

T-EXPERT

# 『6과목』-미니프로젝트 날씨에 따른 교통사고 발생 확률 예측

2024.06.10-2024.06.21

Prepared by DaeKyeong Kim

Ph.D.



KOREATECH



## 진행 순서

- 이 워크샵에서는 날씨에 따른 교통사고 발생 확률을 예측하는 인공지능 시스템을 개발할 수 있다.
- 진행 순서는 다음과 같다.

1. Problem Scoping

---

2. Data Acquisition

---

3. Data Exploration

---

4. Modeling

---

5. Evaluate

---

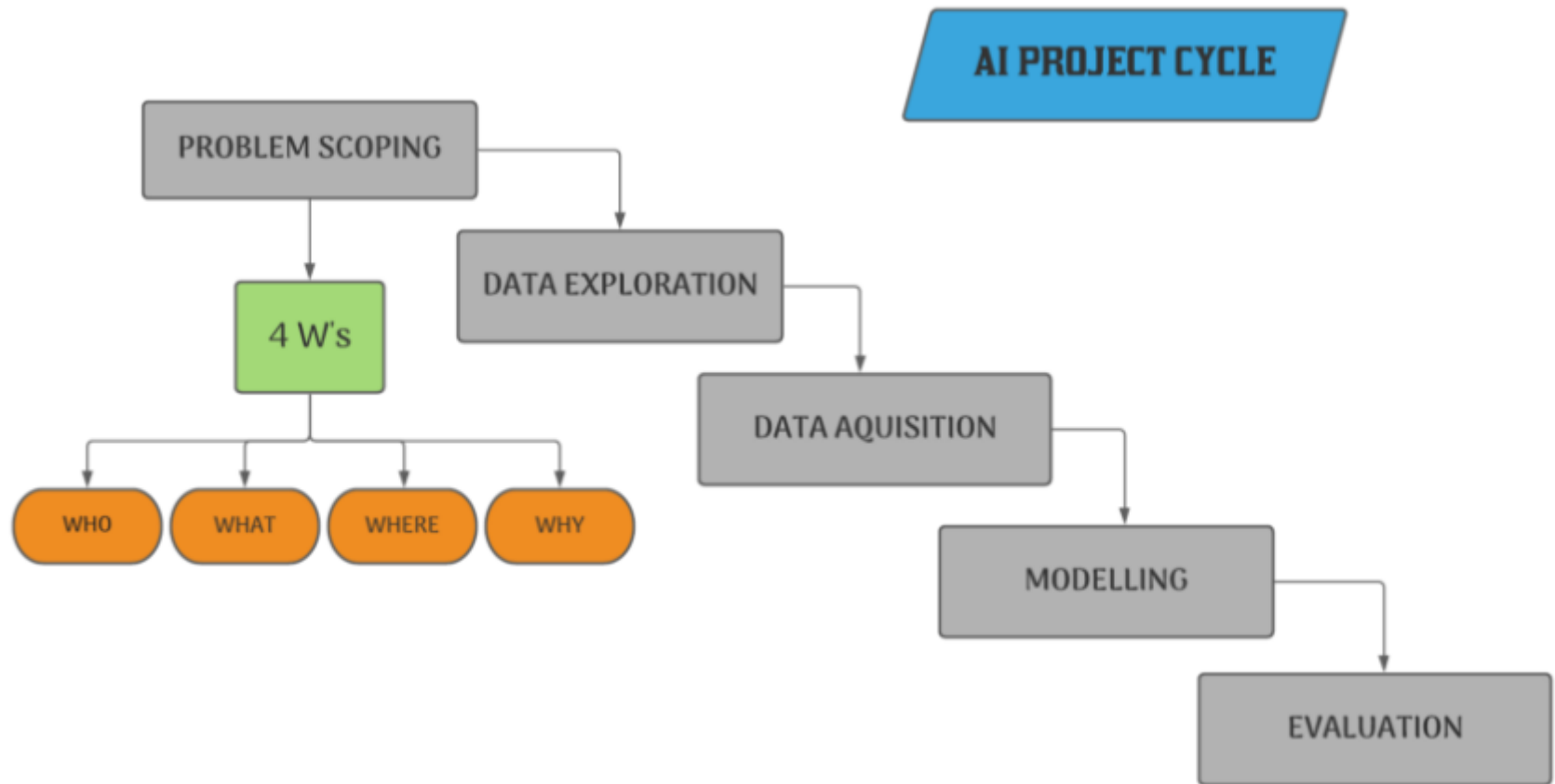
6. Deployment

---



# 1. Problem Definition

## 4ws problem canvas



# 1. Problem Definition

## 분석 목표 정의서

분석 기본 정의	분석 명칭	날씨에 따른 교통사고 발생 확률 예측 인공지능 시스템 개발	분석목표 확정일	2021-12-XX
	분석 목적	날씨에 따른 사망사고 발생 확률을 예측할 수 있게 하여 운전을 하지 않거나 주의하여 운전	분석 목표 워크숍	2021-12-XX
	분석 우선순위	상	담당 조직명	
	분석 접근 방안			
성과 측정	정성적 기준			
	정량적 기준			
데이터 정보	내부 데이터		데이터 입수 난이도	중
	외부 데이터	공공포털 사이트에서 2018년 서울시 기상 데이터	데이터 입수 난이도	

# 1. Problem Definition

## 프로젝트 헌장

프로젝트 헌장(Project Charter)			
프로젝트 명 (Project Name)	날씨에 따른 교통사고 발생 확률 예측 인공지능 시스템 개발		
프로젝트 설명 (Project Description)			
프로젝트 매니저(Project Manager, PM )		승인 날짜(Date Approved)	
프로젝트 스폰서(Project Sponsor)		서명(Signature)	인용
비즈니스 케이스 (Business Case)	목표(Goals) / 산출물(Deliverables)		
팀 구성원(Team Member)			
이름(Name)	역할(Role)		
위험과 제약사항(Risk and Constraints)		주요 일정(Milestones)	

