|  |  |
| --- | --- |
| 개발기관  배정연번 | 2024-XXXX-XX |

|  |  |
| --- | --- |
| * 1. **개발이력** | |
| * 1. **최초 개발** | 2024.06.01 |
| * 1. **1차 업데이트** |  |
| * 1. **2차 업데이트** |  |
| * 1. **3차 업데이트** |  |

**「T-EXPERT**

**: 삼성전자 온양사업장」**

|  |
| --- |
| **수행 내용 / 인공지능 빅데이터 분석** |

제 출 일 :

소 속 :

성 명 :

| **※ 워크북 제출시 반드시 파일명을 ○○○○○○(제출년월일6자리)\_○○○(성명).hwp로 하여 주십시오.(예시 : 20240621\_홍길동.hwp)** |
| --- |

개발일자 : 2024.06 01

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **훈련과정** | T-EXPERT : 삼성전자 온양사업장 | | | | **훈련기간** | | | | 2024.06.10-2024.06.21 | | |
| **제출일시** | 2024.06.21 | **훈련생** | | (서명) | | | | **훈련교사** | | | 김 대 경 (서명) |
| **훈련직종** | 인공지능기술자  (인공지능엔지니어) | | | **능력단위 또는 교과목** | | | 『1과목』 Python 프로그래밍 기초  『2과목』 Python 데이터 전처리 기초  『3과목』 데이터 분석과 통계, 확률  『4과목』 데이터 분석과 머신러닝  『5과목』 데이터 분석과 인공지능 | | | | |
| **수업목표** | 인공지능 서비스의 요구사항을 실현하기 위한 인공지능 플랫폼의 인프라, 기능, 인터페이스, 지식화를 구현하고 최적화할 수 있는 기술인 육성을 목표로 한다. | | | | | | | | | | |
| **과 제 명** | PBA 공정 데이터 불량률 사전 예지 AI 플랫폼 구축 | | | | | | | | | **과제번호** | |
| **작업유형** | 단독작업 ■ | | 팀 내 단독작업 □ | | | 팀(협동작업) □ | | | |  | |

|  |
| --- |
| 수행 내용 |

󰊱 서비스 기획 요구사항을 통해 인공지능 플랫폼 구축 목적과 범위를 식별한다.

인공지능 플랫폼 서비스 기획의 요구사항과 설정된 인공지능 플랫폼의 목표를 비교하여 누락된 요구사항이 없도록 목적과 구축 범위를 식별하는 단계이다.

머신러닝 기반 데이터 분석의 목적과 비즈니스 문제를 명확히 이해한다.

1. 서비스 방식과 서비스 목적을 검토한다.

※머신러닝 기반 데이터 분석의 목적과 비즈니스 문제를 명확히 이해한다.

가. 해결하려는 문제를 정의한다.

분석이 필요한 주제가 무엇인지 명확히 기술하고, 머신러닝 기반의 데이터 분석이 필요한 영역인지 판단을 내린다. 이때 머신러닝 기반의 데이터 분석을 통해 어떤 부분에서 비즈 니스 성과 개선을 가져오거나 새로운 가치를 창출할 수 있는지 명확하게 정의할 필요가 있다. 만일 머신러닝 기반의 데이터 분석이 아닌 기술 통계기반의 데이터 분석 등이 적합 할 경우 통계기반 데이터 분석 계획하기 수행 절차를 따른다.

나. 문제와 관련된 도메인 영역에 대한 정보 및 문제 해결을 위한 아이디어를 수집한다. 해결하려는 문제와 관련된 도메인 영역 지식과 정보를 수집하고, 조직 내 기존 업무를 분 석하여 이슈를 체계화한다. 이 과정에서 필요하다면 현업 비즈니스 전문가와의 인터뷰를 실시하고, 다양한 내/외부 실무자와의 의견교환을 통해 최초의 문제정의에 대한 관점을 지속적으로 보완 및 점검할 필요가 있다.

|  |
| --- |
| 귀하께서 맡고 계시는 업무와 업무 내의 역할은 어떤 것입니까? |
| 귀하의 팀에서 운영 도는 관리하는 시스템의 종류는 어떤 것이 있습니까? |
| 관련 현업 조직(고객사)의 조직 구성은 어떻게 되어 있습니까? |
| 현업 조직의 업무에 대해 간략하게 설명해 주십시오. |
| 현업 조직의 업무가 현재 ... 시스템을 통해 관리되고 있습니까? 관리되고 있다면 관련된 프로그세스는 몇 개나 있습니까? |
| 귀하께서 맡고 계시는 기능 업무 중 업무적 이슈가 있다면 어떤 것이 있는지 구체적으로 말씀해 주십시오. |

※분석 목표 정의서 예시

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 분석 기본 정의 | 분석 명칭 | PBA 공정 데이터 불량률 사전 예지 AI 플랫폼 구축  XXX의 보안 XXX 모니터링 XXX 데이터의 추세 예측을 통한 사전 예지 Saas 개발 | 분석목표 확정일 | 2022-10-13 |
| 분석 목적 | PBA 공정 데이터 불량률에 대한 추세 모니터링을 통해 장비 점검 시점 사전 예측을 통해 설비가동률 향상 및 불량률 감소, 실패비용 감소  오픈 스택 서버의  CPU I/O Wait  값에 대한 추세 분석을 통한 서버 가용성 고도화를 위한 사전 예지 | 분석 목표 워크숍 | 2022-10-13 |
| 분석 우선순위 | 상 | 담당 조직명 | - |
| 분석 접근 방안 | 과거 품질불량 판정 데이터를 통해 시계열시뮬레이션을 돌려보며 시간/주별불량률 추세 분석과 불량률 사전 예측  이후 사전 예측 결과를 토대로 부품별이슈 파악 후 사전 조치 가능할 수 있도록 하는 것이 목표 | | |
| 성과 측정 | 정성적 기준 | 신규 기법/기술 : 다양한 EDA 및 예측 기법 및 머신러닝/ 딥러닝활용으로Best Model 선정  기존 데이터: 공정에서 발생하는 불량 정보 포함 데이터 정보  신규 데이터 : … | |  |
| 정량적 기준 | 기존 제조공정 대비설비 가동률 50% 이상 향상  제품 불량률 2% 이상 감소(2.5% → 0.05%) 목표 달성 | |  |
| 데이터 정보 | 내부 데이터 | 2020-01-21 부터 2020-01-25까지 A\_Line데이터 셋 | 데이터 입수 난이도 | 중 |
| 외부 데이터 |  | 데이터 입수 난이도 |  |

다. 구축 목표와 범위 정의를 포함한 프로젝트 헌장을 작성한다.

