

- 이진규의 Profile
 - 1. 개인정보
 - 1.1. 프로필
 - 1.2. 나와 데이터의 Story
 - 1.3. 학력과 교육이력
 - 2. Skill (Language & Data)
 - 2.1. Python
 - 2.1.1. Python 기본
 - 2.1.2. Python 패키지 사용
 - 2.1.3. 머신러닝
 - 2.1.4. 딥러닝
 - 2.1.5 LLM
 - 2.1.6 etc
 - 2.2 SQL
 - 2.3. etc
 - 2.3.1. Streamlit 기반 웹 API 구현
 - 2.3.2. 스프레드시트
 - 3. Skill (Environment)
 - 3.1. Windows
 - 3.1.1. Python 환경 구축
 - 3.2. Linux
 - 3.2.1. Python 환경 구축
 - 3.2.2. Jupyter Server와 SSH 원격 접속 환경 구축
 - 3.2.3. Docker
 - 3.3. etc
 - 3.3.1 편의성을 고려한 개발환경 구축
 - 4. Project
 - 4.1. 데이터 수집 프로젝트
 - 4.2. 탐색적 데이터 분석 프로젝트
 - 4.3. 머신러닝 / 딥러닝 프로젝트
- 이진규의 Profile
 - 1. 개인정보
 - 1.1. 프로필
 - 1.2. 나와 데이터의 Story
 - 1.3. 학력과 교육이력
 - 2. Skill (Language & Data)

- 2.1. Python
 - 2.1.1. Python 기본
 - 2.1.2. Python 패키지 사용
 - 2.1.3. 머신러닝
 - 2.1.4. 딥러닝
 - 2.1.5 LLM
 - 2.1.6 etc
- 2.2 SQL
- 2.3. etc
 - 2.3.1. Streamlit 기반 웹 API 구현
 - 2.3.2. 스프레드시트
- 3. Skill (Environment)
 - 3.1. Windows
 - 3.1.1. Python 환경 구축
 - 3.2. Linux
 - 3.2.1. Python 환경 구축
 - 3.2.2. Jupyter Server와 SSH 원격 접속 환경 구축
 - 3.2.3. Docker
 - 3.3. etc
 - 3.3.1 편의성을 고려한 개발환경 구축
- 4. Project
 - 4.1. 데이터 수집 프로젝트
 - 4.2. 탐색적 데이터 분석 프로젝트
 - 4.3. 머신러닝 / 딥러닝 프로젝트

이진규의 Profile

1. 개인정보

1.1. 프로필

이름 : 이진규

연락처 : 010-9799-4175

생년월일 : 1991-10-05

1.2. 나와 데이터의 Story

어린 시절을 생각해보면 컴퓨터 환경은 생각보다 빠르게 접했습니다. 기억에 남는 선에서는 펜티엄3 컴퓨터 + 윈도우 95 + MS-DOS 환경을 최초로 사용했었고(더 이전에는 486이 있었던것 같기도 합니다), MS-DOS의 Mdir을 활용했던 경험도 어렴풋이 남아 있습니다. 이러한 영향 때문인지 현재에도 개인적으로 사용하는 컴퓨터 환경은 늘 최신 및 고사양으로 유지하고 있습니다.

놀이 좋아했던 어린 시절인 만큼 컴퓨터는 80%가 게임 및 취미 탐색 정도로, 20% 정도가 학업과 관련된 정보 찾기로 활용하였습니다. 프로그래머 임요환이 컴맹이라는건 유명할 정도로 게임 많이 한다고 컴퓨터 잘하는건 아닌만큼 제가 게임을 하더라도 남들과 차별되는 점은 프로그램에 대한 이해력과 탐구심이었습니다. 소프트웨어의 세세한 설정을 위해 UI상의 옵션 외에 config를 탐색 및 수정하여 원하는 결과를 얻어내는 것은 물론 환경 구축에도 관심이 많아 지금도 그렇지만 친구들 사이에서 같이 플레이하는 게임을 위한 데디케이트드 서버 구축은 언제나 제 몫이었습니다.

한편 학업적인 측면에서의 컴퓨터 활용에서는 아래아 한글을 비롯한 워드프로세서는 당연하게 사용했고, 가장 흥미롭게 활용했던 프로그램은 스프레드시트였습니다. 엑셀의 여러가지 기능 중 특히 함수 기능에 매우 큰 흥미를 가졌으며 간단한 텍스트와 수치형 데이터를 엑셀 함수를 최대한 활용하여 어렸을 때보다 남들보다 훨씬 유연하고 빠르게 처리했습니다. 이는 마케팅 회사의 취업 및 업무 진행에 매우 큰 도움이 되었습니다.

본격적인 데이터와 AI에 큰 관심을 가지게 된 계기는 취업한 퍼포먼스 마케팅 회사에서의 경력 때문이라고 할 수 있겠습니다. 네이버, 구글과 같은 플랫폼에서의 검색광고를 운영하는 특성상 광고 성과에 대한 raw 단위의 데이터와 엑셀을 업무시간 내내 달고 살기 때문에 자연스럽게 빅데이터에 대한 관심으로 이어졌고 보다 전문적인 데이터의 처리와 분석을 하고 싶은 마음이 생기게 되었습니다. 당시에는 어디서부터 시작해야 할지 알 수 없어 Microsoft Access에 대해서 알아보는 정도였습니다.

AI 측면에서 이야기해보자면 1차적으로는 알파고가 큰 화제를 끌어 저 또한 최소한의 관심을 가지게 되었으나 AI에 더 큰 관심을 가지고 관련 업무를 직접 하고 싶었던 결정적인 계기는 Stable Diffusion의 등장이었습니다. 개인적인 생각으로 인류의 창의성과 독창성이 가장 중요한 예술 분야에서의 AI의 활용은 늦게 이루어질 것이라고 생각하고 있었으나 정작 이 업계가 AI로 인하여 가장 먼저 뒤집어지자 AI의 발전은 엄청난 속도로 이루어지고, 모든 산업에 종사하는 종사자들에게는 현실적인 위협이 될 수 있다는 사실에 큰 충격을 받아 본격적으로 데이터와 AI에 대한 공부를 진행했습니다.

1.3. 학력과 교육이력

- 재학기간 : 2010-03 ~ 2019-02

본디 건축공학과를 선택하여 대학교에 진학하고 학업을 진행하였으나 건축공학과와의 특성이 개인적으로 선호하는 방향과 맞지 않는 부분이 있어 결과적으로는 학과 선택에 있어서는 좋은 결과로 남지 않아서 아쉬운 부분입니다. 그렇더라도 공학계열인 만큼 공학도로서의 사고와 수치 계산 분야에 있어서는 배울 점은 상당히 많았다고 생각하고 있습니다.

메가스터디IT아카데미 교육 수료

- 과목명 : 빅데이터 분석기반 AI 알고리즘 개발과정
- 교육기간 : 2023-11 ~ 2024-05

비전공자가 AI 및 데이터 업계에 입문하기 위하여 많이 수강하는 국비강의 수업을 이수하였습니다. 해당 과정으로 Python 프로그래밍과 SQL문 사용, 머신러닝 및 딥러닝 과정을 익혔습니다. 다만 6개월의 국비과정을 수료하는것 만으로 전공자들을 포함한 뛰어난 인재들과 경쟁하는 것은 어렵습니다. 그렇기에 저는 별도의 독학이나 스터디를 통하여 DACON, Kaggle과 같은 AI 및 데이터 사이언스 경쟁 플랫폼에서의 추가적인 공부를 진행하고 취업에 필요한 자격증 취득도 병행하였습니다. 국비과정에서는 배우지 않은 LLM과 같은 생성형 AI 분야의 지식은 별도 학습을 통하여 얻게 되었습니다.

2. Skill (Language & Data)

2.1. Python

2.1.1. Python 기본

데이터 수집 및 정제에 요구되는 Python의 기본적인 문법을 활용할 수 있습니다.

- 학습기간 : 2023.01 ~
 - 학습기간 : 2023.01 ~
 - 습득방법 : 독학, 사설교육기관 등
 - 히스토리
 - [Github1](#)
 - [Github2](#)
- 상세 보유기술 요약
 - Python 기초 문법 (연산자, 제어문)

- 리스트, 튜플, 딕셔너리 등의 컬렉션 타입의 이해
- 함수의 사용과 클래스 객체의 활용

2.1.2. Python 패키지 사용

데이터와 통계적인 수단을 편하고 빠르게 사용할 수 있는 패키지들의 종류와 각 패키지 별로 어떠한 기능이 있는지, 또한 필요한 과정에 따라 어떤 객체나 메서드를 사용해야 하는지 파악하고 있습니다.