# 1. Problem Definition

## 분석 설계

날씨에 따른 교통사고 발생 확률 예측 인공지능 시스템 개발	
목표	날씨에 따른 사망사고 발생 확률을 예측할 수 있게 하여 운전을 하지 않거나 주의하여 운전
핵심개념	
데이터 수집	○○○ 데이터 셋 : ○○○ 저장소에서 다운로드
데이터 준비	수집한 데이터 파일 병합
데이터 탐색	1. 정보 확인 : info() 2. 기술 통계 확인 : describe(), unique(), value_counts)
결과 시각화	
xgb_model을 이용한 피쳐 엔지니어링	<b>TimeSeries 분석을 이용한 시각화</b>
	<div><div>날씨에 따른 교통사고 발생확률 예측 AI 시스템</div><div>일기예보를 확인하세요. 기상청 예보 사이트를 참고하세요.</div><div>기온( ): 오늘 기온은 얼마인가요? <input type="text"/></div><div>강수량(mm): 오늘 비가 오나요? 온다면 얼마나 올까요? <input type="text"/></div><div>풍속(m/s): 오늘 바람이 부나요? 온다면 세기가 얼마나 될까요? <input type="text"/></div><div>습도(%): 오늘 습도가 얼마일까요? <input type="text"/></div><div>적설량(㎝): 오늘 눈이 오나요? 온다면 얼마나 올까요? <input type="text"/></div><div>교통사고 발생확률 예측 시작 버튼</div></div> <div>예측되는 교통사고 삼상사자 수는 [.....]명 입니다.</div>

# 2. Data Collection/Data Acquisition

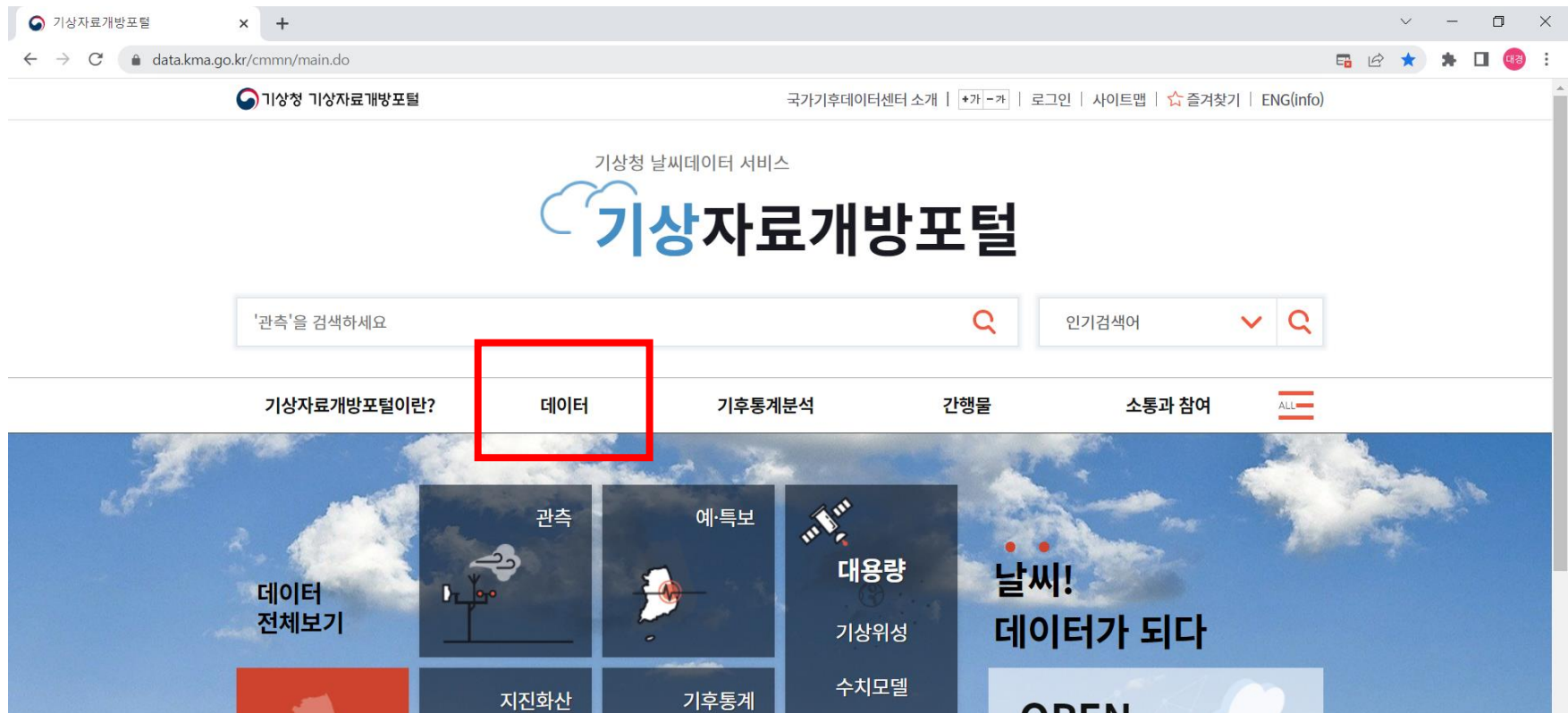
## 데이터 수집 세부 계획서

날씨에 따른 교통사고 발생 확률 예측 인공지능 시스템 개발 데이터 수집 세부 계획서						
날씨에 따른 교통사고 발생 확률 예측 인공지능 시스템 개발을 위한 데이터 수집 세부 계획서					담당자	
문서번호		작성자		작성일자		
1. 분석 목적						
2. 수집 데이터 상세 조사 내용						
데이터 유형	통계   문자   텍스트   음성   이미지   동영상   GIS   기타				수집 주기	
위치	수요기업   공급기업 보유 또는 수집   허브 데이터셋   공공 데이터				확보 비용	
크기	레코드수	300	레 코 드 단 위	장	데이터 이관 절차	
	크기	15	단위	MB		
보관 방식						
3. 적절성 검증 방식						
데이터 누락/중복						
데이터 오류						
개인정보 유무					포함   미포함	
데이터 저작권						

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

- <https://data.kma.go.kr/cmmn/main.do> 에 접속 후, 데이터 메뉴를 클릭합니다.