|  |  |  |  |
| --- | --- | --- | --- |
| 프로젝트 현장(Project Charter) | | | |
| 프로젝트 명  (Project Name) | PBA 공정 데이터 불량률 사전 예지 AI 플랫폼 구축 | | |
| 프로젝트 설명  (Project Description) | PBA 공정은 인쇄회로기판에 표면실장(SMT)공정을 거친 것을 말한다. PBA 공정은 고집적화에 따른 제품 성능 향상, PCB 조립의 자동화 쉬움 등 장점이 있는 반면, 부품의 소형화로 불량 수정이 어려운 점 등 단점도 있다.따라서, PBA 공정 데이터 불량률에 대한 추세 모니터링을 통해 장비 점검 시점 사전 예측을 통해 설비가동률 향상 및 불량률 감소, 실패비용 감소를 하고자 한다. | | |
| 프로젝트 매니저  (Project Manager, PM) | 홍길동 | 승인 날짜  (Date Approved) | 2022-10-13 |
| 프로젝트 스폰서  (Project Sponsor) | 대한상공회의소 서울기술교육센터 기업교육팀 | 서명  (Signature) | 홍길동 |
| 비즈니스 케이스  (Business Case) | | 목표(Goals) / 산출물(Deliverables) | |
| 불량률에 대한 추세분석을 통해 장비 점검·교체 | | 제품 생산에 있어 원자재 품질보증 유.무를 검사하여 합격한 제품만 공정으로 투입되고 불합격된 제품은 시정조치를 통한 품질 개선 활동을 하는 공정인 IQC를 비롯, ASS'Y가 완료된 PBA에 대하여 각종 기능을 전수 검사하는 공정인 PBA 성능 및 자동 비전검사 도입 등 지속해서 개선을 통해 성능 향상이 진행됐음에도 불구하고 불량 발생에 대한 사전 예지를 통한 선 대응 체계 마련이 요구되는 상황임. 이에 인공지능 기반 불량 발생에 대한 사전 예지를 통한 장비 점검 및 공구 교환 시기의 적절한 예측 모델 개발을 통해 제품 가공의 최적화와 품질고도화를 이루는 것이 본 기술개발의 목적.  PBA 공정 데이터 불량률 사전 예지 AI 플랫폼 | |
| 팀 구성원(Team Member) | |
| 이름(Name) | 역할(Role) |
| 홍길동 | PM |
| 박문수 | 엔지니어 |
| 위험과 제약사항(Risk and Constraints) | | 주요 일정(Milestones) | |
|  |  |  | 1.문제 정의(Problem Definition)와 question, 알고리즘 선정 |
|  |  |  | 2.헌장(Charter) |
|  |  |  | 3.작업환경 만들기 |
|  |  |  | 4.데이터 획득(Data Acquisition) |
|  |  |  | 5.EDA : Exploratory Data Analysis : 탐색적 데이터 분석(Data Exploration and Analysis)과 데이터 시각화(Data Visualization)를 통한 insight 얻기 |
|  |  |  | 6.Machine Learning Algorithms을 위한 데이터 준비 |
|  |  |  | 7.통계적 모델링 혹은 모형화(Statistical Modeling) 혹은 확증적 데이터 분석(Confirmatory Data Analysis, CDA) |
|  |  |  | 8.효과 검증 |
|  |  |  | 9.Data 도전과 배움 |
|  |  |  | 10.서비스 구현, 활용 계획 |

2. 머신러닝 기반 데이터 분석 시나리오를 구성한다.

분석 주제 및 목적, 사용하고자 하는 머신러닝 기법과 방법론, 필요한 데이터 수집 및 확 보 방법, 분석 실행 절차, 소요 예상 일정, 비용이 발생할 경우 이에 대한 예상 비용 등을 체계적으로 기술한다. 또한, 분석 과정에서 발생할 것으로 예상되는 장애 요인과 이에 대 한 해결 대안 등에서도 기술하는 것이 바람직하다.

|  |  |  |
| --- | --- | --- |
| PBA 공정 데이터 불량률 사전 예지 AI 플랫폼 구축 | | |
| 목표 | PBA 공정 데이터 불량률에 대한 추세 모니터링을 통해 장비 점검 시점 사전 예측을 통해 설비가동률 향상 및 불량률 감소, 실패비용 감소 | |
| 핵심개념 | 2022-04-19\_GE\_PBA\_B\_검사서 2020-01-21\_A\_Line~2020-01-25\_A\_Line까지 파일 5개를 대상으로 기계학습 알고리즘 중 시계열 분석 중 ARIMA 모형과 페이스북의 라이브러리, 그리고 LSTM을 사용 시계열 분석을 함.  시계열 분석이란?  시간의 흐름에 따라 일정한 간격으로 변화하는 데이터를 ‘시계열 자료’라고 부르는데, 이러한 시계열 자료를 통해서 미래에 대한 정보를 예측하고자 이러한 분석을 하게 된다. 소비자물가지수(consumer price index) 분석, 종합주가지수(KOSPI) 분석, 주가 동향 분석 등의 경제 분야뿐만 아니라, 해마다 같은 기간의 감기환자 추이를 분석하거나, 유행성 독감과 같은 특정 질병에 대한 환자의 수 변화추이를 분석하는 등의 의료 분야, 특정 시간대별로 변화하는 교통사고 사망 자 증감 혹은 사고발생 다발지역, 다발시간 분석, 미세먼지 발생 가능 성 예측 등의 사회 분야까지, 더욱 광범위한 분야에서 활발하게 쓰이는 분석기법이다.  ㅇ 이렇듯 시계열 분석은 일반적이고 보편화한 기계학습 방법론이자 다양한 통계 예측(Forecast) 기법의 하나로써, 독립변수((Independent variable), 또는 설명변수(explanatory variable))를 이용하여 종속변수 ((Dependent variable), 또는 반응변수(response variable))를 예측하는 것을 말한다. | |
| 데이터 수집 | ○○○ 데이터 셋 : ○○○ 저장소에서 다운로드한 2022-04-19\_GE\_PBA\_B\_검사서 2020-01-21\_A\_Line~2020-01-25\_A\_Line까지 파일 5개를 대상으로 하였다. | |
| 데이터 준비 | 수집한 데이터 파일 병합 | |
| 데이터 탐색 | 1. 정보 확인 : info()  2. 기술 통계 확인 : describe(), unique(), value\_counts() | |
| 결과 시각화 | | |
| xgb\_model을이용한 피처 엔지니어링 | | TimeSeries분석을 이용한 시각화 |
|  | |  |

3. 데이터 확보 계획을 작성한다.

머신러닝 기반 데이터 분석을 위한 데이터 확보 계획을 수립하고 필요 데이터를 수집한다.

분석하고자 하는 이슈와 분석 시나리오가 명확하게 도출되고 나면, 필요한 데이터를 어떻게 확보할 것인지 세부 수집 계획서를 작성한다. 조직 내 전사 데이터웨어하우스 혹은 데 이터 마트 등에 축적된 데이터의 레이아웃 등을 검토하여 필요 데이터의 저장형태 및 활용 가능성을 판단하는 것이 일반적인 절차라고 할 수 있다. 그러나 전사 데이터웨어하우스 등에 축적되고 있지 않은 데이터가 필요하다면, 원천 소스 영역에서 어떤 형태와 방식으로 데이터를 확보할 것인지 추가적인 검토가 필요하게 된다. 이 경우 조직 내 IT 부서 담당자 혹은 데이터 엔지니어 등과 회의를 통해 수집방안 등에 대한 세부검토가 필요하다. 만일 외부 데이터가 필요하다고 판단될 경우, 추가로 발생할 수 있는 비용 등을 고려하여 특정 외부 데이터에 대한 수집계획을 별도로 작성하여야 한다.

가. 데이터 확보 계획에 따라 필요한 데이터를 수집한다.

필요 데이터가 조직 내 데이터 웨어하우스나 하둡 등 빅데이터 저장소에 축적되어 있을 경우 SQL이나 HiveQL 등을 통해 필요한 데이터를 추출하게 된다. 필요한 데이터가 분석 계 저장소가 아닌 원천 운영 시스템에만 존재한다면, 운영시스템 접근 및 조회 권한이 있 는 데이터 엔지니어 등과의 협업을 통해 필요 데이터를 추가로 연동해야 한다. 만일 외부 데이터 수집이 필요한 경우, 웹 사이트 크롤링, 공공 데이터 API 연동, 데이터 구매 등의 절차를 통해 데이터를 별도로 수집해야 한다.

데이터 수집 세부 계획서

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PBA 공정 데이터 불량률 사전 예지 AI 플랫폼 구축** 데이터 수집 세부 계획서 | | | | | | | | | | | |
| **PBA 공정 데이터 불량률 사전 예지 AI 플랫폼 구축**을 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 | | 레코드단위 | 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 | | | | | | |  | | | | |

󰊲 인프라(하드웨어, 소프트웨어, 네트워크) 환경을 설계한다.

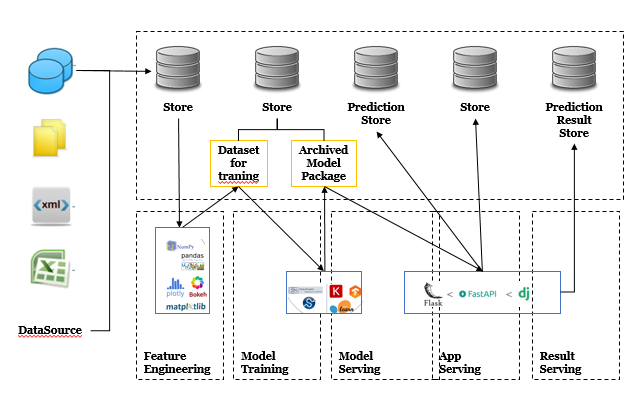
1. 머신러닝 수행을 위한 하드웨어 및 소프트웨어, 플랫폼 등을 구성한다.