- 기본 학습 정보
 - 학습기간 : 2023.01 ~
 - 습득방법 : 독학, 사설교육기관 등
 - 히스토리
 - [Github1](#)
 - [Github2](#)
 - [Github3](#)
- 상세 보유기술 요약
 - 데이터 수집
 - OpenAPI 혹은 웹사이트에 대한 정적 크롤링 (Request, BeautifulSoup)
 - Javascript 기반 웹사이트의 데이터를 수집하는 동적 크롤링(Selenium)
 - 데이터 가공 및 정제, 전처리 작업 수행
 - 수치, 배열 타입 데이터의 분석 및 가공 (Numpy)
 - 시리즈, 데이터프레임 타입 데이터의 정제와 전처리 작업 수행 (Pandas, Polars, scikit-learn, Imblearn)
 - 데이터 시각화
 - Matplotlib, Seaborn을 이용한 데이터 시각화
 - 통계분석
 - 가설검정 및 분산분석 (scipy)
 - 회귀분석, 로지스틱 회귀, 시계열 분석(statsmodels)

2.1.3. 머신러닝

scikit-learn 등의 패키지를 이용한 회귀, 분류분석을 수행하는 일련의 과정을 이해하고 모델의 생성, 예측, 성능평가를 진행할 수 있습니다.

scikit-learn, xgboost, LGBM 등 이미 만들어진 패키지를 활용하는 선을 넘어 머신러닝과 관련된 논문에서 아이디어를 얻어 자체적인 예측 모델을 만들어보는 등의 시도를 하기도 했습니다.

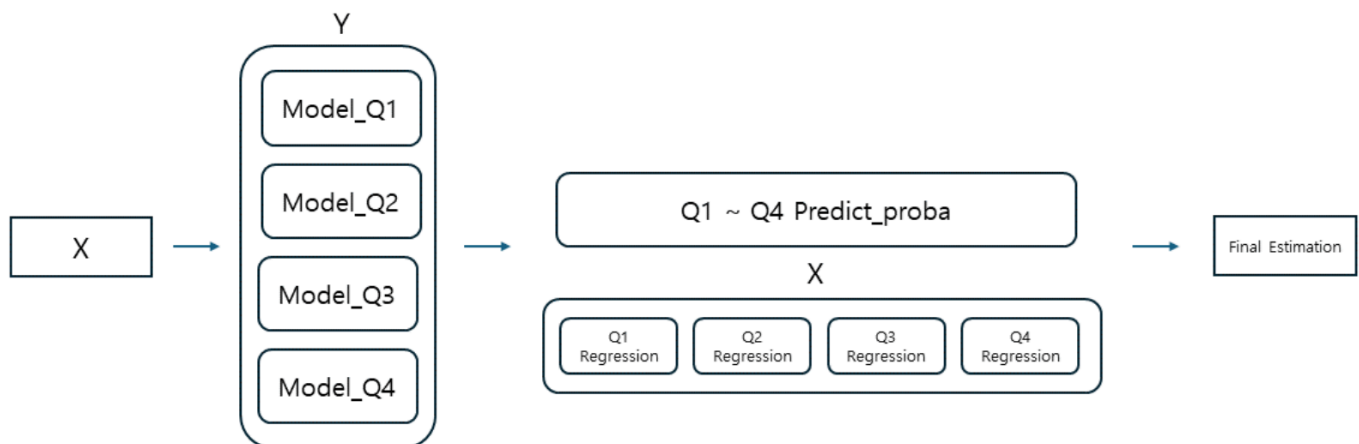
ex) 회귀분석에서의 종속변수의 사분위수를 기준으로 모델을 분할하여 예측하는 모델
(코드 일부)

```
class YjkQuantileRegressor():
    def fit(self, model_reg, model_cls, data, yname):
        if type(yname) != str:
            data = pd.concat([data, yname], axis = 1).copy()
            try:
                yname = yname.name
            except:
                yname = yname.columns[0]

        Q1 = data[yname].quantile(.25)
        Q2 = data[yname].median()
        Q3 = data[yname].quantile(.75)

        data['quantile_for_regression'] = data[yname].apply(lambda x : 1 if x < Q1
else(2 if x < Q2 else(3 if x < Q3 else 4)))
        self.model_cls = dc(model_cls)
        X_cls = data.drop([yname, 'quantile_for_regression'], axis = 1)
        Y_cls = data['quantile_for_regression']
        self.model_cls.fit(X_cls, Y_cls)

    ...
```



- 기본 학습 정보

- 학습기간 : 2023.04 ~
- 습득방법 : 사설교육기관, Kaggle / DACON 경진대회 참여
- 히스토리
 - Repository
 - 머신러닝1
 - 머신러닝2
 - 대회참여1
 - 참여한 경진대회
 - Kaggle1

- [Kaggle2](#)
- [Kaggle3](#)
- [Kaggle4](#)
- [DACON1](#)
- 상세 보유기술 요약
 - 학습 데이터의 기본 정보 확인
 - 데이터의 Shape를 조회하여 데이터의 크기 파악
 - Feature별 데이터 타입 및 기초통계량 조회
 - 종속변수 분포 확인 및 머신러닝 목표 설정
 - 데이터 전처리
 - 이상치 및 결측치 파악 및 정제
 - 스케일링 (Standard / MinMax / Robust / Log Scaling) 적용
 - 인코딩 (Onehot-Encoding, Label-Encoding) 적용
 - 변수 가공, 파생변수 생성 등 Feature Engineering 진행 (binning, PCA 등)
 - 모델 학습
 - 훈련 데이터셋 준비
 - 데이터의 특성에 따른 훈련 / 검증용 데이터 분리 (무작위 분리, 시계열 기반 timeseries split 등)
 - 학습 모델 선정
 - 패키지로 제공되는 모델 종류 파악 (scikit-learn 제공 모델, xgboost, lightgbm, catboost)
 - 앙상블 기법들의 종류 및 특성 이해 (voting, bagging, boosting, stacking)
 - 모델 훈련 및 성능 평가
 - 훈련 데이터셋을 사용한 모델 훈련
 - 모델 성능 평가를 위한 평가지표 파악(RMSE, MAE, accuracy, ROC_AUC 등)
 - 모델 성능 개선을 위한 하이퍼파라미터 튜닝

2.1.4. 딥러닝

인공신경망을 구성하여 예측을 수행하는 딥러닝 패키지를 활용할 수 있습니다. 분류 및 회귀를 위한 딥러닝 학습 시에는 주로 tensorflow를 활용하고 있으나 LLM에서는 pytorch의 활용 비중이 높아 해당 패키지의 사용법도 숙지하고 있습니다.

- 기본 학습 정보
 - 학습기간 : 2023.04 ~

- 습득방법 : 사설교육기관, Kaggle / DACON 경진대회 참여
- 히스토리
 - [Github1](#)
 - [Github2](#)
- 상세 보유기술 요약
 - 딥러닝 기본 사항
 - 인공신경망의 구조와 특성, 머신러닝과의 차이점 이해
 - 딥러닝 관련 패키지 사용 (tensorflow, pytorch, keras)
 - 인공신경망 구성
 - Layer 구조의 이해 및 Sequential 모델 생성
 - 다중 은닉층 구성 (Dense, CNN, RNN 및 파생 Layer)
 - 활성화 함수의 종류와 각 특성의 이해
 - 모델 성능 개선을 위한 Dropout, BatchNormalization, Polling 층 활용
 - 인공신경망 훈련
 - 옵티마이저, 손실함수, 평가지표의 종류의 이해
 - 훈련 데이터로 인공신경망 훈련 및 성능평가
 - 모델 튜닝으로 성능 개선

2.1.5 LLM

현재 AI 시장의 트렌드는 단연 생성형 AI라고 할 수 있습니다. LLM은 텍스트 데이터를 토큰화 과정을 통해 가공 후 학습하여 분류, 텍스트 생성 등의 역할을 수행할 수 있는 대형 언어 모델로, Stable Diffusion으로 대표되는 이미지 생성 모델과 함께 각 기업들이 큰 관심을 가지고 연구하는 분야입니다.

저 또한 이러한 트렌드에 따라가기 위하여 LLM의 사용법을 이해하고자 관련 내용과 지식을 습득, 직접 모델을 사용하여 경험을 축적하고 있습니다.