기상청 기상자료개방포털

국가기후데이터센터 소개 | +가-가 | 로그인 | 사이트맵 | ☆ 즐겨찾기 | ENG(info)

기상청 날씨데이터 서비스

기상자료개방포털

'관측'을 검색하세요

인기검색어

기상자료개방포털이란? **데이터** 기후통계분석 간행물 소통과 참여

데이터 전체보기

관측

예·특보

대용량

지진화산

기후통계

수치모델

날씨! 데이터가 되다



## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

- 검색조건에서 2020/01/01 - 2020/12/31 입력합니다.

data.kma.go.kr/data/grnd/selectAsosRltmList.do?pgmNo=36



- 공공기관 기상관측  
- 계절관측  
- 항사관측(PM10)  
- 낙뢰관측

해양 +

고층 +

항공 +

세계기상전문(GTS) +

기상위성

레이더

기상예보

수치모델

기후

응용기상

지진화산

날씨 이슈별 데이터

역사기후

메타데이터

품질정보

**유의사항**

- 1회 조회 가능 최대 기간: 분 1일, 시간 1년, 일 10년, 월-연 제한 없음(장기간 자료는 '파일셋 조회' 메뉴 이용)
- 시간/분 자료에 대해 관측값의 정상 여부를 판단하는 품질검사 플래그(QC FLAG) 정보 제공
- \* 제공 요소: 기온, 습도, 기압, 지면온도, 풍향, 풍속, 일조 / 플래그 종류(의미): 0(정상), 1(오류), 9(결측)
- 전일 자료는 당일 10시 이후 확인 가능

**비고**

- 10분 또는 1시간 최대강수량은 최대강수가 나타난 시작 시간으로, (-) 표기가 있는 경우 전날을 뜻함
- 강수량은 겨울철(11월~익년 3월) 3시간 간격으로 제공

**지침** 요소별 관측방법이나 자료 산출방식에 대한 상세 설명은 [자상기상관측지침] 참조

자료

파일셋

캘린더

OPEN-API

#### ■ 검색조건

자료형태 시간 자료

기간 20220101 ~ 20240523

지점 지도로 선택

☐ 전체  
☐ 강원특별자치도  
☐ 경기도  
☐ 경상남도  
☐ 경상북도  
☐ 광주광역시  
☐ 대구광역시  
☐ 대전광역시  
☐ 부산광역시  
☐ 서울특별시  
☐ 세종특별자치시  
☐ 울산광역시  
☐ 인천광역시  
☐ 전라남도

> 조회

#### ■ 자료보기

CSV

Excel

지점

시간

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

- 지점 > 서울특별시 -관악산, 서울 체크합니다.

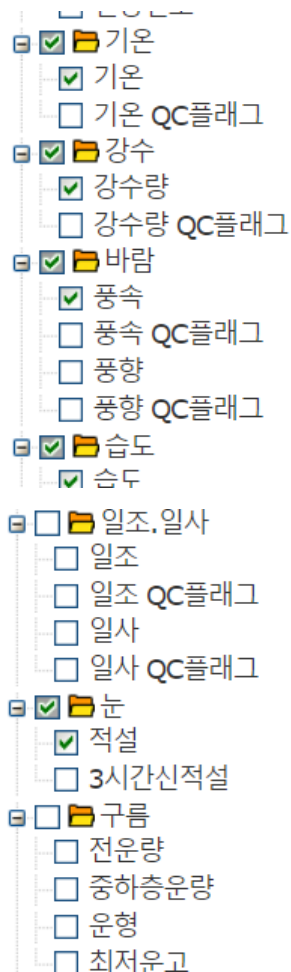
The screenshot shows the '기상자료개방포털[데이터:기상관]' (Weather Data Open API) website. The search criteria are as follows:

- 자료** (Data) tab is selected.
- 검색조건** (Search Conditions):
  - 자료형태** (Data Type): 시간 자료 (Time Data)
  - 기간** (Period): 20210101 01 ~ 20211231 00
  - 지점** (Location): 지도로 선택 (Select on Map)
  - 지점 목록** (Location List):
    - 경기도
    - 경상남도
    - 경상북도
    - 광주광역시
    - 대구광역시
    - 대전광역시
    - 부산광역시
    - 서울특별시** (highlighted with a red box)
      - 관악산 (116) 지점상세보기** (highlighted with a blue box)
      - 서울 (108) 지점상세보기** (highlighted with a blue box)
      - 세종특별자치시
      - 울산광역시
      - 인천광역시
      - 전라남도
- 자료보기** (View Data) buttons: CSV, Excel
- 데이터 개방 오픈 API** (Data Open API) logo is visible in the bottom left.

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

- 기온, 강수량, 풍속, 습도, 적설을 체크한 후 조회버튼을 클릭합니다.



## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

- 조회 화면이 나오면, CSV 버튼을 클릭해 다운로드를 받습니다.

data.kma.go.kr/data/grnd/selectAsosRltmList.do?pgmNo=36

지진화산

날씨 이슈별 데이터

역사기후

메타데이터

품질정보

데이터 개방  
오픈 API

☐ 세종특별자치시

☐ 울산광역시

☐ 인천광역시

☐ 전라남도

☒ 기온

☒ 기온

☐ 기온 QC플래그

> 조회

■ 자료보기

※조회 결과는 10건만 표출 됩니다. 상세결과는 파일 다운로드를 이용해주세요

지점	시간	기온(°C)	강수량(mm)	풍속(m/s)	습도(%)	적설(cm)
서울(108)	2022-01-01 01:00	-9.2		1.8	42	
서울(108)	2022-01-01 02:00	-9.5		1.2	43	
서울(108)	2022-01-01 03:00	-9.3		1.4	46	
서울(108)	2022-01-01 04:00	-9.6		1.7	48	
서울(108)	2022-01-01 05:00	-9.7		0.8	51	
서울(108)	2022-01-01 06:00	-9.8		1.9	53	
서울(108)	2022-01-01 07:00	-10.1		1.9	55	
서울(108)	2022-01-01 08:00	-10.2		2.5	56	
서울(108)	2022-01-01 09:00	-9.1		2.1	56	
서울(108)	2022-01-01 10:00	-6.8		2.3	42	

■ 평가하기: ○☆☆☆☆☆ ○☆☆☆☆☆ ○☆☆☆☆☆ ○☆☆☆☆☆ ●☆☆☆☆☆

등록

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

- 비회원으로 다운로드 받을 수 있습니다.

data.kma.go.kr/data/grnd/selectAsosRltmList.do?pgmNo=36

지진화산  
날씨 이슈별 데이터  
역사기후  
메타데이터  
품질정보

데이터 개방  
오픈 API



☐ 세종특별자치시  
☐ 울산광역시  
☐ 인천광역시  
☐ 전라남도

☒ 기온  
☒ 기온  
☐ 기온 QC플래그

> 조회

자료보기

CSV Excel

※ 조회 결과는 10건만 표출됩니다. 상세 결과는 파일 다운로드를 이용해주세요

로그인

생활기상정보와 해구별 예측자료는 비회원도 다운로드할 수 있습니다.  
비회원의 최대 다운로드 건수는 일(31), 시간(24)건입니다.