머신러닝 기반 데이터 분석의 지속성, 비즈니스에의 중요도 및 의사결정 영향도, 조직 내 가용 예산, 자체 기술력 및 유지보수 가능성 등을 고려하여 상용 시스템이나 오픈소스 솔 루션 등을 구성한다. 머신러닝 기반 데이터 분석에 자주 사용되는 소프트웨어 혹은 프로 그래밍 환경은 R, 파이썬(Python), SAS, JAVA, Scala 등이 있으며, 특히 현업에서는 다양한 라이브러리와 확장성 등의 이유로 R과 파이썬이 많이 사용되고 있다. 또한, 빅데이터 플 랫폼 환경 혹은 상용 서비스 환경에서는 스파크 혹은 머하웃 라이브러리 등을 사용하여 머신러닝을 배치 혹은 실시간 형태로 수행하기도 한다.

클라우드 환경에서는 다양한 인프라(컴퓨팅, 저장 장치, 네트워크 등)와 기계학습 플랫폼 (TensorFlow Server, Cluster, 기계학습 라이브러리, 데이터 라벨링 등), 관리 환경이 사업 자에 의해 제공되므로 이 실습 수행은 설치형 환경으로 인프라 설계를 하는 것으로 한다.

1.인프라 환경을 설계한다.

인공지능 서비스 플랫폼의 인프라는 데이터 수집 > 데이터 정제 > 모델 학습 > 모델 배포 > 앱 배포 > 분석 > 서비스 반영’ 까지 전반적인 머신러닝 프로세스를 수행할 수 있도록 설계한다.



2. 소프트웨어 환경을 설계한다.

제공하고자 하는 인공지능 서비스별로 논리적 측면에서 소프트웨어 환경을 구성한다.

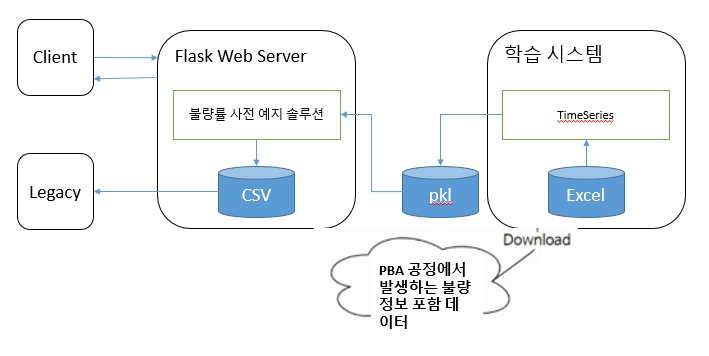
(1) 데이터 수집/전처리 소프트웨어 환경을 설계한다.

(2) 데이터 저장 소프트웨어 환경을 설계한다.

(3) 데이터 분석/처리 소프트웨어 환경을 설계한다.

(4) 데이터 서비스 소프트웨어 환경을 설계한다.

환경 설계 구성 예



3. 작업환경 만들기

|  |
| --- |
| □ 개발 스택 |

가.학습플랫폼 서비스

|  |
| --- |
| 학습플랫폼 서비스 |
| Jupyternotebook |
| openpyxl/xlrd/statsmodels//nbformat  Plotly  Pystan  ipykernel  pandas 버전 : 1.4.3  matplotlib버전 : 3.5.2  numpy버전 : 1.22.4  scipy버전 : 1.8.1  IPython버전 : 8.4.0  sklearn: 1.1.1  fbprophet  joblib |
| Tensorflow2 |
| Miniconda-Python 3.7.13 (default, Mar 28 2022, 08:03:21)[MSC v.1916 64 bit (AMD64)] |
| 윈10 |
| TCP/IP |
| Model Serving - 윈10에서 MinicondaPrompt를 이용해 가상환경 만들기 |

나.서비스 플랫폼

|  |
| --- |
| 서비스 플랫폼 |
| Visual StudionCode |
| sqlite  flask  Fbprophet  Plotly  pystan  scikit-learn==0.21.3  numpy==1.19.5  highcharts  joblib |
| Tensorflow2 |
| Miniconda-Python 3.7.13 (default, Mar 28 2022, 08:03:21)[MSC v.1916 64 bit (AMD64)] |
| 윈10 |
| TCP/IP |
| App Serving - Flask 개발 환경 만들기 |

|  |
| --- |
| □ 윈10에서 Anaconda Prompt를 이용해 가상환경 만들기 |

ㅇ OS

윈10

ㅇ python –version

-----

Python 3.7.13

ㅇ Tensorflow2를 기준으로 설치

- v가상환경(Virtual Environment)을 이용하면 Python 버전 간의 의존성을 고려해서 가상의 격리된 환경을 만들어줌으로써 버전이 다름으로 인해 발생할 수 있는 호환이나 충돌 문제를 미연에 방지할 수 있음.vMiniconda설치

1. 윈도우 환경이라면 Miniconda3 Windows 64-bit를 다운로드합니다.

<https://docs.conda.io/en/latest/miniconda.html>

|  |
| --- |
| (base) C:\Users\DaeKyeong>conda create --name tf37\_cpu python=3.7  (base) C:\Users\DaeKyeong>conda activate tf37\_cpu |

ㅇ 텐서플로2

|  |
| --- |
| (tf37\_cpu) C:\Users\DaeKyeong>pip install --upgrade tensorflow  Collecting tensorflow  Downloading tensorflow-2.9.1-cp37-cp37m-win\_amd64.whl (444.0 MB)  |████████████████████████████████| 444.0 MB 131 kB/s  (tf37\_cpu) C:\Users\DaeKyeong>python  Python 3.7.13 (default, Mar 28 2022, 08:03:21) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32  Type "help", "copyright", "credits" or "license" for more information.  >>> import tensorflow as tf  >>> quit()  (tf37\_cpu) C:\Users\DaeKyeong>pip install ipykernel  Collecting ipykernel  Using cached ipykernel-6.13.1-py3-none-any.whl (133 kB)  Collecting nest-asyncio  Using cached nest\_asyncio-1.5.5-py3-none-any.whl (5.2 kB)  …  (tf37\_cpu) C:\Users\DaeKyeong>python -m ipykernel install --user --name tf37\_cpu --display-name "tf37\_cpu"  Installed kernelspec tf37\_cpu in C:\Users\apr0621\AppData\Roaming\jupyter\kernels\tf37\_cpu  (tf37\_cpu) C:\Users\DaeKyeong> |

|  |
| --- |
| □ 라이브러리 설치 |

ㅇ fbprophet 설치

|  |
| --- |
| (tf37\_cpu) C:\Users\DaeKyeong>conda install -c plotly plotly  (tf37\_cpu) C:\Users\DaeKyeong>pip install pystan  (tf37\_cpu) C:\Users\DaeKyeong>pip install --upgrade setuptools  (tf37\_cpu) C:\Users\DaeKyeong>conda install -c conda-forge fbprophet  (tf37\_cpu) C:\Users\DaeKyeong>python  Python 3.7.12 | packaged by conda-forge | (default, Oct 26 2021, 05:35:01) [MSC v.1916 64 bit (AMD64)] on win32  Type "help", "copyright", "credits" or "license" for more information.  >>> from fbprophet import Prophet  >>> quit()  (tf37\_cpu) C:\Users\DaeKyeong> |

ㅇ openpyxl 설치

|  |
| --- |
| pip install openpyxl |

ㅇ statsmodels 설치

|  |
| --- |
| pip install statsmodels |

ㅇ nbformat 설치

|  |
| --- |
| pip install nbformat  커널 리부팅해야 함!!!! |

|  |
| --- |
| □ 설정 |

작업경로 설정 등을 한다.

C 드라이브의 볼륨에는 이름이 없습니다.