개인이 LLM을 사용함에 있어 가장 큰 걸림돌은 문자 그대로 큰 모델로 인한 매우 많은 파라미터로 모델이 매우 무거운 편이라는 점입니다. 개인용으로 사용할 수 있는 GPU 중에서 VRAM이 가장 큰 제품이 24GB로, 이를 사용하더라도 GPU 병렬구성을 하지 않는 이상 파라미터 7B 이상의 모델들은 OOM(Out Of Memory) 에러로 인하여 순정으로 사용하는것이 사실상 불가능합니다. (skt/ko-gpt-trinity-1.2B-v0.5 사용시 VRAM 22GB 점유 확인)이를 해소하기 위한 여러가지 방법이 있으나 저의 경우에는 LoRA(Low Rank Adaption)을 이용하여 학습시킬 파라미터의 개수를 축소하여 모델을 경량화하는 방식을 택했습니다. 해당 방법으로 7B Mistral, 10.7B Solar 모델까지는 큰 이슈 없이 모델 훈련을 완료한 경험이 있습니다.

ex) google-bert/bert-base-uncased 모델의 LoRA 미적용 / 적용시 VRAM 점유율 비교,
22GB -> 4GB로 VRAM 점유가 1/5 이하로 감소

```
# Without LoRA
# +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
# | NVIDIA-SMI 550.54.15                               Driver Version: 550.54.15           CUDA Version: 12.4           |
# |-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
# | GPU   Name                               Persistence-M | Bus-Id                Disp.A | Volatile Uncorr. ECC |
# | Fan   Temp   Perf                         Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
# |                               |                               |                               | MIG M. |
# |=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+
# |    0  NVIDIA GeForce RTX 3090              Off | 00000000:08:00.0 Off |                  N/A |
# | 75%    80C    P2                          415W / 420W | 21946MiB / 24576MiB |      98%    Default |
# |                               |                               |                               | N/A |
# +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

# +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
# | Processes:                                         |
# | GPU   GI   CI           PID   Type   Process name                               GPU Memory |
# |      ID   ID                                   |              Usage |
# |=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+
# |    0   N/A   N/A       10575    C     /bin/python3                               21940MiB |
# +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
# With LoRA
# +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
# | NVIDIA-SMI 550.54.15                               Driver Version: 550.54.15           CUDA Version: 12.4           |
# |-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
# | GPU   Name                                           Persistence-M | Bus-Id           Disp.A | Volatile Uncorr. ECC |
# | Fan   Temp   Perf           Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
# |                                           |                       |              MIG M. |
# |=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+
# |    0  NVIDIA GeForce RTX 3090                Off | 00000000:08:00.0 Off |                  N/A |
# | 80%    80C    P2              410W /  420W |  3918MiB / 24576MiB |      90%      Default |
# |                                           | 3918MiB           |                  N/A |
# |-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

# +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
# | Processes:                                           |
# | GPU   GI    CI          PID    Type    Process name                      GPU Memory |
# |      ID    ID                                   |            Usage |
# |=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+
# |    0   N/A   N/A         10430     C   /bin/python3                      3912MiB |
# |-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
# +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

- 기본 학습 정보
 - 학습기간 : 2023.09 ~
 - 습득방법 : Kaggle / DACON 경진대회 참여
 - 히스토리
 - Repository
 - [Github1](#)
 - [Github2](#)
 - 참여한 경진대회

- [DACON1](#)

- [Kaggle1](#)

- 상세 보유기술 요약

- LLM의 개념 및 역할 이해
- HuggingFace를 통한 모델 및 토큰나이저 다운로드
 - 사용 모델 : LLaMA, Google Gemma, BERT, Solar, Mistral, GPT-2 등
- 텍스트 데이터의 전처리 수행 (형태소 분석, 불용어 처리 등) 및 토큰화
- 훈련 데이터셋에 대한 LLM Fine-Tuning 및 예측 결과 / 문장 생성
- LoRA를 활용한 모델 및 학습 경량화

2.1.6 etc

Python 사용시 반복적인 작업을 빠르게 수행하기 위해서 패키지를 직접 만들어서 활용한 경험이 있습니다.

ex) 데이터 전처리 : 이상치, 결측치, 스케일링, 인코딩을 한번에 수행하는 패키지 구성

```
1 pp = YjkPreprocessor()
2 pp.set_strategy(outline = 'q', null = 'median', scaler = 's', encoder = 'o')
```

[4] ✓ 0.1s

... 아래와 같이 처리합니다.

범주	처리방법
이상치 처리	경계값 대체
결측치 처리	중앙값 대체
스케일러	StandardScaler
인코딩	원핫인코딩