확인

이메일

비밀번호

☐ 아이디 저장

로그인

회원가입

아이디/비밀번호 찾기

서울(108)	2022-01-01 08:00	-10.2	2.5	56
서울(108)	2022-01-01 09:00	-9.1	2.1	56
서울(108)	2022-01-01 10:00	-6.8	2.3	42

## 2. Data Collection/Data Acquisition

기상청 : 기상자료개방포털

- 다운로드 받은 파일

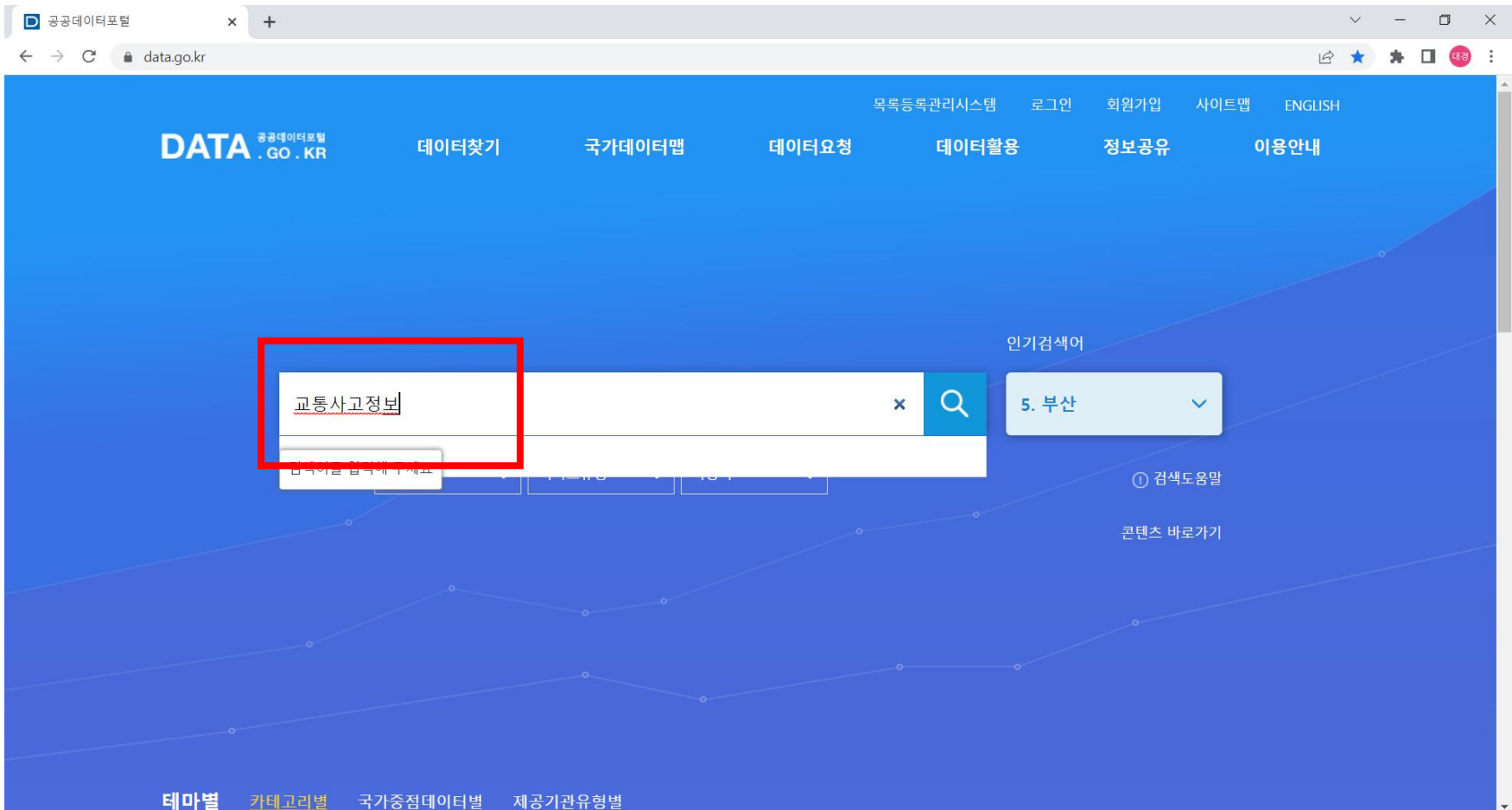
---

 OBS\_ASOS\_TIM\_20240524114001.csv

## 2. Data Collection/Data Acquisition

### 도로교통공단

- <https://www.data.go.kr/>에서 " 교통사고정보 " 라고 조회합니다.



The screenshot shows the homepage of the data.go.kr portal. The header includes the logo 'DATA .GO .KR' and navigation links: '데이터찾기', '국가데이터맵', '데이터요청', '데이터활용', '정보공유', and '이용안내'. A search bar is prominently displayed in the center, with the text '교통사고정보' entered and highlighted by a red rectangle. To the right of the search bar, there is a dropdown menu showing '5. 부산'. Below the search bar, there are links for '검색도움말' and '콘텐츠 바로가기'. The footer contains additional navigation links: '테마별', '카테고리별', '국가중점데이터별', and '제공기관유형별'.

## 2. Data Collection/Data Acquisition

### 도로교통공단

- " CSV JSON + XML 도로교통공단 사망 교통사고 정보 " 를 클릭합니다.

data.go.kr/tcs/dss/selectDataSetList.do?keyword=교통사고정보&brm=&svcType=&instt=&recmSe=N&conditionType=init&extsn=&kwrArray=

"교통사고정보"에 대해 총 37,320건이 검색되었습니다.

