

○○ 마이스터 고등학교

# 파이썬 활용 AI 프로그래밍

『미니프로젝트』

2024. 05. 20. - 2024. 05. 24

Prepared by DaeKyeong Kim

Ph.D.

# 미니프로젝트 lab-1

## 날씨에 따른 교통사고 발생 확률 예측



## 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

---

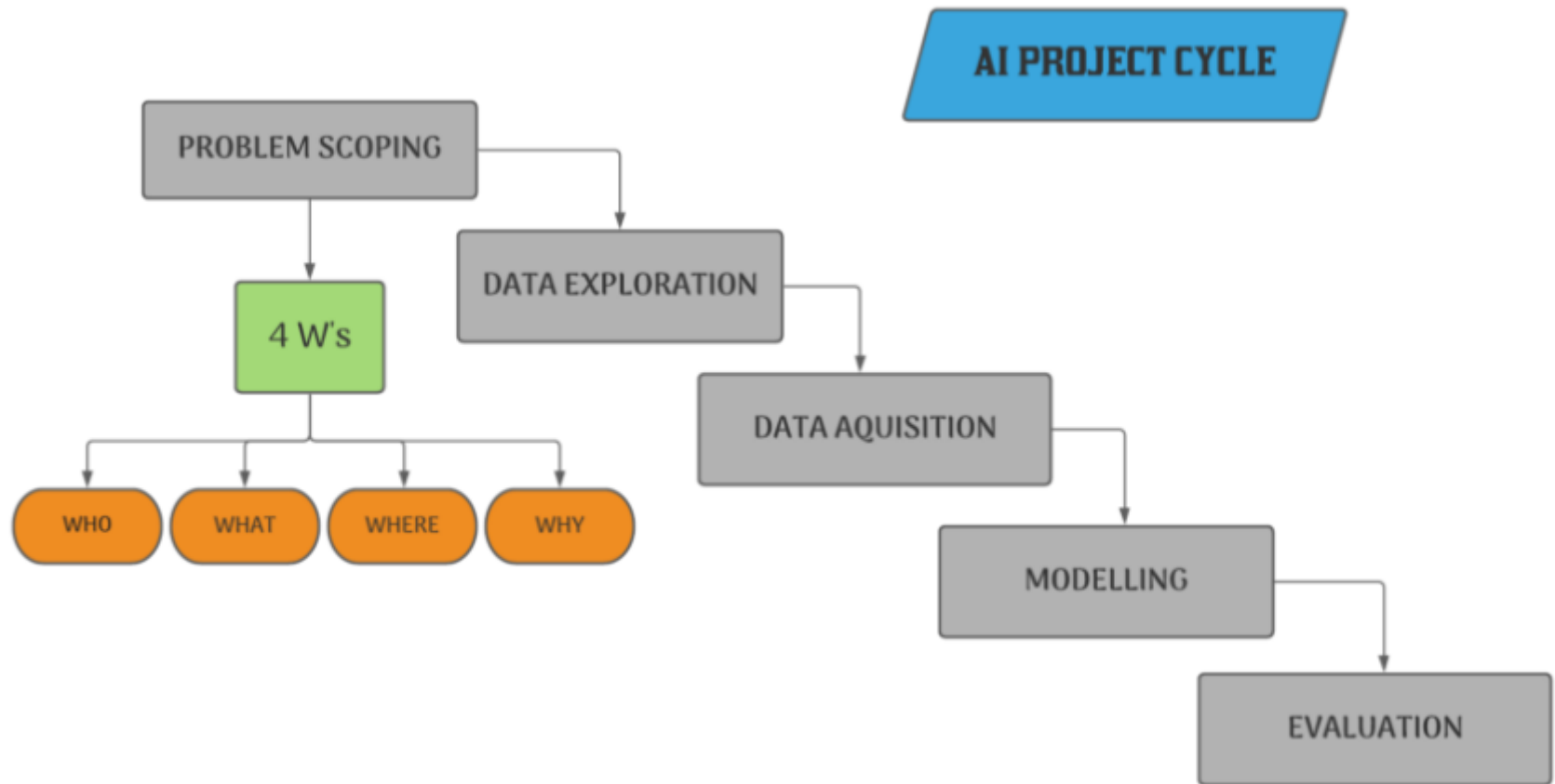
6. Deployment

---



# 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>

The screenshot shows the homepage of the Korea Meteorological Administration's Open Data Portal. The browser address bar displays [data.kma.go.kr/cmmn/main.do](https://data.kma.go.kr/cmmn/main.do). The page header includes the KMA logo, the text "기상청 날씨데이터 서비스", and navigation links for "국가기후데이터센터 소개", "로그인", "사이트맵", "즐거찾기", and "ENG(info)". The main title "기상자료개방포털" is prominently displayed. Below the title is a search bar with the placeholder text "'관측'을 검색하세요" and a magnifying glass icon. To the right of the search bar is a dropdown menu labeled "인기검색어" with a downward arrow and a magnifying glass icon. The navigation bar at the top contains links for "기상자료개방포털이란?", "데이터", "기후통계분석", "간행물", "소통과 참여", and a language selector "ALL". The main content area features a grid of service tiles: "데이터 전체보기" (Data Overview), "관측" (Observation) with a weather station icon, "예·특보" (Forecast & Special Notice) with a map of Korea, "대용량" (Large Volume) with a satellite icon, "기상위성" (Weather Satellite), "수치모델" (Numerical Model), "기상레이더" (Weather Radar), "지진화산" (Earthquake & Volcano) with a volcano icon, "기후통계" (Climate Statistics) with a bar chart icon, and "지도로 찾기" (Find on Map) with a map icon. A large banner on the right side of the main content area reads "날씨! 데이터가 되다" (Weather! Becomes Data) and "OPEN API" with a cloud icon.

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

