수집( 전처리)

2.모델 생성

3.학습 FIT

4.예측 PREDICT

5.시각화



```
관련 라이브러리 추가
import tensorflow as tf # 머신러닝: tensorflow
import pandas as pd # 데이터: Pandas
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
데이터 가져오기
레모네이드 = pd.read_csv('https://raw.githubusercontent.com/blackdew/tensorflow1/master/csv/lemonade.csv')
독립 = 레모네이드[['온도']]
종속 = 레모네이드[['판매량']]
레모네이드
```

	온도	판매량
0	20	40
1	21	42
2	22	44
3	23	46
4	24	48
5	25	50



##2. 모델 구조 생성

```
X = tf.keras.layers.Input(shape=[1])
```

```
Y = tf.keras.layers.Dense(1)(X)
```

```
model = tf.keras.models.Model(X,Y)
```

```
model.compile(loss='mse') # mse, mean square error
```

### ##3. 데이터로 모델 학습(fit)

```
model.fit(독립, 종속, epochs=2000, verbose=False) # verbose=False 옵션은 내뱉보기 안함
model.fit(독립, 종속, epochs=10) # default는 내뱉보기
```

```
Epoch 1/10
1/1 ————— 0s 22ms/step - accuracy: 0.0000e+00 - loss: 9.5479e-04
Epoch 2/10
1/1 ————— 0s 25ms/step - accuracy: 0.0000e+00 - loss: 9.5399e-04
Epoch 3/10
1/1 ————— 0s 26ms/step - accuracy: 0.0000e+00 - loss: 9.5322e-04
Epoch 4/10
1/1 ————— 0s 24ms/step - accuracy: 0.0000e+00 - loss: 9.5269e-04
Epoch 5/10
1/1 ————— 0s 26ms/step - accuracy: 0.0000e+00 - loss: 9.5199e-04
Epoch 6/10
1/1 ————— 0s 25ms/step - accuracy: 0.0000e+00 - loss: 9.5132e-04
Epoch 7/10
1/1 ————— 0s 24ms/step - accuracy: 0.0000e+00 - loss: 9.5058e-04
Epoch 8/10
1/1 ————— 0s 25ms/step - accuracy: 0.0000e+00 - loss: 9.4983e-04
Epoch 9/10
1/1 ————— 0s 24ms/step - accuracy: 0.0000e+00 - loss: 9.4907e-04
Epoch 10/10
1/1 ————— 0s 24ms/step - accuracy: 0.0000e+00 - loss: 9.4843e-04
<keras.src.callbacks.history.History at 0x172d0a9fb20>
```

##4. 모델을 이용하여 예측

##DataFrame 생성하여 테스트

```
테스트 = pd.DataFrame({'온도':[18,18.2,19.3,19.6]})
```

```
print("온도에 따른 판매량 예측:\n",model.predict(테스트))
```

1/1 ————— 0s 20ms/step

온도에 따른 판매량 예측:

[[36.37152 ]

[36.75596 ]

[38.870377]

[39.447044]]

##5. 가중치 정보확인

```
print(model.get_weights())
```

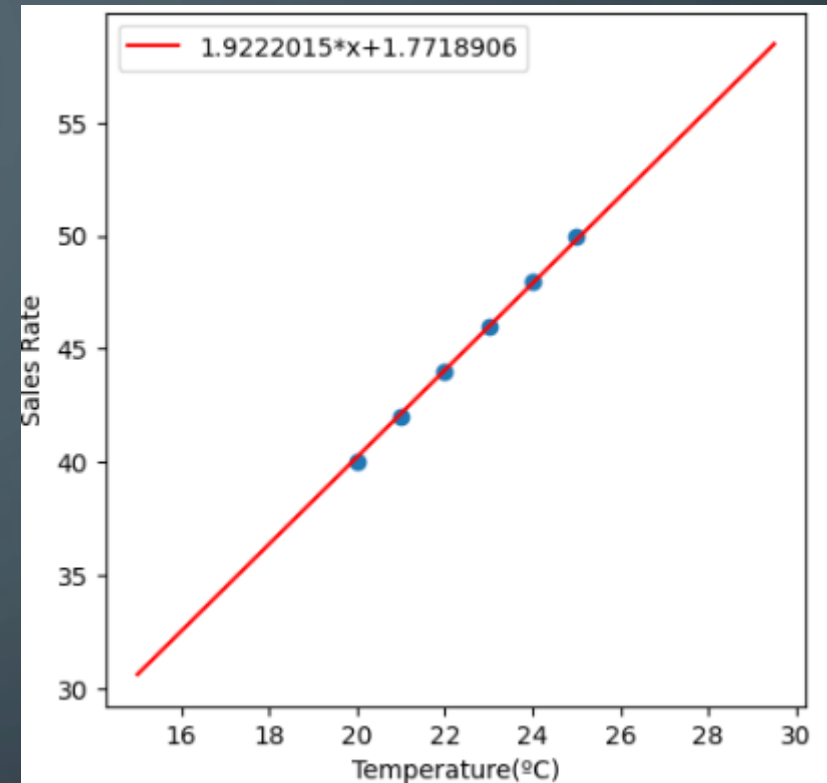
```
[array([[1.9222015]], dtype=float32), array([1.7718906], dtype=float32)]
```

##6. 시각화

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
기울기, 절편
ww = model.get_weights()

데이터
temp = np.arange(15.0, 30.0, 0.5, dtype=np.float64)
sales = ww[0][0][0]*temp+ww[1][0]

그래프
fig = plt.figure(figsize=(5,5))
ax = fig.add_subplot(111)
ax.set_xlabel('Temperature(°C)')
ax.set_ylabel('Sales Rate')
ax.scatter(독립, 종속)
ax.plot(temp,sales,color='red', label='1.9222015*x+1.7718906')
ax.legend()
plt.show()
```



감사합니다.