조건검색

초기화

분류체계	서비스유형	제공기관유형	확장자
국가중점데이터 분류 조건 추가하기 +			
조건열기			

전체(37,320건)	파일데이터(20,504건)	오픈 API(6,744건)	표준데이터셋202개(9,497건)	연계데이터(575건)
정확도순 5개씩 정렬				

'연계데이터'는 「공공데이터 관리지침」 제14조에 따라, 개별 포털로부터 수집되는 데이터입니다.

파일데이터 (20,504건)

더보기

재난안전

공공기관

미리보기

CSV

JSON + XML

도로교통공단\_사망 교통사고 정보

- 사망 교통사고에 대한 개별정보 제공(발생일시, 사고유형, 위치좌표 등)- 부상자수 = 중상자수 + 경상자수 + 부상신고자수

제공기관 도로교통공단 수정일 2023-07-25 조회수 28936 다운로드 9073 주기성 데이터 9 키워드 사망교통사고,개별정보,위치정보

다운로드



# 2. Data Collection/Data Acquisition

## 도로교통공단

- <https://www.data.go.kr/data/15070340/fileData.do> 에서

2022	2022-01-05 22:00	야	수	1	2	0	2	0	부산	동구	차대차
2022	2022-01-05 22:00	야	수	1	0	0	0	0	강원	횡성군	차대사람

### 주기성 과거 데이터 (8건)

도로교통공단_사망 교통사고 정보_20211231	등록
도로교통공단_사망 교통사고 정보_20201231	등록
도로교통공단_사망 교통사고 정보_20201123	등록
도로교통공단_사망 교통사고 정보_20190910	등록
도로교통공단_사망 교통사고 정보_20190910	등록

목록

## 2. Data Collection/Data Acquisition

### 도로교통공단

- <https://www.data.go.kr/data/15070340/fileData.do> 에서

data.go.kr/data/15070340/fileData.do#layer\_data\_information

차기 등록 예정일	2022-07-31
매체유형	텍스트
전체 행 수	2983
확장자	CSV
다운로드(바로가기) 횟수	1759
등록일	2021-06-18
수정일	2022-06-22
데이터 다운로드	<input type="button" value="CSV"/> <input type="button" value="JSON"/>
비용부과유무	무료
비용부과기준 및 단위	건
이용허락범위	<a href="#">이용허락범위 제한 없음</a>

데이터 개방문의 1566-00

[개인정보처리방침](#)



행정안전부

닫기

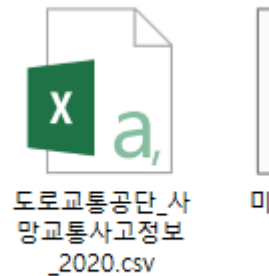
## 2. Data Collection/Data Acquisition

### Data Ingest

- 데이터 가져오기

 OBS_ASOS_TIM_20240523192641.csv	2024-05-23 오후 8:14	Microsoft Excel ...	
 도로교통공단_사망교통사고정보(2020).csv	2024-05-23 오후 7:51	Microsoft Excel ...	4€

도로교통공단\_사망교통사고정보\_2020.csv 이름 바꾸기



# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 날씨 데이터 가져오기

파일	소스코드							
실습환경	준비 Tf38_cpu							
	import pandas as pd							
소스코드	OBS_ASOS_TIM_20240524114001 = pd.read_csv('OBS_ASOS_TIM_20240524114001.csv', encoding='euc-kr') OBS_ASOS_TIM_20240524114001.head()							
결과값1	지점 0	지점명 108 NaN	일시 서울	기온(°C) 2020-01-01 01:00	강수량(mm) -5.9	풍속(m/s) NaN	습도(%) 1.7	적설(cm) 40
	1	108 NaN	서울	2020-01-01 02:00	-5.7	NaN	0.1	42
	2	108 NaN	서울	2020-01-01 03:00	-5.6	0.0	0.0	46
	3	108 NaN	서울	2020-01-01 04:00	-5.4	NaN	0.0	50
	4	108 NaN	서울	2020-01-01 05:00	-5.2	NaN	0.0	55
비고								

# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 교통사고 데이터 가져오기

파일	소스코드							
실습환경	준비 Tf38_cpu							
소스코드	도로교통공단_사망교통사고정보_2020 = pd.read_csv('도로교통공단_사망교통사고정보_2020.csv', encoding='euc-kr')  도로교통공단_사망교통사고정보_2020.head()							
결과값1	발생년	발생년월일시 부상신고자수 가해자_당사자종별	주야 발생지시도 피해자_당사자종별	요일 ... 발생위치X_UTMK	사망자수 사고유형 발생위치Y_UTMK	부상자수 가해자법규위반 경도	중상자수 도로형태_대분류 위도	경상자수 도로형태
	0	2020 0 기타단일로	2020-01-01 05 0 승용차	서울 보행자	... 959928.0 수	1 차도통행중 1943154.0	0 안전운전 의무 불이행 127.046727	0 단일로 37.486761
	1	2020 0 기타단일로	2020-01-01 05 0 승용차	경기 화물차	... 951237.0 수	1 추돌 1916995.0	0 안전운전 의무 불이행 126.950147	0 단일로 37.250560
	2	2020 0 기타단일로	2020-01-01 07 0 승용차	충남 보행자	... 967688.0 수	1 횡단중 1869920.0	0 안전운전 의무 불이행 127.137666	0 단일로 36.826933
	3	2020 0 기타단일로	2020-01-01 10 0 화물차	경남 없음	... 1038413.0 수	1 공작물충돌 1748968.0	0 안전운전 의무 불이행 127.924796	0 단일로 35.736290
	4	2020 0 기타단일로	2020-01-01 11 0 농기계	경기 없음	... 1011197.0	1 도로이탈 추락 1900537.0	0 안전운전 의무 불이행 127.626014	0 단일로 37.103415
	5 rows × 23 columns							
비고								

# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 필요한 차원 데이터 셋으로 저장하기-날씨 데이터