#### ● Home 데이터기상관측지상종관기상관측(ASOS)

The screenshot shows the homepage of the Korea Meteorological Administration's Open Data Portal (ASOS). The browser address bar displays 'data.kma.go.kr/cmmn/main.do'. The page features a search bar with the text '관측' (Observation) and a search button. Below the search bar, there are navigation tabs: '기상자료개방포털이란?' (What is the Open Data Portal?), '데이터' (Data), '기후통계분석' (Climate Statistics Analysis), '간행물' (Publications), and '소통과 참여' (Communication and Participation). The '데이터' tab is selected. The main content area is divided into four columns, each with a list of data categories and sub-categories. The left column is titled '기상관측' (Weather Observation) and includes sub-categories like '지상' (Surface), '해양' (Marine), '고층' (Upper Air), '항공' (Aviation), and '세계기상전문(GTS)'. The second column is titled '기상위성' (Weather Satellite) and includes '천리안 위성 1호' (Cheonri-an Satellite 1) and '천리안 위성 2A호' (Cheonri-an Satellite 2A). The third column is titled '레이더' (Radar) and includes '사이트' (Site) and '합성' (Synthesis). The fourth column is titled '기상예보' (Weather Forecast) and includes '동네예보' (Neighborhood Forecast), '기상특보' (Weather Special Report), '태풍예보' (Typhoon Forecast), and '영향예보' (Impact Forecast). The bottom of the page shows a URL: 'https://data.kma.go.kr/data/grnd/selectAsosRltmList.do'.

기상청 날씨데이터 서비스

# 기상자료개방포털

'관측'을 검색하세요

인기검색어

기상자료개방포털이란?   **데이터**   기후통계분석   간행물   소통과 참여

- > 기상관측
  - 지상
  - 해양
  - 고층
  - 항공
  - 세계기상전문(GTS)
- > 수치모델
  - 수치분석알기도
  - 단-중기예측
  - 초단기예측
  - 파랑모델
- > 날씨 이슈별 데이터
  - 폭염

https://data.kma.go.kr/data/grnd/selectAsosRltmList.do

- > 기상위성
  - 천리안 위성 1호
  - 천리안 위성 2A호
- > 기후
  - 기후변화감시
  - 가뭄
- > 역사기후
  - 자기기록지
  - 종이일기도

- > 레이더
  - 사이트
  - 합성
- > 응용기상
  - 기상지수
  - 기상자원지도
- > 메타데이터
  - 지정정보
  - 통계정보

- > 기상예보
  - 동네예보
  - 기상특보
  - 태풍예보
  - 영향예보
- > 지진화산
  - 지진화산 특·정보
- > 품질정보
  - 품질현황
  - 데이터품질리포트

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

#### ● Home 데이터기상관측지상종관기상관측(ASOS)

기상자료개방포털[데이터:기상관] x +

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

고층 +  
항공 +  
세계기상전문(GTS) +  
기상위성 >  
레이더 >  
기상예보 >  
수치모델 >  
기후 >  
응용기상 >  
지진화산 >  
날씨 이슈별 데이터 >  
역사기후 >  
메타데이터 >  
품질정보 >