볼륨 일련 번호: 2A59-BA4A

C:\Users\908\_00\mlops-works 디렉터리

[.] [..]

[.ipynb\_checkpoints] 2020-01-21\_A\_Line.xlsx

2020-01-22\_A\_Line.xlsx 2020-01-23\_A\_Line.xlsx

2020-01-24\_A\_Line.xlsx 2020-01-25\_A\_Line.xlsx

learning\_project\_ex1.ipynb

6개 파일 5,302,624 바이트

3개 디렉터리 401,720,279,040 바이트 남음

|  |
| --- |
|  |
| 라이브러리 설치1 |

|  |
| --- |
|  |
| 라이브러리 설치2 |

|  |
| --- |
|  |
| 작업경로 설정 |

Data Ingestion(수집), Data Acquisition(취득,획득

1. 데이터 가져오기

|  |
| --- |
| □ 데이터 선정 |

ㅇ 학습에 사용된 데이터는 깃 저장소(2020-01\_A\_Line.zip: https://github.com/kznetwork/KoreaTech\_SamsungElectronics/blob/main/datasets/2020-01\_A\_Line.zip)에서 다운로드한 2020-01-21\_A\_Line~2020-01-25\_A\_Line까지 파일 5개를 대상으로 하였다.

ㅇ 여러 개의 통합 문서 합치기

|  |
| --- |
| import openpyxl  import pandas as pd  import glob  import os  all\_workbooks = glob.glob('\*.xlsx')  data\_frames = []  for workbook in all\_workbooks:  all\_worksheets = pd.read\_excel(workbook, sheet\_name=None, index\_col=None)  for worksheet\_name, data in all\_worksheets.items():  data\_frames.append(data)  all\_data\_concatenated = pd.concat(data\_frames, axis=0, ignore\_index=True)  writer = pd.ExcelWriter("A\_Line\_2020\_01.xlsx")  all\_data\_concatenated.to\_excel(writer, sheet\_name='all\_data\_all\_workbooks', index=False)  writer.save() |

#불러올 파일의 경로를 filename 변수에 저장

filename = './A\_Line\_2020\_01.xlsx'

import pandas as pd

#pandas read\_excel로 불러오기

A\_Line\_2020\_01 = pd.read\_excel(filename)

A\_Line\_2020\_01.head()

머신러닝 기반 데이터 분석을 위한 분석 환경 구축, 데이터 이해 및 전처리를 수행한다.

2. 머신러닝 적용을 위한 데이터 전처리와 기초 이해를 수행한다.

본격적인 머신러닝 기법을 데이터에 적용하기 전에 일반적으로 다음과 같은 데이터 전처 리와 기초 탐색을 반복적으로 수행하며 데이터 준비와 이해도를 동시에 점진적으로 높여 간다.

(1) 결측지 및 이상치 처리

(2) 데이터 형태 변환, 수치 가공

(3) 데이터 정규화 및 표준화

(4) 데이터 집계 및 요약

(5) 탐색적 관점의 데이터 시각화

4. Machine Learning Algorithms을 위한 데이터 준비

ㅇ 앞서 진행한 예측 변수와 레이블을 분리와 Text와 Categorical 속성 처리하기

|  |
| --- |
| □ 특징 가공(Feature engineering) |

|  |
| --- |
| Result=pd.get\_dummies(data["Result"])  Result.head(10) |

FAIL OK

0 0 1

1 0 1

2 0 1

3 0 1

4 0 1

5 0 1

6 0 1

7 0 1

8 0 1

9 0 1

|  |
| --- |
| data=pd.concat([data,Result],axis=1)  data.head() |

Date Time Result Periods WRITING BLE MAC ADDRESS FCTVER MLBSERIAL FATPSERIAL DSNSERIAL ... F1\_F0 FN\_FN5 SET\_CH.1 SET\_Frequency(MHz).1 RSSI RX\_PER BLE DEVICENAME BLE RSSI FAIL OK

0 2022-05-07 08:36:55 OK 15.3 OK AC4D16DA9128 Dec 15 2021 GRS65006337020A25700000 WIP25071QUH100000 25071QUH100000 ... NaN NaN NaN NaN NaN NaN NaN NaN 0 1

1 2022-05-07 08:36:55 OK 14.8 OK AC4D16DA914C Dec 15 2021 GRS65006337020A25700003 WIP25071QUH100003 25071QUH100003 ... NaN NaN NaN NaN NaN NaN NaN NaN 0 1

2 2022-05-07 08:36:55 OK 15.0 OK AC4D16DA913E Dec 15 2021 GRS65006337020A25700001 WIP25071QUH100001 25071QUH100001 ... NaN NaN NaN NaN NaN NaN NaN NaN 0 1

3 2022-05-07 08:36:55 OK 14.6 OK AC4D16DA9169 Dec 15 2021 GRS65006337020A25700002 WIP25071QUH100002 25071QUH100002 ... NaN NaN NaN NaN NaN NaN NaN NaN 0 1

4 2022-05-07 08:36:55 OK 14.3 OK AC4D16DA9151 Dec 15 2021 GRS65006337020A25700004 WIP25071QUH100004 25071QUH100004 ... NaN NaN NaN NaN NaN NaN NaN NaN 0 1

|  |
| --- |
| Date\_Result=data[['Date','FAIL']]  Date\_Result.head()  Date\_Result['Date'].value\_counts() |

2022-05-10 44532

2022-04-21 40692

2022-04-22 35892

2022-04-19 35568

2022-05-07 31932

2022-05-04 29178

2022-04-25 29034

2022-05-01 28350

2022-04-26 27360

2022-05-05 25668

2022-04-23 24750

2022-05-06 23850

2022-04-27 23616

2022-05-03 23418

2022-04-20 23364

2022-04-18 20700

2022-04-11 19782

2022-04-13 18894

2022-04-15 18450

2022-04-16 17100

2022-04-29 16182

2022-04-12 15750

2022-05-08 15570

2022-04-14 14400

2022-05-09 10530

Name: Date, dtype: int64

|  |
| --- |
| Data Cleaning and Munging |

|  |
| --- |
| Date\_Result\_gs=Date\_Result.groupby('Date')['FAIL'].agg(\*\*{'FAIL':'sum'}).reset\_index()  Date\_Result\_gs |

Date FAIL

0 2022-04-11 44

1 2022-04-12 4

2 2022-04-13 19

3 2022-04-14 29

4 2022-04-15 29

5 2022-04-16 6

6 2022-04-18 8

7 2022-04-19 25

8 2022-04-20 10

9 2022-04-21 37

10 2022-04-22 44

11 2022-04-23 19

12 2022-04-25 22

13 2022-04-26 23

14 2022-04-27 30

15 2022-04-29 6

16 2022-05-01 17

17 2022-05-03 12

18 2022-05-04 7

19 2022-05-05 11

20 2022-05-06 14

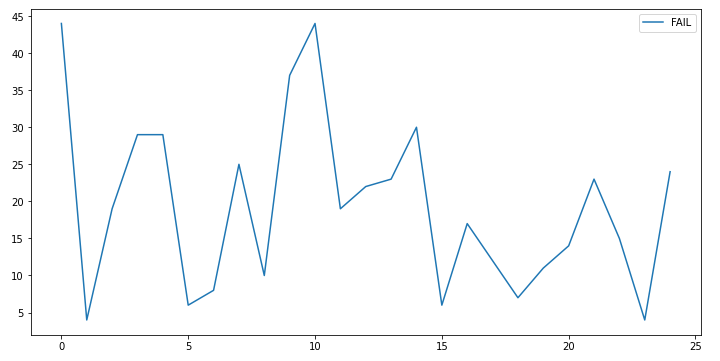
21 2022-05-07 23

22 2022-05-08 15

23 2022-05-09 4

24 2022-05-10 24

|  |
| --- |
| import matplotlib.pyplot as plt  Date\_Result\_gs.plot(figsize=(12,6))  plt.show() |



5.필요한 머신러닝 기법에 따른 모델 훈련을 수행한다.