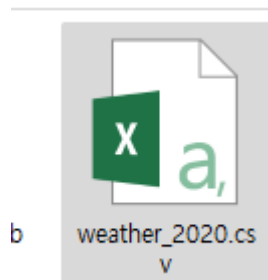
파일	소스코드						
실습환경	준비 Tf38_cpu						
소스코드	weather_2020 = OBS_ASOS_TIM_20240524114001.drop(['지점','지점명'], axis=1)						
	weather_2020.head()						
결과값1		일시	기온(°C)	강수량(mm)	풍속(m/s)	습도(%)	적설(cm)
	0	2020-01-01 01:00	-5.9	NaN	1.7	40	NaN
	1	2020-01-01 02:00	-5.7	NaN	0.1	42	NaN
	2	2020-01-01 03:00	-5.6	0.0	0.0	46	NaN
	3	2020-01-01 04:00	-5.4	NaN	0.0	50	NaN
	4	2020-01-01 05:00	-5.2	NaN	0.0	55	NaN
비고							

# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 필요한 차원 데이터 셋으로 저장하기-날씨 데이터

파일	소스코드						
실습환경	준비 Tf38_cpu						
소스코드	weather_2020.rename(columns={'일시':'년월일시'}, inplace=True)						
	weather_2020.head()						
	weather_2020.to_csv('weather_2020.csv', index=False)						
결과값1	년월일시	기온(°C)	강수량(mm)	풍속(m/s)	습도(%)	적설(cm)	
	0	2020-01-01 01:00	-5.9	NaN	1.7	40	NaN
	1	2020-01-01 02:00	-5.7	NaN	0.1	42	NaN
	2	2020-01-01 03:00	-5.6	0.0	0.0	46	NaN
	3	2020-01-01 04:00	-5.4	NaN	0.0	50	NaN
	4	2020-01-01 05:00	-5.2	NaN	0.0	55	NaN
비고							







# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 필요한 차원 데이터 셋으로 저장하기-교통사고 데이터

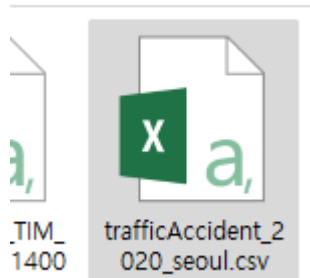
파일	소스코드																										
실습환경	준비 Tf38_cpu																										
소스코드	<pre>trafficAccident_2020_seoul = 도로교통공단_사망교통사고정보_2020[['발생년월일시','사망자수']]</pre> <pre>trafficAccident_2020_seoul</pre>																										
결과값1	<table><thead><tr><th>발생년월일시</th><th>사망자수</th></tr></thead><tbody><tr><td>0</td><td>2020-01-01 05 1</td></tr><tr><td>1</td><td>2020-01-01 05 1</td></tr><tr><td>2</td><td>2020-01-01 07 1</td></tr><tr><td>3</td><td>2020-01-01 10 1</td></tr><tr><td>4</td><td>2020-01-01 11 1</td></tr><tr><td>...</td><td>...</td></tr><tr><td>2978</td><td>2020-12-31 11 1</td></tr><tr><td>2979</td><td>2020-12-31 18 1</td></tr><tr><td>2980</td><td>2020-12-31 19 1</td></tr><tr><td>2981</td><td>2020-12-31 21 1</td></tr><tr><td>2982</td><td>2020-12-31 21 1</td></tr><tr><td colspan="2">2983 rows × 2 columns</td></tr></tbody></table>	발생년월일시	사망자수	0	2020-01-01 05 1	1	2020-01-01 05 1	2	2020-01-01 07 1	3	2020-01-01 10 1	4	2020-01-01 11 1	...	...	2978	2020-12-31 11 1	2979	2020-12-31 18 1	2980	2020-12-31 19 1	2981	2020-12-31 21 1	2982	2020-12-31 21 1	2983 rows × 2 columns	
발생년월일시	사망자수																										
0	2020-01-01 05 1																										
1	2020-01-01 05 1																										
2	2020-01-01 07 1																										
3	2020-01-01 10 1																										
4	2020-01-01 11 1																										
...	...																										
2978	2020-12-31 11 1																										
2979	2020-12-31 18 1																										
2980	2020-12-31 19 1																										
2981	2020-12-31 21 1																										
2982	2020-12-31 21 1																										
2983 rows × 2 columns																											
비고																											

## 2. Data Collection/Data Acquisition

### Data Ingest

- 필요한 차원 데이터 셋으로 저장하기-교통사고 데이터

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<code>trafficAccident_2020_seoul.to_csv('trafficAccident_2020_seoul.csv', index=False)</code>
결과값1	
비고	



# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 필요한 차원 데이터 셋으로 저장하기-교통사고 데이터

파일	소스코드																										
실습환경	준비 Tf38_cpu																										
소스코드	<pre>trafficAccident_2020_seoul = pd.read_csv('trafficAccident_2020_seoul.csv')  trafficAccident_2020_seoul.head()</pre>																										
결과값1	<table><tr><th>발생년월일시</th><th>사망자수</th></tr><tr><td>0</td><td>2020-01-01 05 1</td></tr><tr><td>1</td><td>2020-01-01 05 1</td></tr><tr><td>2</td><td>2020-01-01 07 1</td></tr><tr><td>3</td><td>2020-01-01 10 1</td></tr><tr><td>4</td><td>2020-01-01 11 1</td></tr><tr><td>...</td><td>...</td></tr><tr><td>2978</td><td>2020-12-31 11 1</td></tr><tr><td>2979</td><td>2020-12-31 18 1</td></tr><tr><td>2980</td><td>2020-12-31 19 1</td></tr><tr><td>2981</td><td>2020-12-31 21 1</td></tr><tr><td>2982</td><td>2020-12-31 21 1</td></tr><tr><td colspan="2">2983 rows × 2 columns</td></tr></table>	발생년월일시	사망자수	0	2020-01-01 05 1	1	2020-01-01 05 1	2	2020-01-01 07 1	3	2020-01-01 10 1	4	2020-01-01 11 1	...	...	2978	2020-12-31 11 1	2979	2020-12-31 18 1	2980	2020-12-31 19 1	2981	2020-12-31 21 1	2982	2020-12-31 21 1	2983 rows × 2 columns	
발생년월일시	사망자수																										
0	2020-01-01 05 1																										
1	2020-01-01 05 1																										
2	2020-01-01 07 1																										
3	2020-01-01 10 1																										
4	2020-01-01 11 1																										
...	...																										
2978	2020-12-31 11 1																										
2979	2020-12-31 18 1																										
2980	2020-12-31 19 1																										
2981	2020-12-31 21 1																										
2982	2020-12-31 21 1																										
2983 rows × 2 columns																											
비고																											

