T-EXPERT

『6과목 』-미니프로젝트 PBA 불량률 사전 예지 플랫폼 구현

2024.06.10-2024.06.21

Prepared by DaeKyeong Ki

Ph.D.

한국기술교육대학교 산학협력단

KOREATECH 한국기술교육대학교

미니프로젝트 lab :

PBA 불량률 사전 예지 플랫폼 구현

1. library
2. Data Ingestion, Data Acquisition
3. Machine Learning Algorithms을 위한 데이터 준비
4. 웹 서비스 플랫폼 구현
5. 인터페이스 명세서와 파일
6. 테이블 목록 보기
7. CSV 데이터 sqlite DB에 저장

학습목표

- 이 워크샵에서는 **Data Ingestion(수집), Data Acquisition(취득,획득)과 Machine Learning Algorithms**을 위한 데이터 준비를 하고, 학습과 학습 후 결과에 대해 활용할 수 있도록 저장을 할 수 있습니다.

Subsection 1

library

Library 확인 및 설치

○ fbprophet 설치

```
(tf37_cpu) C:\Users\k8s>conda install -c plotly plotly

(tf37_cpu) C:\Users\k8s>pip install pystan

(tf37_cpu) C:\Users\k8s>pip install --upgrade setuptools

(tf37_cpu) C:\Users\k8s>conda install -c conda-forge fbprophet

(tf37_cpu) C:\Users\k8s>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()
```