```
1 df2 = pp.fit_transform(df)
```

[6] ✓ 0.1s

... 데이터프레임 크기 : (506, 14)
데이터프레임에 결측치는 없습니다.
연속형 데이터 컬럼 : ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT', 'MEDV'], 총 14개

- [Github1](#)

2.2 SQL

SQL문을 이용한 DB의 조회, 데이터 관리, 수정, 삭제를 수행할 수 있습니다.

개인적으로 주로 사용하는 DBMS 관리 도구는 DBeaver입니다.

- 기본 학습 정보

- 학습기간 : 2023.03 ~
- 습득방법 : 독학, 사설교육기관

- 상세 보유기술 요약
 - 사용 DBMS
 - Oracle, MySQL(MariaDB)
 - DBMS 활용
 - 테이블의 조회, 수정, 삭제
 - Python과 연동한 DBMS 데이터 관리 (cx_Oracle - Oracle)

2.3. etc

2.3.1. Streamlit 기반 웹 API 구현

Python을 이용한 웹페이지 구성은 Django, FastAPI 등이 주로 사용되지만 저는 입문용으로 간편하게 사용할 수 있는 Streamlit 패키지를 이용하여 간단한 머신러닝 / 딥러닝 프로세스를 구현한 적이 있습니다. 해당 패키지는 대쉬보드 제작을 목표로 구현된 프로젝트이며, 데이터에 대한 시각화, 데이터 직접 입력 등을 지원합니다.

ex) 구현한 웹 상에서 데이터 입력 및 조회

YJK's ML Platform

데이터 불러오기

데이터 불러오는 방법 선택

☒ 파일 직접 업로드

csv와 xlsx 파일 지원

☐ URL에서 다운로드

URL 링크는 csv 혹은 xlsx 파일을 직접 지정해야 함

파일 업로드



Drag and drop file here

Limit 200MB per file

Browse files



pima_indians_diabetes.xlsx 47.2KB



업로드 데이터 확인

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Ag
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	
5	5	116	74	0	0	25.6	0.201	
6	3	78	50	32	88	31	0.248	
7	10	115	0	0	0	35.3	0.134	
8	2	197	70	45	543	30.5	0.158	
9	8	125	96	0	0	0	0.232	

ex) 업로드한 데이터에 대한 전처리 수행

연속형 변수 처리

연속형 변수에 대한 스케일링 방법을 정의하고 전처리를 수행합니다.

스케일링 방법 정의

- ☐ 처리하지 않음
- ☒ StandardScaler
- ☐ MinMaxScaler
- ☐ RobustScaler
- ☐ LogScale

상용로그를 적용하여 변환합니다. 데이터 컬럼에 0 이하의 실수가 없는지 확인하십시오.

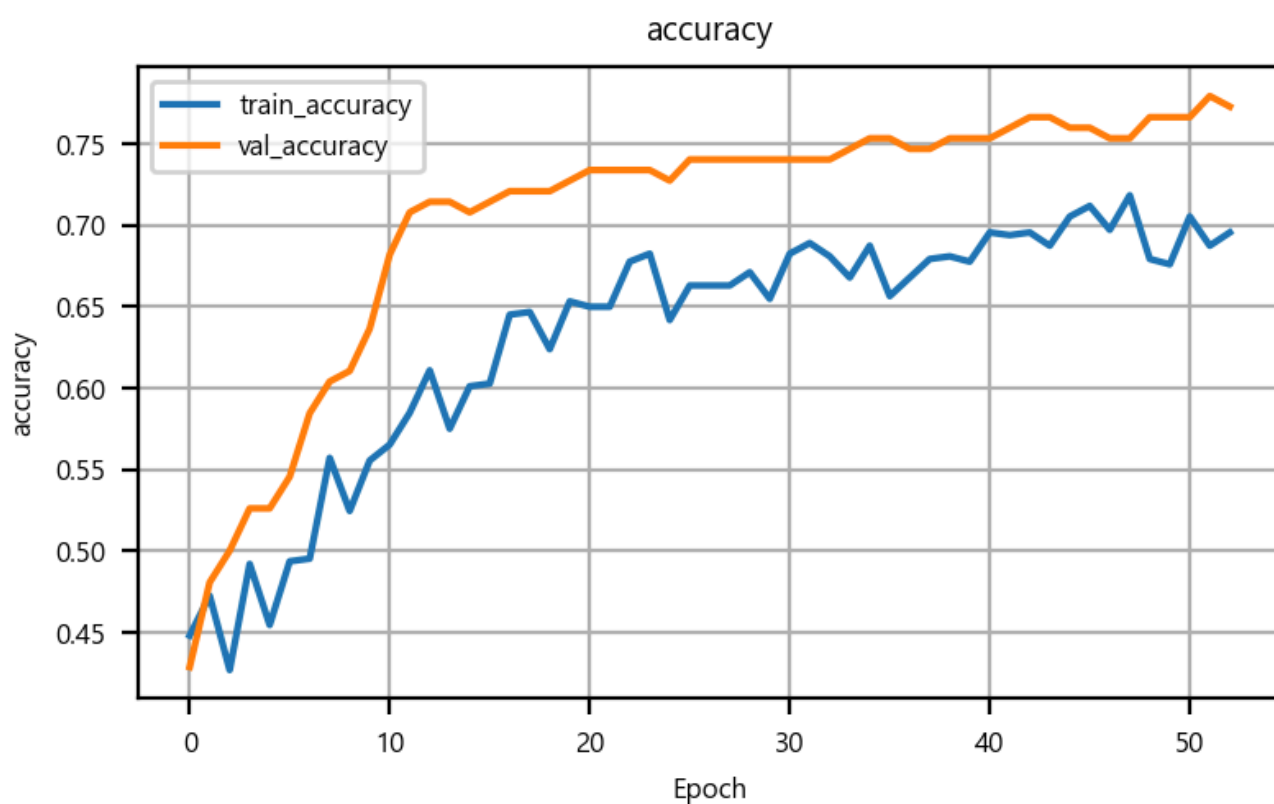
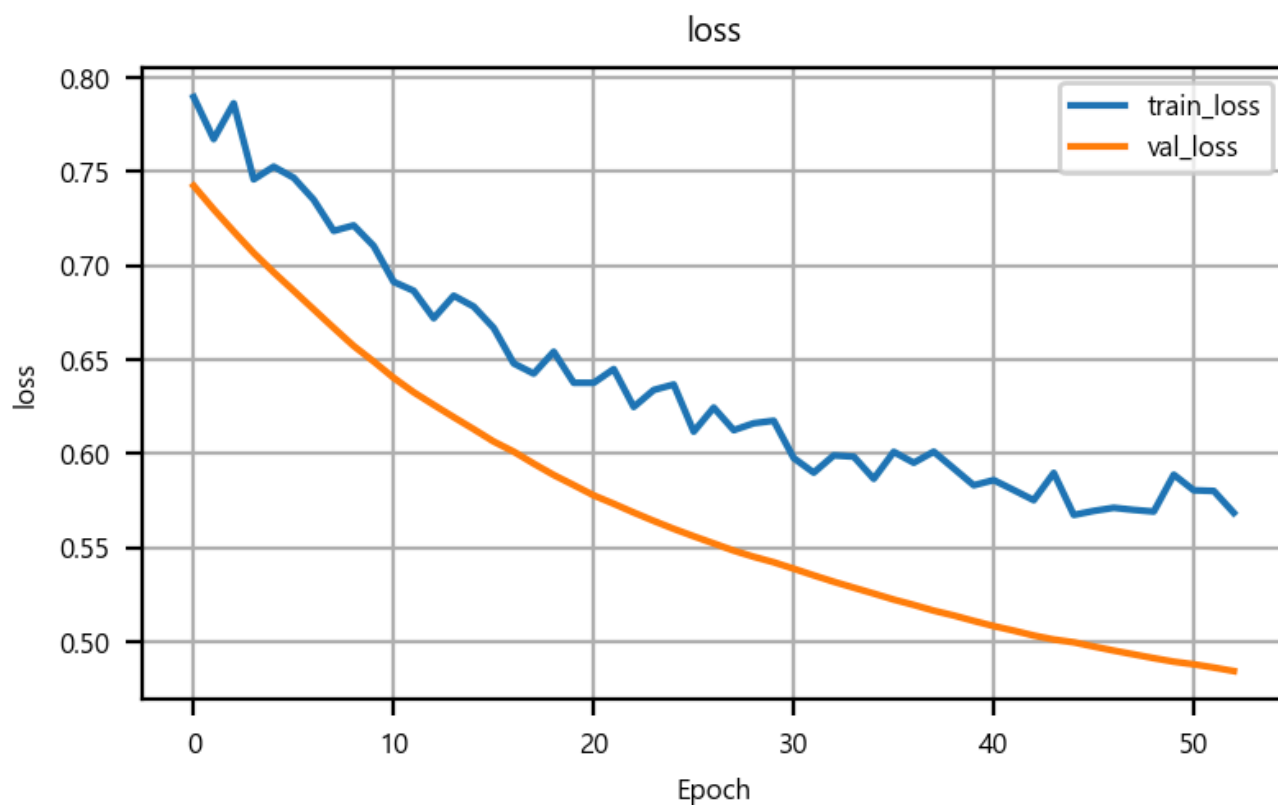
스케일링 적용 결과

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.6399	0.8483	0.1496	0.9073	-0.6929	0.204	0.4685	1.4
1	-0.8449	-1.1234	-0.1605	0.5309	-0.6929	-0.6844	-0.3651	-0.1
2	1.2339	1.9437	-0.2639	-1.2882	-0.6929	-1.1033	0.6044	-0.1
3	-0.8449	-0.9982	-0.1605	0.1545	0.1233	-0.494	-0.9208	-1.0
4	-1.1419	0.5041	-1.5047	0.9073	0.7658	1.4097	5.4849	-0.0