# 3. Data Exploration

## Data Preprocessing

### ● 교통사고 데이터 셋 전처리

파일	소스코드												
실습환경	준비 Tf38_cpu												
소스코드	<pre>trafficAccident_2020_seoul = pd.read_csv('trafficAccident_2020_seoul.csv')  trafficAccident_2020_seoul.head()</pre>												
결과값1	<table><tr><th>발생년월일시</th><th>사망자수</th></tr><tr><td>0</td><td>2020-01-01 05 1</td></tr><tr><td>1</td><td>2020-01-01 05 1</td></tr><tr><td>2</td><td>2020-01-01 07 1</td></tr><tr><td>3</td><td>2020-01-01 10 1</td></tr><tr><td>4</td><td>2020-01-01 11 1</td></tr></table>	발생년월일시	사망자수	0	2020-01-01 05 1	1	2020-01-01 05 1	2	2020-01-01 07 1	3	2020-01-01 10 1	4	2020-01-01 11 1
발생년월일시	사망자수												
0	2020-01-01 05 1												
1	2020-01-01 05 1												
2	2020-01-01 07 1												
3	2020-01-01 10 1												
4	2020-01-01 11 1												
비고													

# 3. Data Exploration

## Data Preprocessing

### ● 교통사고 데이터 셋 전처리

파일	소스코드
실습환경	<pre>준비 Tf38_cpu  trafficAccident2019_seoul['발생년월일시']=trafficAccident2019_seoul['발생년월일시'].apply(lambda x: pd.to_datetime(str(x), format='%Y-%m-%d %H')) trafficAccident2019_seoul.head()</pre>
소스코드	<pre>trafficAccident2019_seoul['발생년월일시']=pd.to_datetime(trafficAccident2019_seoul['발생년월일시']).dt.floor('H') trafficAccident2019_seoul.head()  trafficAccident2019_seoul.rename(columns={'발생년월일시':'년월일시'}, inplace=True) trafficAccident2019_seoul.head()  trafficAccident2019_seoul.set_index(trafficAccident2019_seoul['년월일시'], inplace=True) trafficAccident2019_seoul.head()  trafficAccident2019_seoul.drop('년월일시', axis=1) trafficAccident2019_seoul.head()</pre>
결과값1	
비고	

# 3. Data Exploration

## Data Preprocessing

### ● 데이터 가공

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<pre>import numpy as np  trafficAccident2019_seoul=trafficAccident2019_seoul.resample('H').apply({'사망자수':np.sum}, axis=1).fillna(0) trafficAccident2019_seoul.head()</pre>
결과값1	사망자수 년월일시 2019-01-01 00:00:00 1 2019-01-01 01:00:00 0 2019-01-01 02:00:00 0 2019-01-01 03:00:00 1 2019-01-01 04:00:00 0
비고	



# 4. Data Preparation

## Data Preparation

### ● 데이터 준비

파일	소스코드
실습환경	준비 <code>Tf38_cpu</code>
소스코드	<code>weather_2019.head()</code> <code>trafficAccident2019_seoul.head()</code>
결과값1	기온(°C) 강수량(mm) 풍속(m/s) 습도(%) 적설(cm) 년월일시 2019-01-01 01:00 -5.9 NaN 1.8 56.0 NaN 2019-01-01 02:00 -6.5 NaN 1.2 60.0 NaN 2019-01-01 03:00 -6.9 NaN 2.2 62.0 NaN 2019-01-01 04:00 -7.2 NaN 1.2 57.0 NaN 2019-01-01 05:00 -7.6 NaN 2.1 59.0 NaN
결과값2	사망자수 년월일시 2019-01-01 00:00:00 1 2019-01-01 01:00:00 0 2019-01-01 02:00:00 0 2019-01-01 03:00:00 1 2019-01-01 04:00:00 0
비고	

# 4. Data Preparation

## Data Preparation

### ● 데이터 준비

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<pre>preparation_data=pd.merge(     weather_2019,     trafficAccident2019_seoul,     how='outer',     left_index=True,     right_index=True ) preparation_data.head()</pre>
결과값1	<p>기온(°C) 강수량(mm) 풍속(m/s) 습도(%) 적설(cm) 사망자수 년월일시</p> <pre>2019-01-01 00:00:00 NaN NaN NaN NaN NaN 1.0 2019-01-01 01:00:00 NaN NaN NaN NaN NaN 0.0 2019-01-01 02:00:00 NaN NaN NaN NaN NaN 0.0 2019-01-01 03:00:00 NaN NaN NaN NaN NaN 1.0 2019-01-01 04:00:00 NaN NaN NaN NaN NaN 0.0</pre>
비고	





# 4. Data Preparation

## Data Preparation

### ● 데이터 준비

파일	소스코드
실습환경	<pre>준비 Tf38_cpu</pre>
소스코드	<pre>preparation_data=preparation_data.fillna(o) preparation_data.head()  preparation_data['사망자수']=preparation_data['사망자수'].astype(int) preparation_data.head()  preparation_data.info()</pre>
결과값 <sup>1</sup>	<pre>&lt;class 'pandas.core.frame.DataFrame'&gt; Index: 17489 entries, 2019-01-01 00:00:00 to 2019-12-31 00:00:00 Data columns (total 6 columns): #   Column    Non-Null Count  Dtype ---  - 0   기온(°C)   17489 non-null float64 1   강수량(mm) 17489 non-null float64 2   풍속(m/s)  17489 non-null float64 3   습도(%)    17489 non-null float64 4   적설(cm)   17489 non-null float64 5   사망자수   17489 non-null int64 dtypes: float64(5), int64(1) memory usage: 956.4+ KB</pre>
비고	



# 4. Data Preparation

## Data Preparation

### ● 데이터 준비

파일	소스코드
실습환경	준비 Tf38_cpu
	<code>preparation_data.to_csv('preparation_data.csv', index=False)</code>
소스코드	<code>import os</code> <code>os.getcwd()</code>  <code>os.chdir("")</code>  <code>os.listdir()</code>
결과값1	
비고	