데이터 개방  
오픈 API

자료 파일셋 캘린더 OPEN-API

검색조건

자료형태 시간 자료 기간 20210101 01 ~ 20211231 00

지점 지도로 선택

경기도  
경상남도  
경상북도  
광주광역시  
대구광역시  
대전광역시  
부산광역시  
서울특별시  
관악산 (116) 지점상세보기  
서울 (108) 지점상세보기  
세종특별자치시  
울산광역시  
인천광역시  
전라남도

전체  
지면.초상온도  
지면온도  
지면온도 QC플래그  
5cm 지중온도  
10cm 지중온도  
20cm 지중온도  
30cm 지중온도  
현상번호  
현상번호  
기온  
기온 OC플래그

> 조회

자료보기 CSV Excel

지점	시간
----	----

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

#### ● Home 데이터기상관측지상종관기상관측(ASOS)

- ☒ ☒ 기온
  - ☒ 기온
  - ☐ 기온 QC플래그
- ☒ ☒ 강수
  - ☒ 강수량
  - ☐ 강수량 QC플래그
- ☒ ☒ 바람
  - ☒ 풍속
  - ☐ 풍속 QC플래그
  - ☐ 풍향
  - ☐ 풍향 QC플래그
- ☒ ☒ 습도
  - ☒ 습도
- ☐ ☐ 일조, 일사
  - ☐ 일조
  - ☐ 일조 QC플래그
  - ☐ 일사
  - ☐ 일사 QC플래그
- ☒ ☒ 눈
  - ☒ 적설
  - ☐ 3시간신적설
- ☐ ☐ 구름
  - ☐ 전운량
  - ☐ 중하층운량
  - ☐ 운형
  - ☐ 최저유고

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

#### ● Home 데이터기상관측지상종관기상관측(ASOS)

■ 자료보기

CSV

Excel

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

지점	시간	기온(°C)	강수량(mm)	풍속(m/s)	습도(%)	적설(cm)
서울(108)	2021-01-01 01:00	-8.7		2.4	68	
서울(108)	2021-01-01 02:00	-9.1		1.6	69	
서울(108)	2021-01-01 03:00	-9.3		1.1	70	
서울(108)	2021-01-01 04:00	-9.3		0.3	71	
서울(108)	2021-01-01 05:00	-9.7		1.9	72	
서울(108)	2021-01-01 06:00	-9.7		2	75	
서울(108)	2021-01-01 07:00	-9.3		1.6	71	
서울(108)	2021-01-01 08:00	-9.3		1.6	72	
서울(108)	2021-01-01 09:00	-8.6		2.5	74	
서울(108)	2021-01-01 10:00	-6.1		1.1	68	

## 2. Data Collection/Data Acquisition

### 기상청 : 기상자료개방포털

#### ● Home 데이터기상관측지상종관기상관측(ASOS)

기상자료개방포털[데이터:기상관] ×

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

날씨 이슈별 데이터  
역사기후  
메타데이터  
품질정보

데이터 개방  
오픈 API

용도신청

해당 자료의 사용분야를 선택해 주세요.

☐ 농업 ☐ 축산업 ☐ 어업 ☐ 광업  
☐ 인쇄/출판 ☐ 전기/통신 ☐ 법률/보험 ☐ 학술/연구  
☐ 제조업 ☐ 운송업 ☐ 서비스/영업 ☐ 토목/건축  
☐ 스포츠/레저 ☐ 보건/의료 ☒ 교육/행정 ☐ 환경/정화  
☐ 임업

취소 신청

	/s	습도(%)	적설(cm)
		68	
		69	
		70	
		71	
		72	
서울(108)	2021-01-01 05:00	-9.7	1.9
서울(108)	2021-01-01 06:00	-9.7	2
서울(108)	2021-01-01 07:00	-9.3	1.6
서울(108)	2021-01-01 08:00	-9.3	1.6
서울(108)	2021-01-01 09:00	-8.6	2.5
서울(108)	2021-01-01 10:00	-6.1	1.1

평가하기

# 2. Data Collection/Data Acquisition

## 도로교통공단

- <https://www.data.go.kr/>

The screenshot shows the homepage of the data.go.kr website. The header includes the logo 'DATA GO.KR' and navigation links: '데이터찾기', '국가데이터맵', '데이터요청', '데이터활용', '정보공유', and '이용안내'. A search bar is prominently displayed with the text '교통사고정보' (Traffic Accident Information) entered. To the right of the search bar is a dropdown menu showing '5. 부산' (Busan). Below the search bar, there is a提示 '검색어를 입력해 주세요' (Please enter a search term). On the right side of the page, there are links for '인기검색어' (Popular Search Terms), '① 검색도움말' (Search Help), and '콘텐츠 바로가기' (Content Direct Link). The footer contains the text '테마별' (By Theme) followed by '카테고리별' (By Category), '국가중점데이터별' (By National Key Data), and '제공기관유형별' (By Provider Type).