5	0.343	-0.1532	0.253	-1.2882	-0.6929	-0.8113	-0.8181	-0.2
6	-0.251	-1.3425	-0.9877	0.7191	0.0712	-0.126	-0.6761	-0.6
7	1.8278	-0.1845	-3.5726	-1.2882	-0.6929	0.4198	-1.0204	-0.3
8	-0.5479	2.3819	0.0462	1.5346	4.0219	-0.1894	-0.9479	1.6
9	1.2339	0.1285	1.3904	-1.2882	-0.6929	-4.0605	-0.7245	1.7

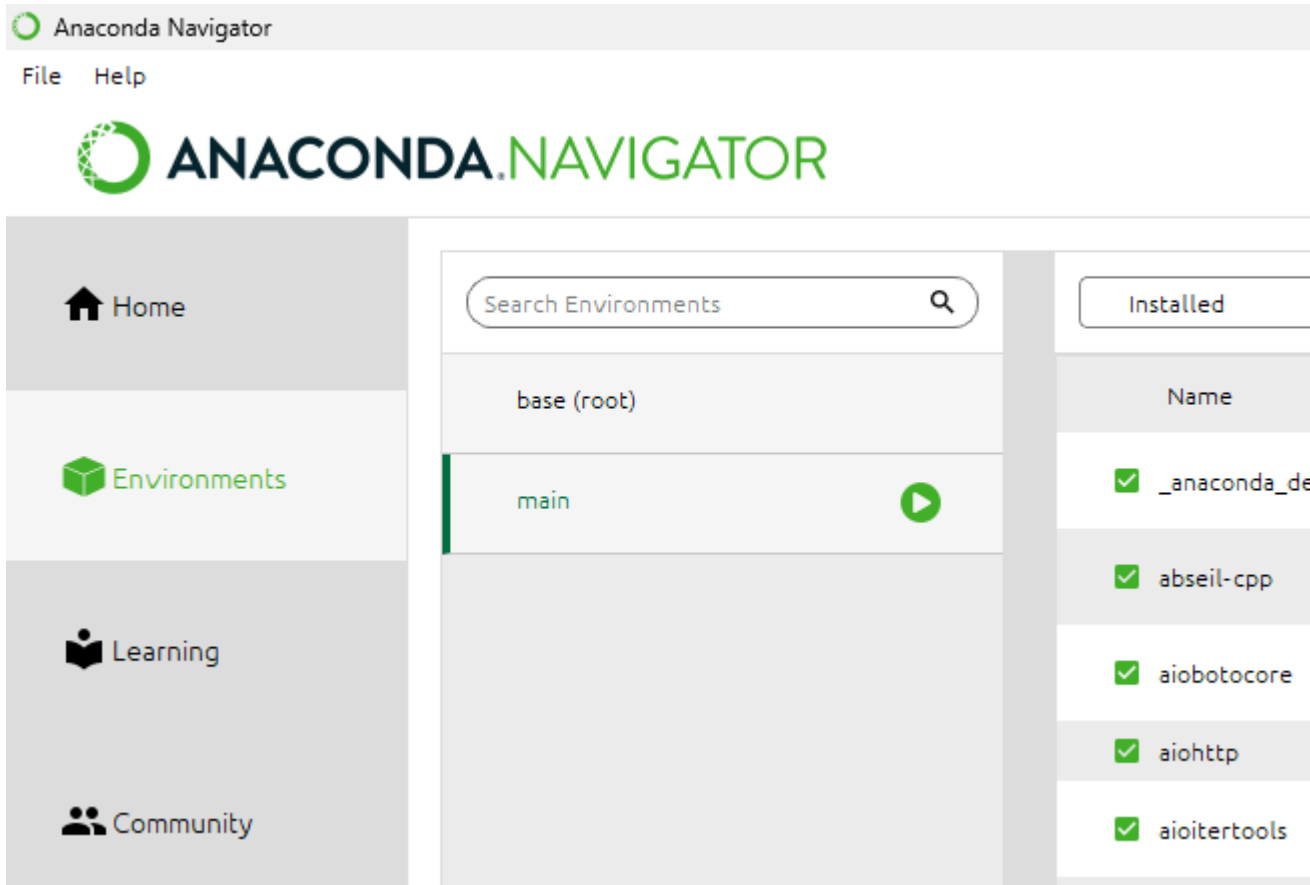
ex) 딥러닝 학습 수행 및 결과 확인

훈련 결과 확인

학습 곡선



Anaconda3에 기반한 Python 데이터 분석 환경 구축을 수행할 수 있습니다. Anaconda3를 이용하면 기본 개발환경 구성에 더하여 필요시 추가적인 env를 구성하여 별도 파이썬 / R 환경을 구축하여 별도 운영이 가능합니다.



Windows 환경의 경우 Linux에 비하여 데이터 분석을 위한 패키지 지원이 미비하여 일부 패키지 사용시에는 버전 이슈를 겪거나 특정 기능을 지원하지 않는 경우가 많습니다. 예를 들어, Nvidia 사의 CUDA 장치를 딥러닝 학습 환경에서 이용하려고 할 경우 Tensorflow 패키지는 Windows 환경에서는 2.10 버전을 마지막으로 GPU 가속 지원이 중단되어 최신 버전의 Tensorflow에서는 GPU 가속환경을 사용할 수 없습니다. 딥러닝 공부를 수행하면서 많은 시행착오를 겪으며 찾은 개인적인 최적의 Python 환경 조합은 아래와 같습니다.

- Python 3.10 (3.11부터 Tensorflow 2.10.0 미지원)
- CUDA v11.8
- cuDNN 8.9.0
- Tensorflow 2.10.0
- Pytorch (CUDA v11.8 ver.)

위 환경대로 사용하더라도 일부 패키지는 버전 이슈가 여전히 발생하고 있어(ex. LoRA에 사용되는 bitsandbytes 패키지는 최신 버전에서 이슈 발생) Linux 환경 구축에 관심을 가지게 된 계기가 되었습니다.

3.2. Linux

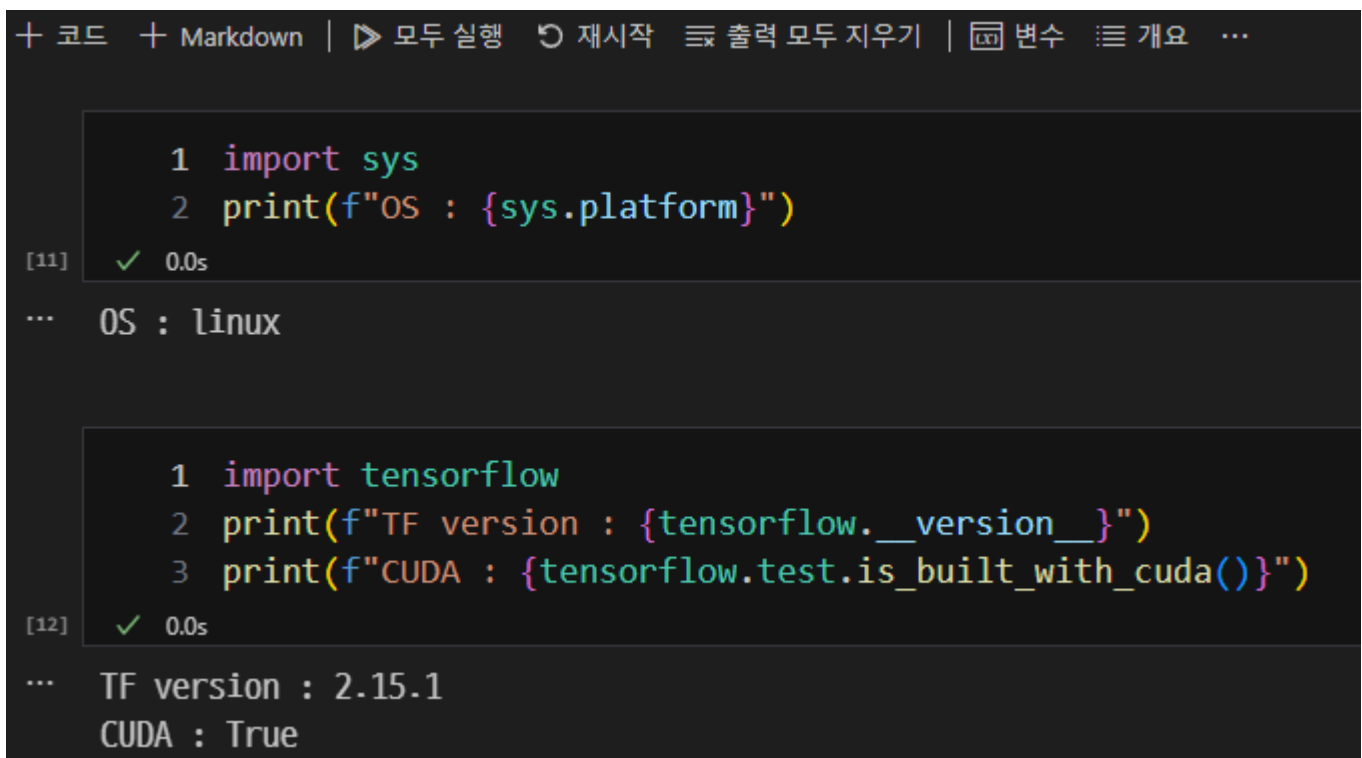
Linux는 Windows보다 데이터 분석 및 머신러닝 / 딥러닝 작업을 더욱 효율적으로 활용할 수 있는 환경입니다. 다만 일반인의 접근은 Windows보다 친화적이지 않아 저 또한

데이터 분석 직군에 관심을 가지고서야 직접적으로 접하였습니다. 다만 데이터 분석 업무를 한다면 Linux 환경은 필연적으로 접할 수 밖에 없기 때문에 Linux를 PC에 직접 설치하여 여러가지 기능을 직접 체험해보고 환경에 익숙해지고자 노력하고 있습니다.

- Linux 환경 구축 과정에서 기록한 업무별 노트 히스토리
 - [Github1](#)
- Linux의 기능과 이해
 - Linux의 기본 명령어 사용
 - sudoer 권한에 대한 이해와 계정별 권한 부여 설정
 - vim, nano 에디터의 사용
 - cron 서비스를 통한 업무 자동화
 - tmux를 통한 다수 터미널 동시 관리

3.2.1. Python 환경 구축

리눅스 배포판 중 우분투 설치 및 Python 환경 구축이 가능합니다. Windows 항목에서 상술한 바와 같이 데이터 분석 / 머신러닝 관련 패키지의 지원은 Windows보다는 Linux 환경에서 더욱 적극적으로 이루어지고 있기 때문에 Linux 환경을 직접 구축하였습니다. 사용한 우분투 버전은 22.04 서버 버전으로, CLI 환경으로 불필요한 백그라운드 프로세스가 실행되지 않아 컴퓨터 자원을 최대한으로 사용할 수 있습니다.