○ statsmodels 설치

```
(tf37_cpu) C:\Users\k8s>pip install statsmodels
```

○ nbformat 설치

```
(tf37_cpu) C:\Users\k8s>pip install nbformat
```

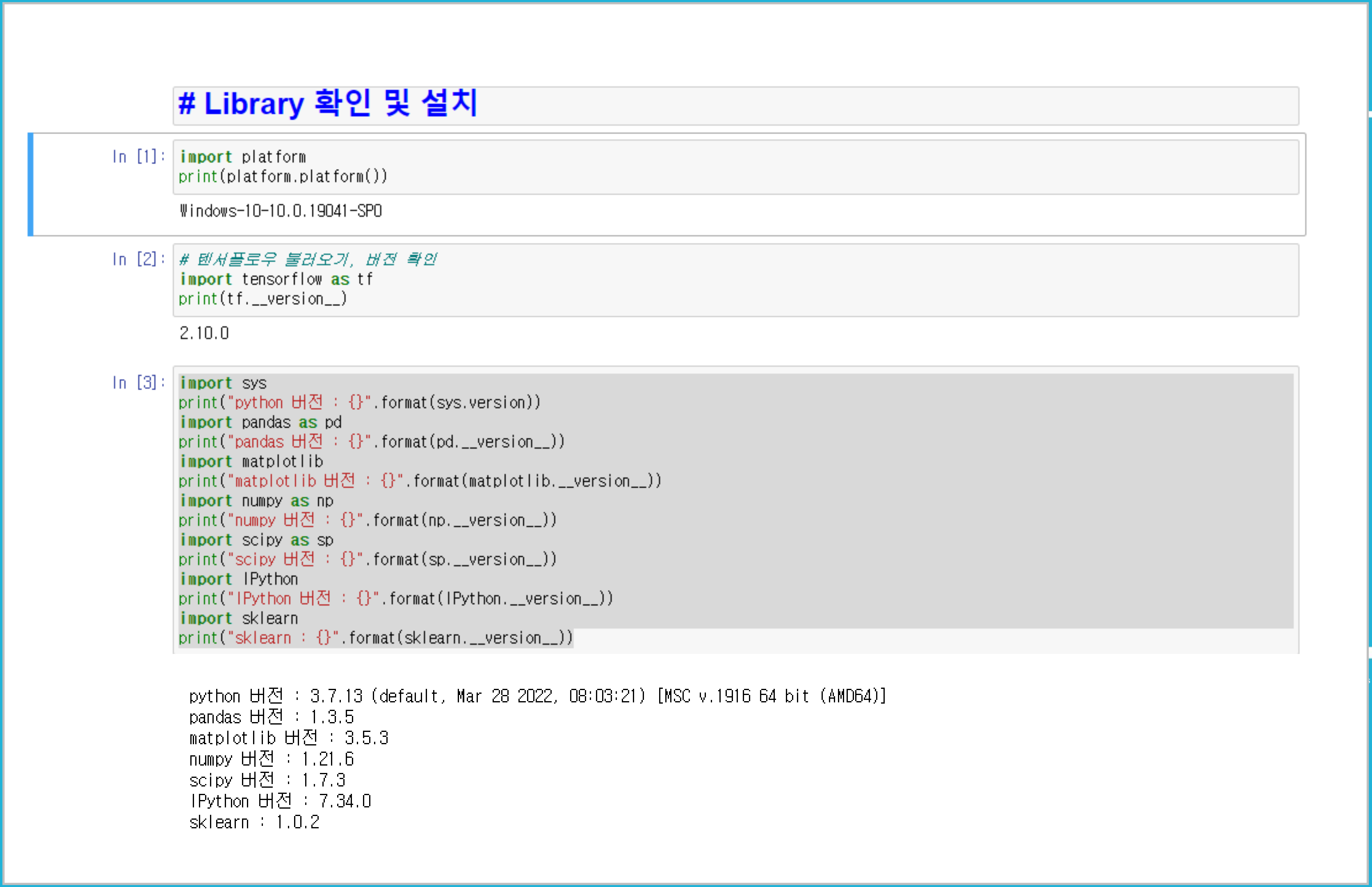
Unit 1

library

Library 확인 및 설치

```
import sys
print("python 버전 : {}".format(sys.version))
import pandas as pd
print("pandas 버전 : {}".format(pd.__version__))
import matplotlib
print("matplotlib 버전 : {}".format(matplotlib.__version__))
import numpy as np
print("numpy 버전 : {}".format(np.__version__))
import scipy as sp
print("scipy 버전 : {}".format(sp.__version__))
import IPython
print("IPython 버전 : {}".format(IPython.__version__))
import sklearn
print("sklearn : {}".format(sklearn.__version__))



----------------------------



pip install numpy scipy sklearn pandas matplotlib
pip install xlrd=1.2.0
pip install openpyxl
```

Library 확인 및 설치



Library 확인 및 설치

In [1]:
```
import platform
print(platform.platform())
```

```
Windows-10-10.0.19041-SP0
```

In [2]:
```
# 텐서플로우 불러오기, 버전 확인
import tensorflow as tf
print(tf.__version__)
```

```
2.10.0
```

In [3]:
```
import sys
print("python 버전 : {}".format(sys.version))
import pandas as pd
print("pandas 버전 : {}".format(pd.__version__))
import matplotlib
print("matplotlib 버전 : {}".format(matplotlib.__version__))
import numpy as np
print("numpy 버전 : {}".format(np.__version__))
import scipy as sp
print("scipy 버전 : {}".format(sp.__version__))
import IPython
print("IPython 버전 : {}".format(IPython.__version__))
import sklearn
print("sklearn : {}".format(sklearn.__version__))
```

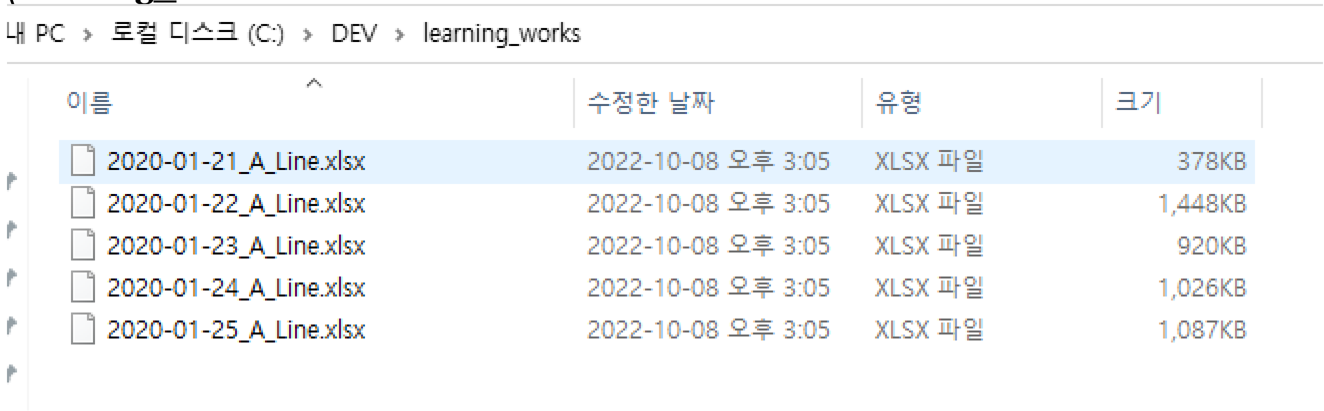
```
python 버전 : 3.7.13 (default, Mar 28 2022, 08:03:21) [MSC v.1916 64 bit (AMD64)]
pandas 버전 : 1.3.5
matplotlib 버전 : 3.5.3
numpy 버전 : 1.21.6
scipy 버전 : 1.7.3
IPython 버전 : 7.34.0
sklearn : 1.0.2
```