# 5. Learning

## Learning

### ● 라이브러리 가져오기

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<pre>import pandas as pd import numpy as np #import tensorflow as tf import tensorflow.compat.v1 as tf tf.disable_v2_behavior()  preparation_data=pd.read_csv('preparation_data.csv') preparation_data.head()</pre>
결과값1	기온(°C) 강수량(mm) 풍속(m/s) 습도(%) 적설(cm) 사망자수 0 0.0 0.0 0.0 0.0 0.0 0.0 1 1 0.0 0.0 0.0 0.0 0.0 0.0 0 2 0.0 0.0 0.0 0.0 0.0 0.0 0 3 0.0 0.0 0.0 0.0 0.0 0.0 1 4 0.0 0.0 0.0 0.0 0.0 0.0 0
비고	

# 5. Learning

## Learning

### ● 데이터 나누기

파일	소스코드
실습환경	준비 <code>Tf38_cpu</code>
소스코드	<pre>xy=np.array(preparation_data, dtype=np.float32) print(xy)  x_data=xy[:, :-1] y_data=xy[:, -1:] print(x_data) print(y_data)</pre>
결과값1	<pre>[[ 0.  0.  0.  0.  0.]  [ 0.  0.  0.  0.  0.]  [ 0.  0.  0.  0.  0.]  ...  [-3.6  0.  4.748  0.]  [-4.3  0.  5.552  0.]  [-5.7  0.  7.466  0.]] [[1.]  [0.]  [0.]  ...  [0.]  [0.]  [0.]]</pre>
비고	

# 5. Learning

## Learning

### ● 학습

파일	소스코드
실습환경	<pre>준비 Tf38_cpu  X=tf.placeholder(tf.float32, shape=[None, 5]) y=tf.placeholder(tf.float32, shape=[None, 1])  W=tf.Variable(tf.random_normal([5,1]), name="weight") b=tf.Variable(tf.random_normal([1]), name="bias")  hypothesis=tf.matmul(X, W) + b</pre>
소스코드	<pre>cost=tf.reduce_mean(tf.square(hypothesis-y))  optimizer=tf.train.GradientDescentOptimizer(learning_rate=0.000005) train=optimizer.minimize(cost)  sess=tf.Session() init=tf.global_variables_initializer() sess.run(init)</pre>
결과값1	
비고	

# 5. Learning

## Learning

### ● 예측

파일	소스코드
실습환경	<pre>준비 Tf38_cpu</pre>
소스코드	<pre>for step in range(100000):      cost_opti, hypo_opti, _opti=sess.run(         [cost, hypothesis, train],         feed_dict={X: x_data, y: y_data}     )      step=step+1      if step % 500 == 0:         print("\n step(학습 횟수) :", step, "\n cost(예측값과 실제값의 편차) :",cost_opti)         print("# 예측되는 사망자 수 : ", hypo_opti[o])</pre>
결과값1	<pre>step(학습 횟수) : 99000 cost(예측값과 실제값의 편차) : 0.70900816 # 예측되는 사망자 수 : [0.10489665]  step(학습 횟수) : 99500 cost(예측값과 실제값의 편차) : 0.7073289 # 예측되는 사망자 수 : [0.10556816]  step(학습 횟수) : 100000 cost(예측값과 실제값의 편차) : 0.7056548 # 예측되는 사망자 수 : [0.10623872]</pre>
비고	



# 6. Evaluate



## | 평가

- ...

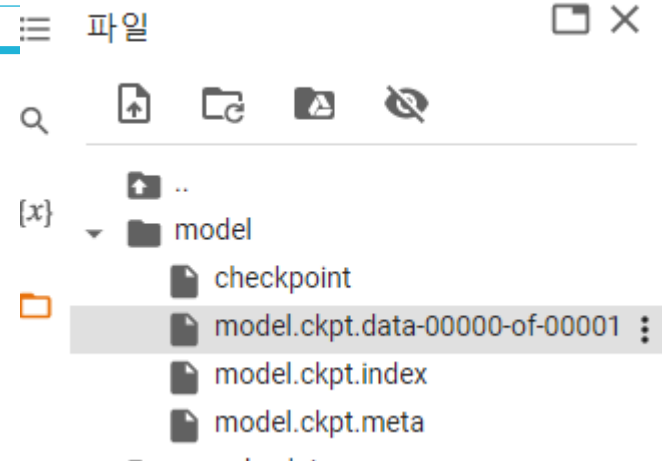
# 7. Deployment

## 학습 모델 저장

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<pre>saver=tf.train.Saver() save_path=saver.save(sess, "./model/model.ckpt")  print("\n ##### 학습된 모델을 저장하였습니다. #####")</pre>

결과값1

비고





# 7. Deployment

## 서비스하기

파일	소스코드
실습환경	<pre>준비 Tf38_cpu  import numpy as np #import tensorflow as tf import tensorflow.compat.v1 as tf tf.disable_v2_behavior()  tf.reset_default_graph()  X=tf.placeholder(tf.float32, shape=[None, 5]) y=tf.placeholder(tf.float32, shape=[None, 1])  W=tf.Variable(tf.random_normal([5,1]), name="weight") b=tf.Variable(tf.random_normal([1]), name="bias")  hypothesis=tf.matmul(X, W) + b  saver=tf.train.Saver() init=tf.global_variables_initializer()</pre>
결과값1	
비고	

# 7. Deployment

## 서비스하기

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<pre>temp=float(input('온 도 : ')) rain=float(input('강수량 : ')) windflow=float(input('풍 속 : ')) humidity=float(input('습 도 : ')) snowfall=float(input('적 설 : '))</pre>
결과값1	<pre>온 도 : 15 강수량 : 30 풍 속 : 20 습 도 : 50 적 설 : 0</pre>
비고	

# 7. Deployment

## 서비스하기

파일	소스코드
실습환경	<pre>준비 Tf38_cpu  with tf.Session() as sess:     sess.run(init)     save_path="./model/model.ckpt"     saver.restore(sess, save_path)</pre>
소스코드	<pre>data=((temp, rain, windflow, humidity, snowfall),) arr=np.array(data, dtype=np.float32)  x_data=arr[:] dict=sess.run(hypothesis, feed_dict={X: x_data})  print(dict[o])</pre>
결과값1	<pre>INFO:tensorflow:Restoring parameters from ./model/model.ckpt [-37.981976]</pre>
비고	