```
+ 코드 + Markdown | ▶ 모두 실행 ↺ 재시작 ≡ 출력 모두 지우기 | 📄 변수 ≡ 개요 ...

1 import sys
2 print(f"OS : {sys.platform}")

[11] ✓ 0.0s
... OS : linux

1 import tensorflow
2 print(f"TF version : {tensorflow.__version__}")
3 print(f"CUDA : {tensorflow.test.is_built_with_cuda()}")

[12] ✓ 0.0s
... TF version : 2.15.1
    CUDA : True
```

우분투 리눅스에서는 Windows와 다르게 패키지별로 최신 버전을 사용하더라도 GPU 등의 장치를 최상의 퍼포먼스로 활용할 수 있었습니다. (LLM 파인 튜닝 기준 Windows 대비 10% 빠르게 이루어짐)

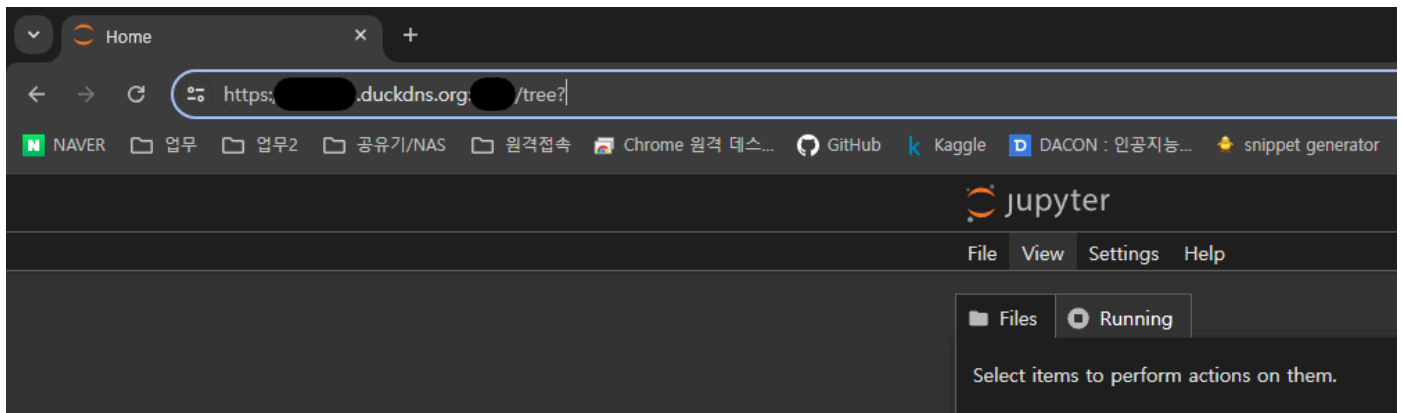
- Python 3.11
- CUDA v12.4
- cuDNN 9.1.0
- Tensorflow 2.15.1
- Pytorch(CUDA v12.1 ver.)

3.2.2. Jupyter Server와 SSH 원격 접속 환경 구축

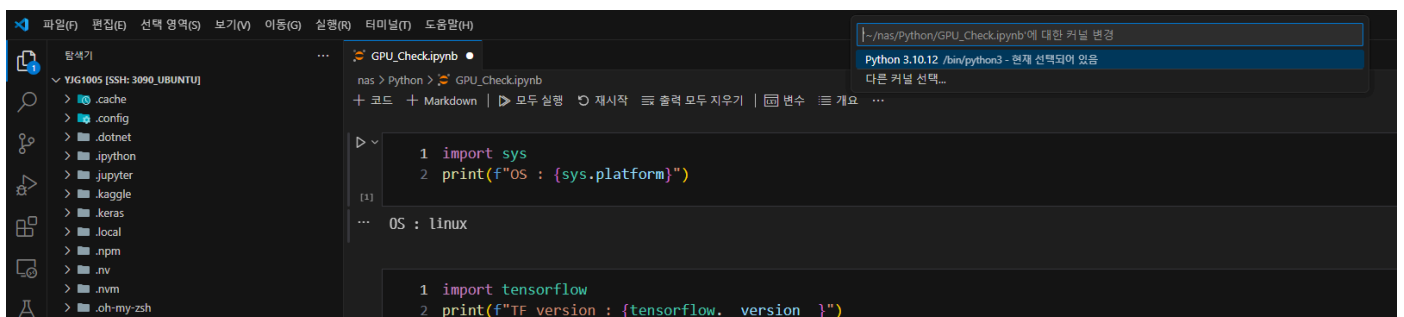
Linux를 설치한 PC는 CLI 환경이기 때문에 해당 PC에서 직접 Python을 사용하는데는 여러가지 제약 사항이 있기 때문에 별도 클라이언트 PC를 통하여 작업하는 것이 바람직합니다. 이를 위한 원격 접속 환경을 구축하여 언제 어디서든 인터넷이 연결된 PC만 있다면 구축한 Linux 서버 PC에 원격으로 접속하여 데이터 / 머신러닝 작업이 가능하도록 세팅하였습니다.

별도 어플리케이션 설치 없이 Web 기반 Jupyter Notebook 서버를 구축하거나 VSCode 상에서 SSH로 Linux Python 환경에 직접 접속하는 환경을 모두 세팅하여 필요에 따라 적합한 방법으로 활용하고 있습니다.

ex1) Jupyter Notebook WebUI 서버 구축



ex2) VSCode의 SSH 원격 접속을 통하여 Linux 환경에 접속



3.2.3. Docker

Docker는 Linux의 대표적인 컨테이너 기술 및 서비스로 여러가지 서비스를 편리하고 안전하게 운영할 수 있습니다. 딥러닝 환경을 예시로 하였을 때 직접 딥러닝 서비스를 구축하는 일련의 과정 (Python 패키지 설치, CUDA 환경 구축, Jupyter 서버 활성화)을 미리 생성된 이미지를 로드하는 것 만으로 빠르게 수행할 수 있습니다. 저는 해당 서비스를 기반으로 GPU와 연동되는 Tensorflow - Jupyter 서버를 한번에 구축 가능한 이미지를 구동하여 활용한 경험이 있습니다. 다만 아직까지는 직접 구축한 환경이 더욱 친숙하고 관리가 편리한 면이 있어 Docker 환경 활용에 더욱 능숙해지고나서 활용도를 차츰 늘려나갈 계획에 있습니다.

3.3. etc

3.3.1 편의성을 고려한 개발환경 구축

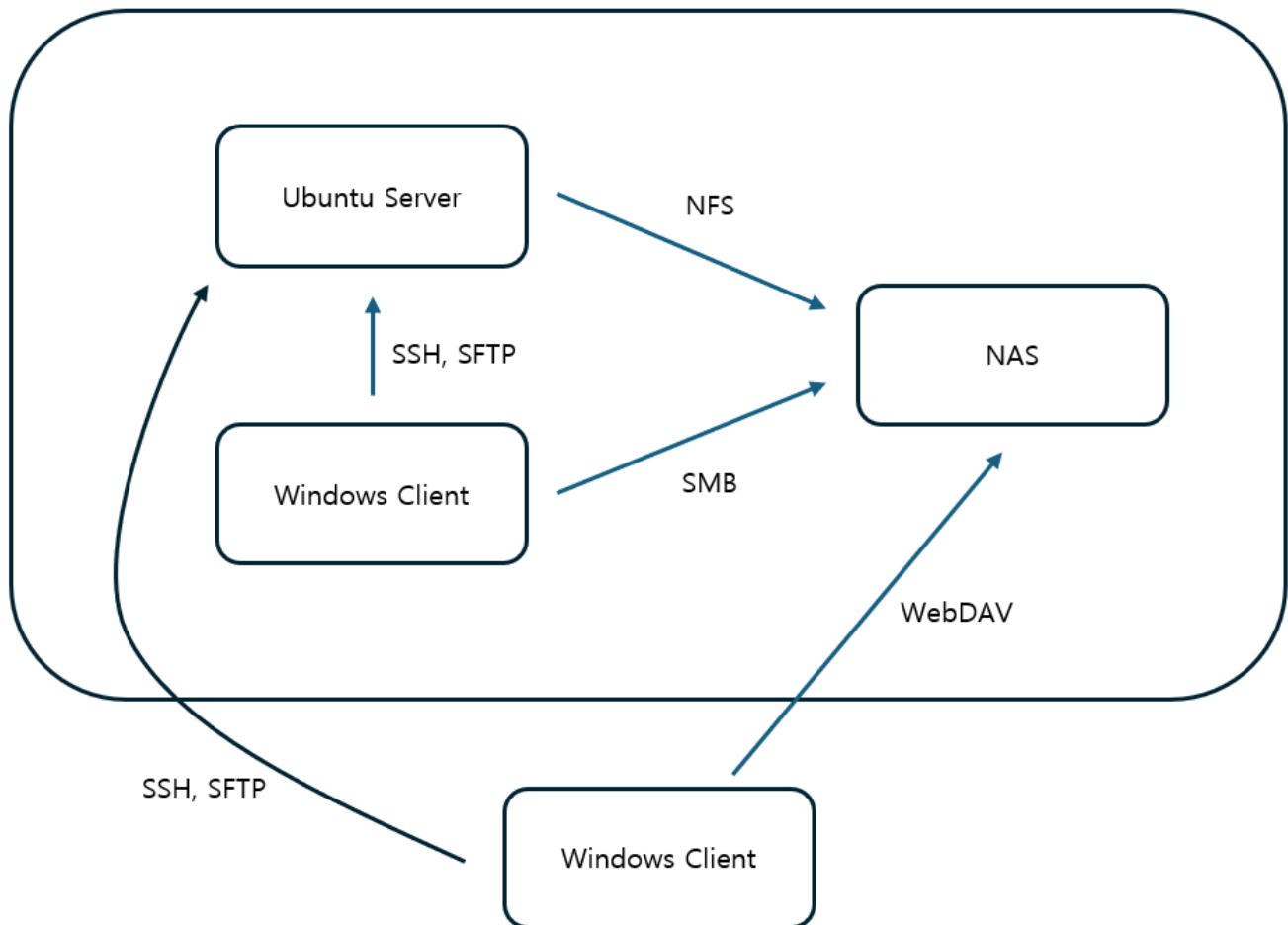
상기한 바와 같이 Windows와 Linux 환경을 모두 운영하고 있고 (Windows PC와 Linux PC는 별개) 각기 환경을 필요에 따라 모두 활용하고 있기에 일반적으로는 작업 환경 동기화가 되지 않는 단점이 있습니다. 저는 이러한 단점을 해소하고자 저의 개발환경에 NAS(Network Attached Storage)를 추가하여 활용중에 있습니다.

데이터 분석 / 머신러닝을 위한 데이터 원본, 코드를 저장하는 Python Root 폴더를 모두 NAS로 설정하면 사용하는 PC에 구애받지 않고 인터넷에 접속만 가능하다면 빠르고 편하게 활용할 수 있는 사실상의 개인 클라우드 서비스를 구축하였습니다. 로컬 네트워크 상에서의 Windows 및 리눅스, 외부 WAN을 통하여 접속하는 Windows 컴퓨터는 모두 다른 방식으로 NAS를 운영체제에 마운트하여 저장소에 접근할 수 있는 방법에 대해 이해하고 있습니다.

가정 환경 특성상 WAN IP는 변할 가능성이 있으므로 외부 접속시 이에 영향을 받지 않도록 DDNS 도메인을 등록하여 접속에 활용중이며, 보안을 위한 비밀번호 설정은 물론 SSH 프로토콜 접속시에는 비밀번호보다는 인증키를 활용하고 있으며 Jupyter 웹 서버에도 인증서를 등록하여 https 접속을 활성화하였습니다. 포트 또한 LAN 내부에서는 디폴트로 설정된 포트를 쓰고 있지만 외부 접속시에는 포트포워딩을 설정하여 별도의 포트로 SSH나 NAS 접속을 하도록 설정하였습니다.

ex) 개인 데이터 분석 개발환경 도식화

Router (LAN)



4. Project

4.1. 데이터 수집 프로젝트

데이터 수집 능력을 평가하는 프로젝트 결과물입니다. 데이터 수집은 네이버 쇼핑몰의 리뷰 데이터를 수집하여 DB에 입력하는 것을 최종 목표로 하였습니다.

일련의 과정은 클래스 객체를 정의하여 실제 데이터 수집시 입력하는 코드를 최소화하는 것으로 목표로 프로젝트를 진행하고 결과물을 산출하였습니다. DBMS 연동 시 데이터형과 관련한 이슈가 발생하였으나 최종적으로는 해당 문제를 해결하여 만족스러운 결과를 산출한 프로젝트입니다.

ex) 데이터 수집 클래스 구성 설명

크롤링 수행 클래스 작성 (naver_review)

메서드	입력데이터	내용	출력데이터
생성자 (__init__)	없음	크롬드라이버 버전 체크 및 실행 후 빈 페이지 상태로 대기	없음
connect_url	텍스트 형식의 url	입력받은 url 접속 후 리뷰 페이지로 이동	없음
rip_review	크롬드라이버 page_source 객체	현재 조회중인 리뷰 페이지에서 최대 20개의 리뷰 데이터 수집	단일 페이지의 리뷰 정보 를 담은 리스트
rip_all	없음	현재 조회중인 페이지부터 시작하여 페이지 끝까지 연속하여 데이터를 수집 단일 페이지의 리뷰는 rip_review 메서드를 호출하여 리뷰를 수집	다수 페이지의 리뷰 정보 를 담은 리스트