# 2. Data Collection/Data Acquisition

## 도로교통공단

- [https://www.data.go.kr/data/15070340/fileData.do#layer\\_data\\_infomation](https://www.data.go.kr/data/15070340/fileData.do#layer_data_infomation)

데이터 상세 | 공공데이터포털

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

2020	2020-01-01 20	야	수	1	0	0	0	0	경기	오산시	차대사람	횡단
2020	2020-01-01 21	야	수	1	0	0	0	0	부산	기장군	차량단독	공작
2020	2020-01-02 02	야	목	1	0	0	0	0	충남	천안시	차량단독	도로

주기성 과거 데이터 (6건) 더보기 닫기

도로교통공단_사망 교통사고 정보_20201123	등록일 : 2020-11-23
도로교통공단_사망 교통사고 정보_20190910	등록일 : 2019-09-10
도로교통공단_사망 교통사고 정보_20190910	등록일 : 2019-09-10
도로교통공단_사망 교통사고 정보_20180601	등록일 : 2018-06-01
도로교통공단_사망 교통사고 정보_20180111	등록일 : 2018-01-11

목록

공공데이터 개방문의 1566-0025 운영자 메일상담

f t b

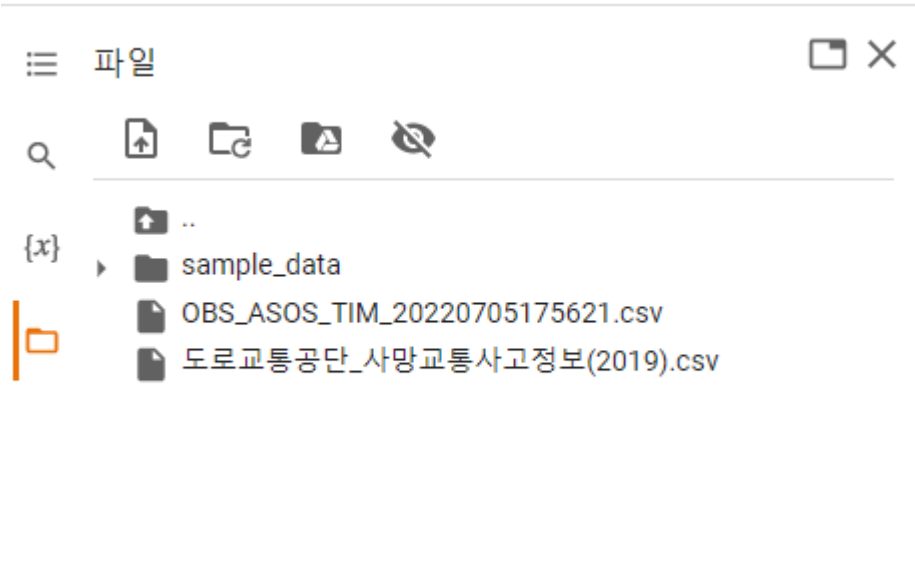
도로교통공단\_전국....csv 도로교통공단\_전국....csv 도로교통공단\_사망....csv 도로교통공단\_사망....csv 모두 표시



# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 데이터 가져오기



# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 데이터 가져오기

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<pre>import pandas as pd  OBS_ASOS_TIM_20220705175621 = pd.read_csv('OBS_ASOS_TIM_20220705175621.csv',encoding='euc-kr') OBS_ASOS_TIM_20220705175621.head()</pre>
결과값1	지점 지점명 일시 기온(°C) 강수량(mm) 풍속(m/s) 습도(%) 적설(cm) 0 108 서울 2019-01-01 01:00 -5.9 NaN 1.8 56.0 NaN 1 108 서울 2019-01-01 02:00 -6.5 NaN 1.2 60.0 NaN 2 108 서울 2019-01-01 03:00 -6.9 NaN 2.2 62.0 NaN 3 108 서울 2019-01-01 04:00 -7.2 NaN 1.2 57.0 NaN 108 서울 2019-01-01 05:00 -7.6 NaN 2.1 59.0 NaN
비고	

# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 데이터 가져오기

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<pre>도로교통공단_2019 = pd.read_csv('도로교통공단_2019.csv',encoding='euc-kr') 도로교통공단_2019.head()</pre>
결과값1	<pre>발생년  발생년월일시 주야 요일 사망자수 부상자수 중상자수 경상자수 부상신고자수 발생지시도 ... 사고유형 가해자법규위반 도로형태_대분류 도로형태 가해자_당사자종별 피해자_당사자종별 발생위치X_UTMK 발생위치Y_UTMK 경도 위도 0 2019 2019-01-01 00 야 화 1 0 0 0 0 부산 ... 기타 중앙선 침범 교차로 교차로내 승용차 보행자 1150507.0 1685987.0 129.152465 35.157884 1 2019 2019-01-01 03 야 화 1 0 0 0 0 경기 ... 정면충돌 중앙선 침범 단일로 기타단일로 승용차 승용차 943018.0 1921233.0 126.857149 37.288292 2 2019 2019-01-01 16 주 화 1 0 0 0 0 경북 ... 측면충돌 안전운전 의무 불이행 교차로 교차로부근 승용차 사륜오토바이(ATV) 1100414.0 1739893.0 128.609230 35.650108 3 2019 2019-01-01 19 야 화 1 0 0 0 0 대전 ... 횡단중 안전운전 의무 불이행 교차로 교차로부근 승용차 보행자 989194.0 1817621.0 127.379560 36.355946 4 2019 2019-01-01 21 야 화 1 0 0 0 0 경북 ... 기타 안전운전 의무 불이행 단일로 기타단일로 승용차 보행자 1127700.0 1767912.0 128.915041 35.899506 5 rows x 23 columns</pre>
비고	



# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 데이터 가져오기

파일	소스코드
실습환경	준비 Tf38_cpu
소스코드	<b>trafficAccident2019_seoul=도로교통공단_2019[['발생년월일시','사망자수']] trafficAccident2019_seoul</b>
결과값1	발생년월일시 사망자수 0 2019-01-01 00 1 1 2019-01-01 03 1 2 2019-01-01 16 1 3 2019-01-01 19 1 4 2019-01-01 21 1 ... .. 3228 2019-12-31 05 1 3229 2019-12-31 08 1 3230 2019-12-31 12 1 3231 2019-12-31 12 1 3232 2019-12-31 16 1 3233 rows x 2 columns`
비고	

# 2. Data Collection/Data Acquisition

## Data Ingest

### ● 데이터 가져오기

파일	소스코드
실습환경	<pre>준비 Tf38_cpu  trafficAccident2019_seoul.to_csv('trafficAccident2019_seoul.csv',index=False)  trafficAccident2019_seoul = pd.read_csv('trafficAccident2019_seoul.csv') trafficAccident2019_seoul.head()</pre>
소스코드	<pre>weather_2019=OBS_ASOS_TIM_20220705175621.drop(['지점','지점명'],axis=1) weather_2019.head()  weather_2019.rename(columns={'일시':'년월일시'}, inplace=True) weather_2019.head()  weather_2019.to_csv('weather_2019.csv', index=False)</pre>
결과값1	
비고	



# 3. Data Exploration

## Data Preprocessing

### ● 데이터 가공

파일	소스코드
실습환경	<pre>준비 Tf38_cpu  trafficAccident2019_seoul = pd.read_csv('trafficAccident2019_seoul.csv') trafficAccident2019_seoul.head()  trafficAccident2019_seoul['발생년월일시']=trafficAccident2019_seoul['발생년월일시'].apply(lambda x: pd.to_datetime(str(x), format='%Y-%m-%d %H')) trafficAccident2019_seoul.head()  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[0])</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

- ...



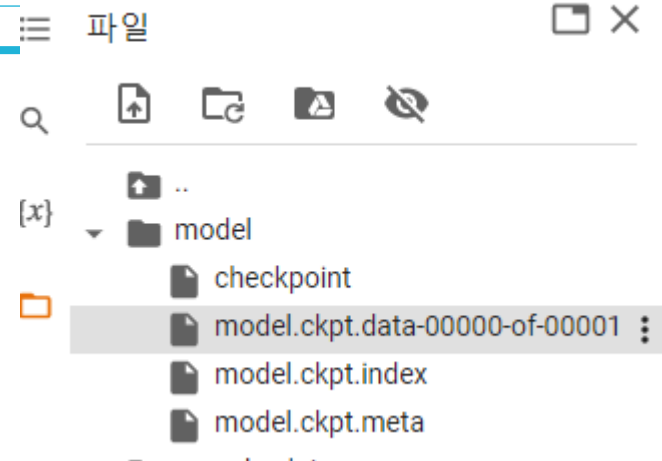
# 7. Deployment

## 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	