데이터에 대한 전처리와 이해가 어느 정도 단계에 도달하면, 필요한 머신러닝 기법을 적 용하게 된다. 지도학습 기법을 적용할 경우, 데이터 세트를 학습용 데이터와 평가용 데이 터로 분할하고 모델 훈련을 진행하게 되며 자율학습(혹은 비 지도학습) 기법을 적용할 경 우, 데이터 세트 분할보다는 탐색적 패턴도출 형태의 분석을 수행하게 된다. 이 과정에서 단일한 머신러닝 기법 적용으로 훈련이 완료되기보다는 다양한 머신러닝 기법을 적용하며 예측정확도를 높이거나 오차를 감소시키는 과정을 거치게 된다.

|  |
| --- |
| □ Facebook Prophet 학습 진행 |

ㅇ 타켓 데이터 불량률에 대해 이전 값이 이후 값에 영향을 미쳐 평균값이 지속적으로 증가하거나 감소하는 추세를 확인할 수 있다.

|  |
| --- |
|  |
| 데이터 셋 준비 |

|  |
| --- |
|  |
| Facebook Prophet 예측 |

|  |
| --- |
|  |
| Facebook Prophet 예측 추세선 확인 |

ㅇ 학습결과 백업

|  |
| --- |
|  |
| 학습 결과 확인 |

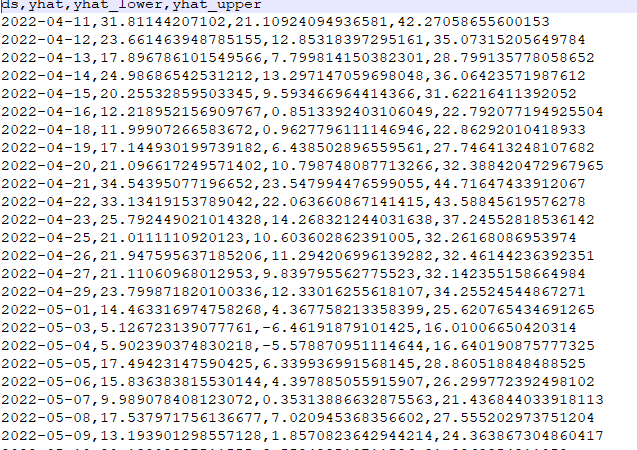
6. 과제 수행 후 단계

머신러닝 결과 도출된 산출물을 유관부서 이해관계자와 공유하기 위한 산출물 및 문서화 를 계획하고 관리한다.

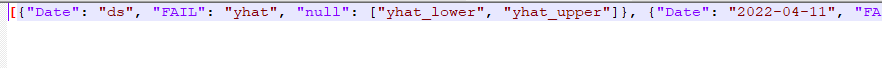
ㅇ 학습 모델 결과 파일과 REST API 활용을 위한 데이터 호환 포맷 데이터 제공

|  |
| --- |
|  |
| REST API 활용을 위한 데이터 호환 포맷 데이터 내보내기 |

Prophet\_forecast.csv 파일



Prophet\_forecast.json



|  |
| --- |
| □ 웹 서비스 플랫폼 |

ㅇ flask 가상환경 생성

|  |
| --- |
|  |
| flask 가상환경 생성 |

ㅇ fbprophet 설치

|  |
| --- |
| ㅇ fbprophet 설치  (flask\_37) C:\DEV\flask\_works>conda install -c plotly plotly  (flask\_37) C:\DEV\flask\_works>pip install pystan  (flask\_37) C:\DEV\flask\_works>pip install --upgrade setuptools  (flask\_37) C:\DEV\flask\_works>conda install -c conda-forge fbprophet  (flask\_37) C:\DEV\flask\_works>python  Python 3.7.12 | packaged by conda-forge | (default, Oct 26 2021, 05:35:01) [MSC v.1916 64 bit (AMD64)] on win32  Type "help", "copyright", "credits" or "license" for more information.  >>> from fbprophet import Prophet  >>> quit()  (tf37\_cpu) C:\Users\k8s> |
| python –version  -----  Python 3.7.4  Fbprophet 설치 |

ㅇ 필요 라이브러리 설치

|  |
| --- |
| scikit-learn==0.21.3 설치 |

(flask\_37) C:\DEV\flask\_works\flask\_webapps>pip install -U scikit-learn==0.21.3

Collecting scikit-learn==0.21.3

Downloading scikit\_learn-0.21.3-cp37-cp37m-win\_amd64.whl (5.9 MB)

|████████████████████████████████| 5.9 MB 726 kB/s

Requirement already satisfied: numpy>=1.11.0 in c:\dev\anaconda3\envs\flask\_37\lib\site-packages (from scikit-learn==0.21.3) (1.21.6)

Requirement already satisfied: scipy>=0.17.0 in c:\dev\anaconda3\envs\flask\_37\lib\site-packages (from scikit-learn==0.21.3) (1.7.3)

Requirement already satisfied: joblib>=0.11 in c:\dev\anaconda3\envs\flask\_37\lib\site-packages (from scikit-learn==0.21.3) (1.1.0)

Installing collected packages: scikit-learn

Attempting uninstall: scikit-learn

Found existing installation: scikit-learn 1.0.2

Uninstalling scikit-learn-1.0.2:

Successfully uninstalled scikit-learn-1.0.2

Successfully installed scikit-learn-0.21.3

(flask\_37) C:\DEV\flask\_works\flask\_webapps>

(flask\_37) C:\DEV\flask\_works\flask\_webapps>pip uninstall numpy

Found existing installation: numpy 1.21.6

Uninstalling numpy-1.21.6:

Would remove:

c:\dev\anaconda3\envs\flask\_37\lib\site-packages\numpy-1.21.6.dist-info\\*

c:\dev\anaconda3\envs\flask\_37\lib\site-packages\numpy\\*

c:\dev\anaconda3\envs\flask\_37\scripts\f2py.exe

Proceed (Y/n)? y

Successfully uninstalled numpy-1.21.6

(flask\_37) C:\DEV\flask\_works\flask\_webapps>pip install numpy==1.19.5

Collecting numpy==1.19.5

Downloading numpy-1.19.5-cp37-cp37m-win\_amd64.whl (13.2 MB)

|████████████████████████████████| 13.2 MB 6.8 MB/s

Installing collected packages: numpy

Successfully installed numpy-1.19.5

(flask\_37) C:\DEV\flask\_works\flask\_webapps>pip uninstall scipy

Found existing installation: scipy 1.7.3

Uninstalling scipy-1.7.3:

Would remove:

c:\dev\anaconda3\envs\flask\_37\lib\site-packages\scipy-1.7.3.dist-info\\*

c:\dev\anaconda3\envs\flask\_37\lib\site-packages\scipy\\*

Proceed (Y/n)? y

Successfully uninstalled scipy-1.7.3

(flask\_37) C:\DEV\flask\_works\flask\_webapps>pip install scipy==1.1.0

Collecting scipy==1.1.0

Downloading scipy-1.1.0-cp37-none-win\_amd64.whl (30.9 MB)

|████████████████████████████████| 30.9 MB 6.4 MB/s

Requirement already satisfied: numpy>=1.8.2 in c:\dev\anaconda3\envs\flask\_37\lib\site-packages (from scipy==1.1.0) (1.19.5)