page_summary	없음	현재 조회중인 페이지의 리뷰 정보와 상품명, 상품 가격을 수집 리뷰 정보는 rip_all 메서드를 호출하여 리뷰를 수집	페이지의 상품 정보와 리뷰를 담은 딕셔너리
rip_list	텍스트형식의 url을 담은 리스트	입력받은 리스트 내 모든 url을 탐색하여 상품 정보 및 리뷰를 모두 수집 단일 url에 대한 정보는 page_summary 메서드를 호출하여 수집	여러 모든 정보를 담은 리스트
close	없음	크롬 드라이버 접속 종료	없음

ex) 프로젝트 DB 입력 예시

데이터베이스 최종 입력 결과 확인

PRODUCT_LIST 테이블

PRODUCT_ID	PRODUCT	PRICE	URL
1	갤럭시 버즈2프로 SM-R510	137480	https://smartstore.naver.com/o-ma/products/7363123499
2	Apple 2023 에어팟 프로 2세대 USB-C 충전 케이스 모델 (MTJV3KH/A)	359000	https://brand.naver.com/applestore/products/9360093290
3	삼성전자 갤럭시 버즈2 SM-R177	99900	https://smartstore.naver.com/uniyuni/products/607155696
4	갤럭시 버즈2프로 SM-R510	137480	https://smartstore.naver.com/o-ma/products/7363123499
5	소니 WF-1000XM5	359000	https://brand.naver.com/sonystore/products/8932776097
6	티3 블루투스 이어폰 무선 노이즈 캔슬링 국내AS QCY	42000	https://smartstore.naver.com/cotini/products/5357757813

REVIEW_LIST 테이블

ID	PRODUCT_ID	PRODUCT_OPTION	REG_DATE	SCORE	POSITIVE_NEGATIVE	REVIEW
821	821	버즈2프로 (색상) : 화이트	22/11/19	5	긍정	노캔 잘 씁니다. 좋은 제품 빠른배송 감사합니다
822	822	버즈2프로 (색상) : 그라파이트	22/11/17	4	긍정	대체적으로 만족함 화이트로 실감 그랬나 후회중
823	823	버즈2프로 (색상) : 화이트	22/11/22	5	긍정	말아미 생일선물로 구매했는데 무척 좋아하네
824	824	버즈2프로 (색상) : 화이트	22/11/20	5	긍정	잘받았습니다 사용은 안해봤지만 삼성이니까...
825	825	버즈2프로 (색상) : 라벤더	22/11/26	5	긍정	너무 좋아요 ***
826	826	버즈2프로 (색상) : 그라파이트	22/12/06	5	긍정	가격도 좋고 상품도 좋고 만족스럽습니다!!
827	827	버즈2프로 (색상) : 그라파이트	22/11/28	5	긍정	너무 너무 감사 드립니다.
828	828	버즈2프로 (색상) : 화이트	22/11/28	5	긍정	너무 너무 감사 드립니다.
829	829	모델 선택: 에어팟 프로 2세대 MTJV... 23/12/16	23/12/16	4	긍정	고양이를 선물로 사줬는데 너무 잘 사용하고 있네요. 소음차단에 잘 돼서 좋다고 하네요. 2
830	830	모델 선택: 에어팟 프로 2세대 MTJV... 23/12/14	23/12/14	5	긍정	프로1쓰다가 바꿨습니다 음질이 뭔가 좋아진거 같습니다!!! 너무 영통해요 배송도 바로오고
831	831	모델 선택: 에어팟 프로 2세대 MTJV... 23/12/15	23/12/15	5	긍정	프로1세대보다 모르게 좋아졌네요 노이즈캔슬링도 그렇고 매우 만족합니다.
832	832	모델 선택: 에어팟 프로 2세대 MTJV... 23/12/07	23/12/07	5	긍정	와이프가 기존에 쓰던 에어팟이 망가져서 올해 마지막 기념으로 선물해줬는데 아주 만족스럽
833	833	모델 선택: 에어팟 프로 2세대 MTJV... 23/12/12	23/12/12	5	긍정	언니 생일선물로 장만해줬습니다*** 엄청 좋아하네요 배송 빨랐고 포장상태도 좋았습니다!
834	834	모델 선택: 에어팟 프로 2세대 MTJV... 23/12/14	23/12/14	5	긍정	에어팟 프로 2세대 이슈없이 잘 사용하고 있습니다. 제조일도 한달 밖에 안되어서 좋았어요
835	835	모델 선택: 에어팟 프로 2세대 MTJV... 23/12/05	23/12/05	5	긍정	에어팟프로 3년정도쓰니까 꼭꼭소리나고 배터리도 금방 옴더라구요ㅠ 이번에 c타입으로 새
836	836	모델 선택: 에어팟 프로 2세대 MTJV... 23/12/17	23/12/17	4	긍정	제가 귀가 많이 작은 타입이라 이게 제대로 안 들어가서 제 기능을 못해요ㅠㅠ 저처럼 것구

프로젝트 기본 정보

- 진행기간 : 2023.12 중
- 프로젝트 목표와 타겟 : 네이버 쇼핑몰에서 이어폰과 관련된 상품페이지에서 리뷰 데이터 추출
 - 리뷰 데이터는 날짜, 별점, 리뷰 내용을 수집
- 진행 히스토리

■ Github1

프로젝트 진행 절차와 코드 정보

- 코드 구성은 웹페이지에서 데이터를 수집하는 클래스 1개, 수집한 데이터를 DBMS에 연동하여 입력하는 클래스 1개로 구성
- 데이터 수집 클래스

- 네이버 쇼핑은 JavaScript 기반으로 크롤링에는 Selenium 사용
- 타겟 URL 접속 - 조회된 리뷰 데이터 수집 - 리뷰 페이지 이동 및 재수집 - 마지막 페이지까지 수집하고 수집 데이터 반환으로 구성됨
- DBMS 연동 클래스
 - 데이터를 입력할 클래스 생성 - 테이블에 데이터 입력 단계로 구성
 - DBMS 연동용 쿼리문 구성에서 Python 스크립트와 SQL문의 구성을 파괴할 수 있는 일부 특수문자에 대한 처리 사전 진행
- 프로젝트 결과와 의의
 - 웹페이지에서 데이터를 크롤링 할 수 있음
 - Python 상에서 Oracle DB와 연동하여 데이터를 직접 입력할 수 있음

4.2. 탐색적 데이터 분석 프로젝트