Installing collected packages: scipy

Successfully installed scipy-1.1.0

(flask\_37) C:\DEV\flask\_works\flask\_webapps>

|  |
| --- |
| □ PBA\_불량률\_예측 플랫폼 솔루션 개발 |

ㅇ main.py

|  |
| --- |
| import flask  from flask import Flask, redirect, url\_for, request, render\_template  from werkzeug.utils import secure\_filename  import pandas as pd  #from sklearn.externals import joblib  import joblib  import numpy as np  from scipy import misc  app = Flask(\_\_name\_\_)  # 메인 페이지 라우팅  @app.route("/")  @app.route("/index")  def index():  return flask.render\_template('index.html')  # 데이터 예측 처리  @app.route('/predict', methods=['POST'])  def make\_prediction():  if request.method == 'POST': # POST 방식으로 전달된 경우  f = request.files['predictor']  # 파일 객체 혹은 파일 스트림을 가져오고, html 파일에서 넘겨지는 값의 이름을 file1으로 했기 때문에 file1임.  f.save(f'test/{secure\_filename(f.filename)}') # 업로드된 파일을 특정 폴더에저장하고,  #df\_to\_html = pd.read\_csv(f'test/{secure\_filename(f.filename)}').to\_html() # html로 변환하여 보여줌  df\_to\_html = pd.read\_csv(f'test/{secure\_filename(f.filename)}')    # 입력 받은 이미지 예측  forecast = model.predict(df\_to\_html)    #print("예측: {}".format(forecast))  #print(forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].head())  # 결과 리턴  #return render\_template('index.html', label=forecast.item())    # 예측 값을 1차원 배열로부터 확인 가능한 문자열로 변환  label = str(np.squeeze(forecast))  # 숫자가 10일 경우 0으로 처리  if label == '10': label = '0'  # 결과 리턴  return render\_template('index.html', label=label)  if \_\_name\_\_ == '\_\_main\_\_':  # 모델 로드  # ml/model.py 선 실행 후 생성  model = joblib.load('./model/PBA\_model.pkl')  # Flask 서비스 스타트  app.run(host='0.0.0.0', port=8000, debug=True) |
| main.py |

ㅇ index.html

|  |
| --- |
| <html>  <head>  <title>PBA\_검사서 Model as a Flask API</title>  <link rel="stylesheet" href="{{ url\_for('static', filename = 'style.css') }}">  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <h1>Predictor PBA\_검사서</h1>  <div class="agile-its">  <h2>Flask with ML Web Service</h2>  <div class="w3layouts">  <div class="photos-upload-view">  <form id="upload" action="/predict" method="POST" enctype="multipart/form-data">  <div class="upload-btn-wrapper">  <button class="btn">파일 업로드</button>  <input type="file" value="Upload" name="predictor">  </div>  <input type="submit" value="예측 결과">  {% if label %}  <span class="result\_lable">  {{ label }}  </span>  {% endif %}  </form>  </div>  </div>  </div>  </body>  </html> |
| index.html |

style.css

`

/\*-- Reset-Code --\*/

html,body,div,span,applet,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,a,abbr,acronym,address,big,cite,code,del,dfn,em,img,ins,kbd,q,s,samp,small,strike,strong,sub,sup,tt,var,b,u,i,dl,dt,dd,ol,nav ul,nav li,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,article,aside,canvas,details,embed,figure,figcaption,footer,header,hgroup,menu,nav,output,ruby,section,summary,time,mark,audio,video{margin:0;padding:0;border:0;font-size:100%;font:inherit;vertical-align:baseline;}

article, aside, details, figcaption, figure,footer, header, hgroup, menu, nav, section {display: block;}

ol,ul{list-style:none;margin:0px;padding:0px;}

blockquote,q{quotes:none;}

blockquote:before,blockquote:after,q:before,q:after{content:'';content:none;}

table{border-collapse:collapse;border-spacing:0;}

/\* start editing from here \*/

a{text-decoration:none;}

.txt-rt{text-align:right;}/\* text align right \*/

.txt-lt{text-align:left;}/\* text align left \*/

.txt-center{text-align:center;}/\* text align center \*/

.float-rt{float:right;}/\* float right \*/

.float-lt{float:left;}/\* float left \*/

.clear{clear:both;}/\* clear float \*/

.pos-relative{position:relative;}/\* Position Relative \*/

.pos-absolute{position:absolute;}/\* Position Absolute \*/

.vertical-base{vertical-align:baseline;}/\* vertical align baseline \*/

.vertical-top{vertical-align:top;}/\* vertical align top \*/

nav.vertical ul li{display:block;}/\* vertical menu \*/

nav.horizontal ul li{display: inline-block;}/\* horizontal menu \*/

img{max-width:100%;}

/\*-- //Reset-Code --\*/

body {

background:#00BCD4;

background-size: cover;

font-family: 'Open Sans', sans-serif;

background-attachment: fixed;

background-position: center;

}

h1,h2,h3,h4,h5,h6{

font-family: 'Montserrat', sans-serif;

}

h1{

color:#fff;

font-size:45px;

text-align:center;

margin-top:50px;

}

.agile-its{

width:35%;

margin:50px auto;

background:#fff;

}

h2{

color:#fff;

background: #f3c500;

padding:10px 0;

text-align:center;

font-size:20px;

}

.w3layouts{

padding:50px;

}

.w3layouts p{

font-size:14px;

line-height:27px;

color:#000;

margin-bottom:10px;

}

.upload-btn-wrapper {

position: relative;

overflow: hidden;

display: inline-block;

bottom: 20px;

left: 130px;

}

.btn {

border: 2px solid gray;

color: gray;

background-color: white;

padding: 8px 20px;

border-radius: 8px;

font-size: 20px;

font-weight: bold;

}

.upload-btn-wrapper input[type=file] {

font-size: 100px;

position: absolute;

left: 0;

top: 0;

opacity: 0;

}

h3{

text-align:center;

margin-top:10px;

}

.result\_lable{

color: #ff6000;

font-weight: bold;

}

input[type="submit"]{

width: 54%;

background:#00BCD4;

border: none;

padding: 10px 0;

color: #fff;

font-size:14px;

cursor:pointer;

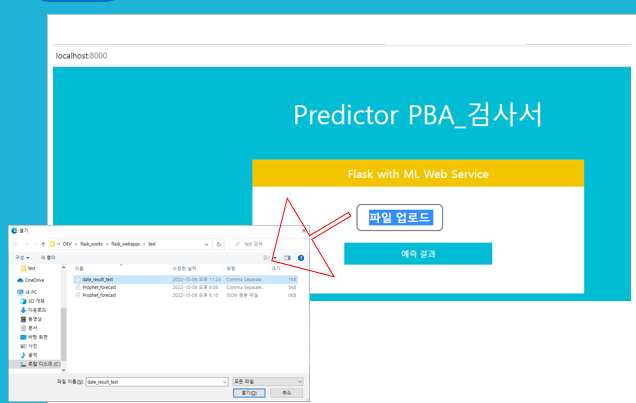
margin-left:23%;

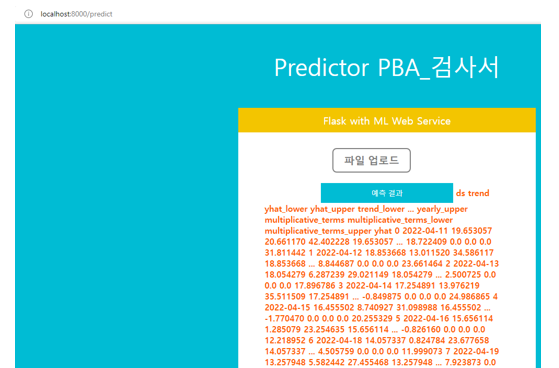
}

ㅇ PBA\_불량률\_예측 플랫폼 솔루션 실행

(flask\_37) C:\DEV\flask\_works\flask\_webapps>python main.py

http://127.0.0.1:8000/





ㅇ 사전 예측기 솔루션은 본 과제 PBA 조립공정 불량률 감소를 위한 개선을 하는 데 기여했다.

󰊱 인터페이스 설계 요건에 따라 외부 인터페이스를 구현한다.

인터페이스 설계서에 명시된 송/수신 시스템에 따라 연계 방식과 저장 유형은 다르게 구 성이 된다. 시스템별 데이터의 특성 등을 고려하여 구현해야 한다.

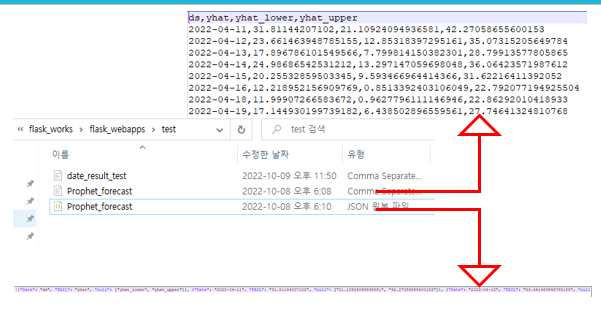
1. 수집 환경의 외부 인터페이스를 구현한다.

(1) 인터페이스 설계 요건에 따라 구성된 인터페이스 환경을 확인한다.

본 과제에서의 인터페이스 명세서

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I/F 번호 | | IF1 | | | | | I/F명 | | | PBL 학습 플랫폼과 서비스 플랫폼 연계 | | | |
| 송신 | I/F ID | A\_ | | | | | 수신 | I/F ID | | G\_SYS | | | |
| I/F명 | PBL 학습 플랫폼 정보 시스템 | | | | | I/F명 | | 서비스 플랫폼 | | | |
| 주기 및 방식 | | 매일 12시 정각 | | | | | DB 및 파일 형식 | | | DB | | | |
| 송신 | | | | | | | 수신 | | | | | | |
| 한글명 | | 영문명 | Type | 길이 | PK | Code 여부 | 한글명 | | 영문명 | Type | 길이 | PK | Code여부 |
| 날짜 | | ds | DATE |  |  |  | 날짜 | | ds | DATE |  |  |  |
| 예측값 | | yhat | FLOAT |  |  |  | 예측값 | | yhat | FLOAT |  |  |  |
| 예측 하한값 | | yhat\_lower | FLOAT |  |  |  | 예측 하한값 | | yhat\_lower | FLOAT |  |  |  |
| 예측 상한값 | | yhat\_upper | FLOAT |  |  |  | 예측 상한값 | | yhat\_upper | FLOAT |  |  |  |
| 처리 내용 | | •수신 시스템에 맞추어 ds 의 TYPE을 DATE로 표준화하여 연계한다.•수신 시스템에 맞추어 yhat의 TYPE을 FLOAT로 표준화하여 연계한다.•수신 시스템에 맞추어 yhat\_lower의 TYPE을 FLOAT로 표준화하여 연계한다.•수신 시스템에 맞추어 yhat\_upper의 TYPE을 FLOAT로 표준화하여 연계한다. | | | | | | | | | | | |

파일



(2) 대상 시스템 및 송/수신 정보를 확인한다.

(3) 스크립트 로그 방식으로 연계 방식을 구현한다.

|  |
| --- |
| □ 데이터 호환 포맷 활용 예 1-테이블 목록 보기 |

ㅇ app.py

|  |
| --- |
| # importing flask  from flask import Flask, render\_template  # importing pandas module  import pandas as pd  app = Flask(\_\_name\_\_)  # reading the data in the csv file  df = pd.read\_csv('test/Prophet\_forecast.csv')  df.to\_csv('test/Prophet\_forecast.csv', index=None)  # route to html page - "table"  @app.route('/')  @app.route('/table')  def table():    # converting csv to html  data = pd.read\_csv('test/Prophet\_forecast.csv')  return render\_template('table.html', tables=[data.to\_html()], titles=[''])  if \_\_name\_\_ == "\_\_main\_\_":  app.run(host="localhost", port=int("5000")) |
| app.py |

ㅇ table.html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title> Table </title>  </head>  <body>  <div align="center">  <table>  <h1>  <!--Displaying the converted table-->  {% for table in tables %}  <h2>{{titles[loop.index]}}</h2>  {{ table|safe }}  {% endfor %}  </h1>  </table>  </div>  </body>  </html> |
| table.html |

ㅇ 테이블 목록 보기

(flask\_37) C:\DEV\flask\_works\flask\_webappspython app.py

\* Serving Flask app 'app' (lazy loading)

\* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

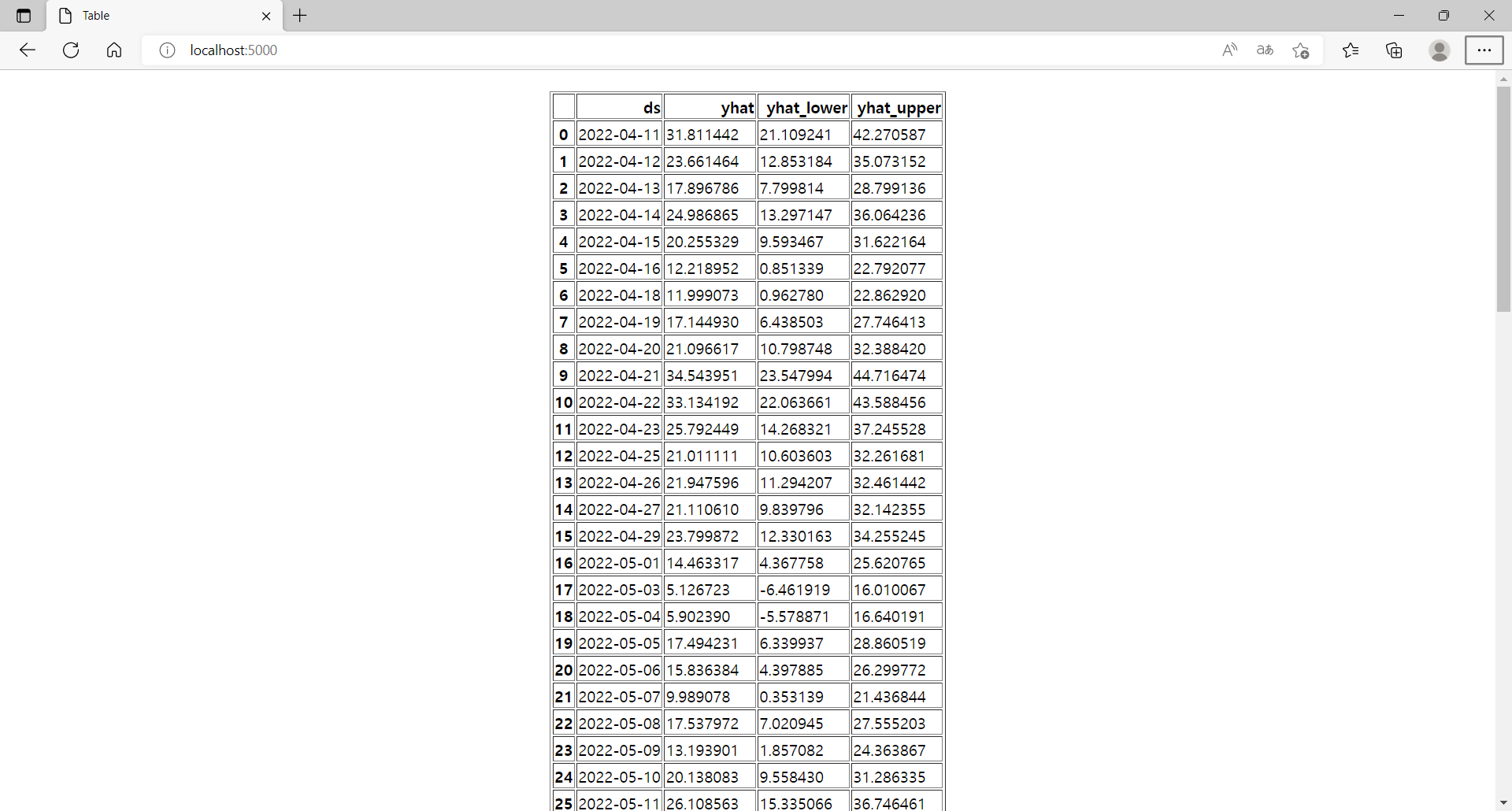
Use a production WSGI server instead.

\* Debug mode: off

\* Running on http://localhost:5000 (Press CTRL+C to quit)

127.0.0.1 - - [07/Jul/2022 16:33:18] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [07/Jul/2022 16:33:18] "GET /favicon.ico HTTP/1.1" 404 -



ㅇ 파일 구성

app.py (Flask run server 파일)

csv-to-sqlite.py (csv를 sqlite DB에 저장할)

db.sqlite(DB파일)

data.csv(시각화할 데이터가 들어있는 파일)

graph.html(html파일)

static 파일들(그래프를 구현해줄 JS 파일)

(flask\_37) C:\DEV\flask\_works\interfaceImpl>tree /f

폴더 PATH의 목록이다.