탐색적 데이터 분석 능력을 확인할 수 있는 프로젝트를 진행한 결과물입니다. 저는 경마 경기 결과를 이용하여 경기 결과와 관련된 여러가지 변수를 탐색하고자 했습니다. 데이터는 한국마사회에서 OpenAPI를 통하여 다수의 양질의 데이터를 제공하고 있어 해당 데이터를 수집하여 사용했습니다.

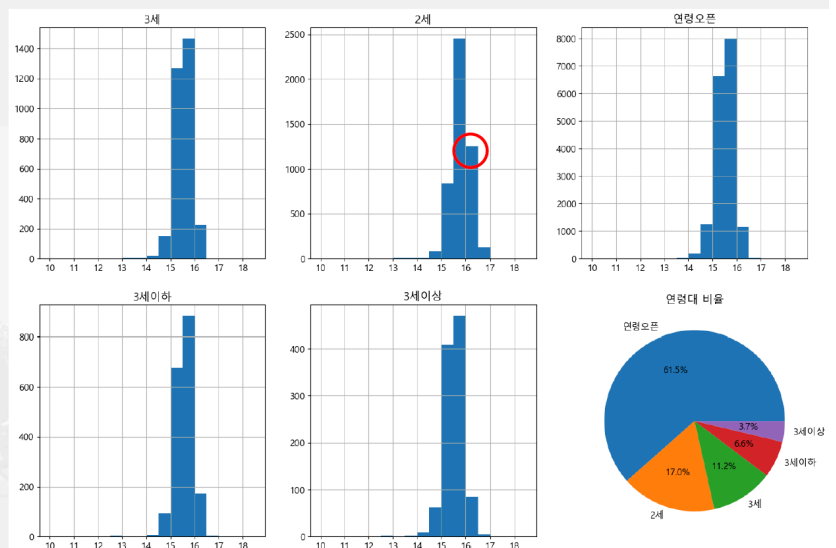
ex) 데이터 분석 결과 예시

분석 결과 도출

- hrSpd와의 상관관계 시각화 - 나이제한 경기별(ageCond)
 - histogram, pieplot 사용
 - 기술통계량 계산

- 낮은 나이의 경기(특히 2세)에서 평균 속력이 훨씬 빠름
- 2세 속력 히스토그램상 16~16.5 속력이 비율이 타 나이에 비해 확연히 높음

	count	mean	std	min	25%	50%	75%	max
3세	3145.0	15.511639	0.373271	10.371651	15.312132	15.527950	15.727392	16.528926
2세	4765.0	15.784070	0.379606	12.448133	15.532100	15.795869	16.025641	16.949153
연령오픈	17243.0	15.493146	0.380516	9.887006	15.306122	15.523933	15.727392	16.750419
3세이하	1839.0	15.541995	0.373973	10.810811	15.348288	15.564202	15.768725	16.891892
3세이상	1039.0	15.517857	0.374184	12.961117	15.306122	15.533981	15.748031	16.750419



다만 프로젝트 진행 과정에서 경기 결과에 영향을 주는 요인을 찾는데 중심이 넘어가서 탐색적 데이터 분석보다는 확증적 데이터 분석과 추론통계에 가까운 결과물이 산출되어 다소 아쉬움이 남는 프로젝트가 되었습니다. 데이터 자체는 탐색적 데이터 분석을 충분히 할 수 있으며 차후 프로젝트 진행 시 원래의 목적을 다시 상기하며 작업을 진행하게 다짐하게 되는 경험이 되었습니다.

- 프로젝트 기본 정보
 - 진행기간 : 2024.01 중
 - 프로젝트 목표와 타겟 : 데이터의 수집과 탐색적 데이터 분석
 - 대상 : 경마 결과 데이터
 - 진행 히스토리
 - [Github1](#)
- 프로젝트 진행 절차
 - 데이터 수집 : 한국마사회에서 제공하는 OpenAPI로 데이터 수집
 - 수집한 데이터에 대한 데이터 분석과 기초통계량 확인 및 각 데이터에 적합한 그래프를 활용하여 데이터 시각화 및 분석 진행
- 프로젝트 결과와 의의
 - OpenAPI를 통한 데이터 수집 과정 이해
 - 데이터 분석과 시각화 능력 향상

4.3. 머신러닝 / 딥러닝 프로젝트

데이터 수집부터 전처리, 머신러닝 / 딥러닝 모델을 직접 선정하여 결과를 산출하고 그 의의를 찾는 프로젝트를 진행했습니다. 해당 프로젝트는 고용노동부에서 주최하는 데이터 공모전에 참여하는 목적이 있었습니다.

제가 참여한 팀에서는 산업 재해 데이터를 수집하고 모델 훈련과 그 결과를 토대로 산업 재해의 원인과 위험도를 사전에 측정하고 산업 재해를 예방할 수 있는지에 관하여 탐구 하였습니다.

ex) 산업 재해 발생 유형을 예측하는 모델 성능 예시

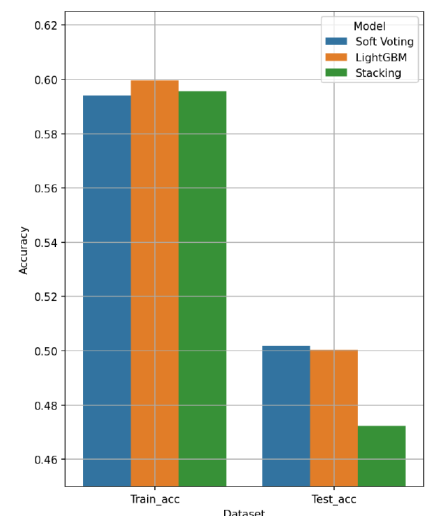
앙상블 모델 성능 측정

- 목적 : 단일 모델로서 일정 이상의 성능을 나타내는 모델들을 앙상블하여 추가적인 성능 향상이 가능한지의 여부를 확인
- 각 모델의 성능을 확인한 결과 XGBoost, LightGBM, CatBoost 3종의 모델을 사용하기로 결정
- 다수의 다른 종류의 모델을 앙상블하는 방법으로 Voting과 Stacking을 선정
 - Stacking의 Final estimator는 XGBoost를 사용



- 앙상블 결과 (Soft) Voting에서 모델 성능이 LightGBM 단일 모델 성능보다 우수한 결과를 보임

	Train_acc	Test_acc	Type
Model			
Soft Voting	0.594155	0.501912	Ensemble
LightGBM	0.599681	0.500425	Single
Stacking	0.595696	0.472376	Ensemble



산업 재해 발생의 원인은 다양한 요인에 의해 발생하지만 발생 여부를 결정하는 가장 주요한 요인은 안전 교육과 방호 대책의 유무였습니다. 모델 성능 또한 기대한 수준으로 확인되어 원인 및 결과 분석, 연구 결과의 도출에 있어서 유의미한 결과가 나왔다고 생각합니다.

- 프로젝트 기본 정보

- 진행기간 : 2024.04 ~
- 프로젝트 목표와 타겟 : 머신러닝 프로젝트의 전체 과정 구성
 - 대상 : 산업 재해 데이터
 - 선정이유 : 산업 재해가 발생하는 요인에 대해 분석하고 이를 예방할 수 있는 방안 모색
- 프로젝트 구성인원 4인
- 진행 히스토리
 - [Github1](#)

- 프로젝트 진행 절차

- 프로젝트 목표 수립
- 프로젝트 진행에 필요한 데이터 수집 (Selenium을 이용하여 데이터 수집)
- 머신러닝 / 딥러닝 적용을 위한 데이터 가공 및 전처리
 - 정제되지 않은 텍스트 Feature에 대한 텍스트 분리 작업
 - 연속형 데이터에 대한 파생변수 생성 (PCA 적용)
 - 성능 향상에 도움이 될 별도 데이터 추가 수집 및 데이터 병합 (기상청 제공 기상 데이터)
- 모델 선정 및 훈련, 하이퍼파라미터 튜닝을 통한 모델 성능 개선 작업

- 프로젝트 공헌 내역

- 데이터 수집을 위한 크롤링 코드 작성 (기여도 70%)
- 수집된 데이터 전처리 작업 (기여도 50%)
- 머신러닝 모델 구현 (기여도 50%)
- 딥러닝 모델 구현 (기여도 100%)

- 프로젝트 결과와 의의

- 데이터 수집 ~ 모델 훈련 및 결과 해석에 이르기까지 머신러닝 프로젝트의 모든 과정을 진행하고 그 프로세스에 대한 능력 향상
- 프로젝트 내용의 의의
 - 산업 재해를 원인을 파악할 수 있는 요인 예측
 - 산업 재해를 예방하기 위한 적합한 방안 제안 가능
 - 연구결과를 국가, 기업, 개인별로 적합하게 활용할 수 있는 계기 마련