볼륨 일련 번호는 EA9B-192D이다.

C:.

│ app.py

│ app.py.bak

│ csv-to-sqlite.py

│ csv-to-sqlite.py.bak

│ interfaceImpl.sqlite

│ static.zip

│

├─static

│ exporting.js

│ highcharts-more.js

│ highstock.js

│ jquery-1.8.3.min.js

│

├─templates

│ graph.html

│ graph.html.bak

│

└─test

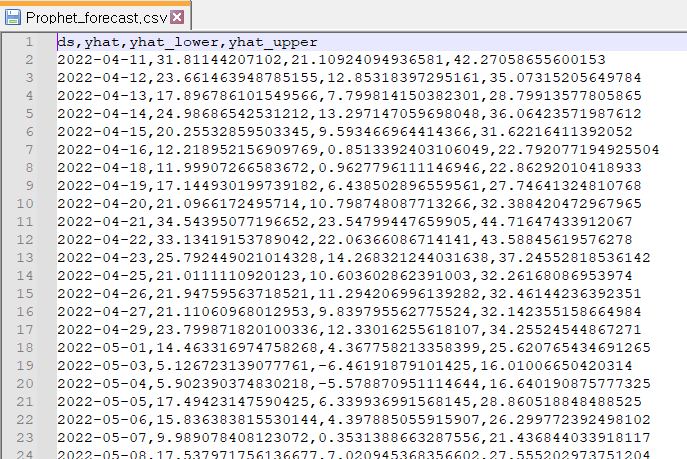
date\_result\_test.csv

Prophet\_forecast.csv

Prophet\_forecast.json

(flask\_37) C:\DEV\flask\_works\interfaceImpl>

CSV 파일



ㅇ CSV 데이터 sqlite DB에 저장

import csv, sqlite3

conn = sqlite3.connect("interfaceImpl.sqlite") # 저장할 DB파일 이름

curs = conn.cursor()

curs.execute("CREATE TABLE defective\_rate (ds DATE, yhat FLOAT, yhat\_lower FLOAT, yhat\_upper FLOAT)")

#TABLE : defective\_rate , 컬럼이름 : (ds , yhat, yhat\_lower, yhat\_upper)

reader = csv.reader(open('test/Prophet\_forecast.csv', 'r')) # CSV파일 읽기모드로 열기

for row in reader: #for 반복문을 통하여 DB에 작성

to\_db = [(row[0]), (row[1]), (row[2]), (row[3])]

curs.execute("INSERT INTO defective\_rate (ds, yhat, yhat\_lower, yhat\_upper) VALUES (?, ?, ?, ?);", to\_db)

conn.commit() #커밋 (쌓아둔 명령 실행)

conn.close()

(flask\_37) C:\DEV\flask\_works\interfaceImpl>python csv-to-sqlite.py

(flask\_37) C:\DEV\flask\_works\interfaceImpl>sqlite3 interfaceImpl.sqlite

SQLite version 3.38.5 2022-05-06 15:25:27

Enter ".help" for usage hints.

sqlite> .database

main: C:\DEV\flask\_works\interfaceImpl\interfaceImpl.sqlite r/w

sqlite> .tables

defective\_rate

sqlite> .schema defective\_rate

CREATE TABLE defective\_rate (ds DATE, yhat FLOAT, yhat\_lower FLOAT, yhat\_upper FLOAT);

sqlite> select \* from defective\_rate ;

ds|yhat|yhat\_lower|yhat\_upper

2022-04-11|31.81144207102|21.1092409493658|42.2705865560015

2022-04-12|23.6614639487852|12.8531839729516|35.0731520564978

2022-04-13|17.8967861015496|7.7998141503823|28.7991357780586

2022-04-14|24.9868654253121|13.2971470596981|36.0642357198761

2022-04-15|20.2553285950334|9.59346696441437|31.6221641139205

2022-04-16|12.2189521569098|0.851339240310605|22.7920771949255

2022-04-18|11.9990726658367|0.962779611114695|22.8629201041893

2022-04-19|17.1449301997392|6.43850289655956|27.7464132481077

sqlite> .quit

(flask\_37) C:\DEV\flask\_works\interfaceImpl>

ㅇ CSV 파일 읽어들여서 highcharts + Flask 그래프 구현

|  |
| --- |
| from flask import Flask, render\_template, request  import sqlite3  import json    app = Flask(\_\_name\_\_)      @app.route("/data.json")  def data():  connection = sqlite3.connect("interfaceImpl.sqlite")  cursor = connection.cursor()  #cursor.execute("SELECT 1000\*ds, yhat, yhat\_lower, yhat\_upper from defective\_rate")  cursor.execute("SELECT yhat from defective\_rate")  results = cursor.fetchall()    return json.dumps(results)    @app.route("/graph")  def graph():  return render\_template('graph.html')      if \_\_name\_\_ == '\_\_main\_\_':  app.run(debug=True, host='127.0.0.1', port=5000) |
| app.py |

|  |
| --- |
| <!DOCTYPE HTML>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>highcharts Example</title>  <script src="{{ url\_for('static', filename='jquery-1.8.3.min.js') }}"></script>  <script type="text/javascript">  $(document).ready(function () {  $.getJSON('http://127.0.0.1:5000/data.json', function (data) {    <!-- var reg = /[\{\}\[\]\/?.,;:|\)\*~`!^\-\_+<>@\#$%&\\\=\(\'\"]/gi; -->  var reg = /[\{\}\[\]\/yhat?;:|\)\*~`!^\-\_+<>@\#$%&\\\=\(\'\"]/gi;  var chk\_array = JSON.stringify(data);  var str=chk\_array.replace(reg,'');  str1=str.slice(1);  <!-- str2=str1.slice(0, -1); -->    console.log(str1);      // Create the chart  $('#container').highcharts('StockChart', {  rangeSelector : {  selected : 1  },  title : {  text : '불량률 추세'  },  series : [{  name : 'Value',  <!-- data : [str1], -->  data : [31.81144207102,23.661463948785155,17.896786101549566,24.98686542531212,20.25532859503345,12.218952156909769,11.99907266583672,17.144930199739182,21.0966172495714,34.54395077196652,33.13419153789042,25.792449021014328,21.0111110920123,21.94759563718521,21.11060968012953,23.799871820100336,14.463316974758268,5.126723139077761,5.902390374830218,17.49423147590425,15.836383815530144,9.989078408123072,17.537971756136677,13.193901298557128,20.13808297511555,26.10856284250992,41.75407365875388,42.55511928991415,37.15554754393476,42.78154761283799,33.85932703924625,33.375824944544675,28.97768861534213,31.334175970109825,16.06343127464909,7.934854942301022,23.06358048903182,54.41788115911973,78.43577445266543,106.81427051044648,128.16486098476344,166.10839729150555,210.9657079748968,244.36641591748528,290.6639013941311,325.61108292126823,360.3006746820074,382.84303524037,416.480866621724,451.2964288912426,468.83313634923365,493.5189963392199,501.34620360976,503.8103286517145,489.5821414810905,482.6114078750855,473.8185517127331,445.6964234790259,423.71014561641726,384.9499792119584,342.0413861159822,284.78614673797773,238.23474863165745,194.34685122208305,136.56298502267686,91.17635346976684,35.96053142711385,15.941989721660924,74.39587190461765,114.21284050637432,143.49820647260526,179.08056940853896,195.13572077422683,214.54900506938063,225.01140569592005,237.37519342133183,227.5688893696836,204.9149767291892,187.52771046879488,150.9029203203159,119.24501211581692,81.52742537021699,49.81339168350296,1.137738840299214,54.2058748901397],  tooltip: {  valueDecimals: 2  }  }]  });  });  });  </script>  </head>  <body>  <script src="{{ url\_for('static', filename='highstock.js') }}"></script>  <script src="{{ url\_for('static', filename='highcharts-more.js') }}"></script>  <script src="{{ url\_for('static', filename='exporting.js') }}"></script>  <div id="container" style="min-width: 310px; height: 400px; margin: 0 auto"></div>  </body>  </html> |
| graph.html |

ㅇ http://127.0.0.1:5000/graph