Unit 2

작업 경로

작업 경로 설정

```
Microsoft Windows [Version 10.0.19043.928]
(c) Microsoft Corporation. All rights reserved.

C:\DEV\learning_works>tree /f
폴더 PATH의 목록입니다.
볼륨 일련 번호는 BED0-C858입니다.
C:.
   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


에 하위 폴더가 없습니다.

C:\DEV\learning_works>
```



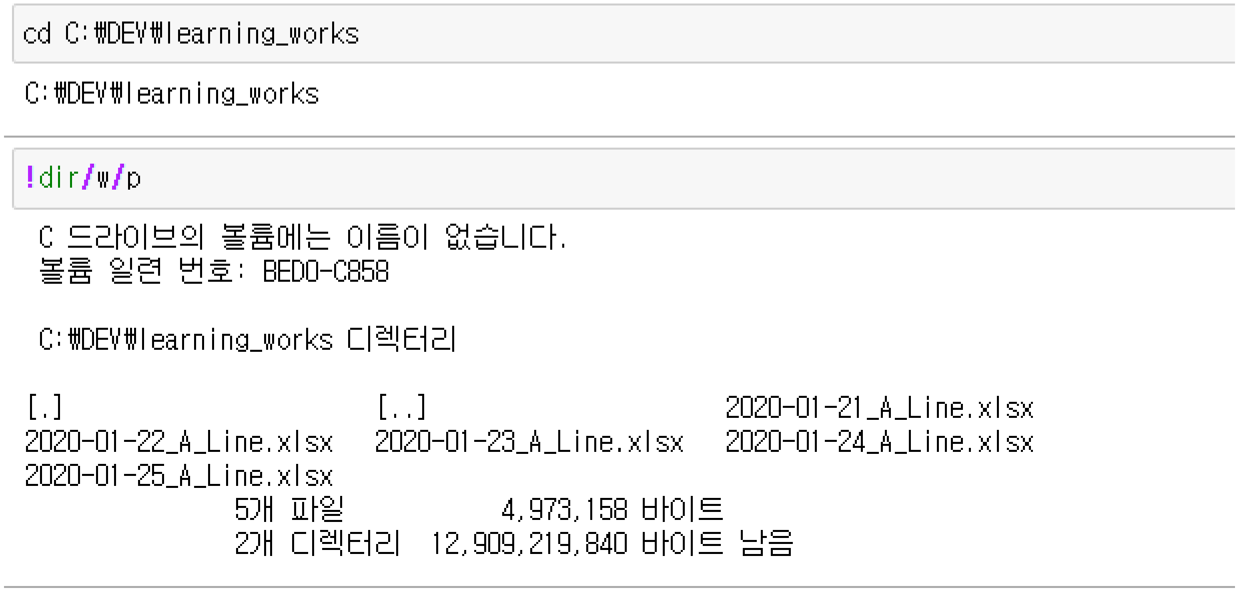
내 PC › 로컬 디스크 (C:) › DEV › learning_works

이름	수정한 날짜	유형	크기
2020-01-21_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	378KB
2020-01-22_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	1,448KB
2020-01-23_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	920KB
2020-01-24_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	1,026KB
2020-01-25_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	1,087KB

Unit 2

작업 경로

작업 경로 설정



```
cd C:₩DEV₩learning_works
```

```
C:₩DEV₩learning_works
```

```
!dir/w/p
```

```
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: BED0-C858

 C:₩DEV₩learning_works 디렉터리

[.]                        [..]                       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
               5개 파일           4,973,158 바이트
               2개 디렉터리  12,909,219,840 바이트 남음
```

Subsection 2

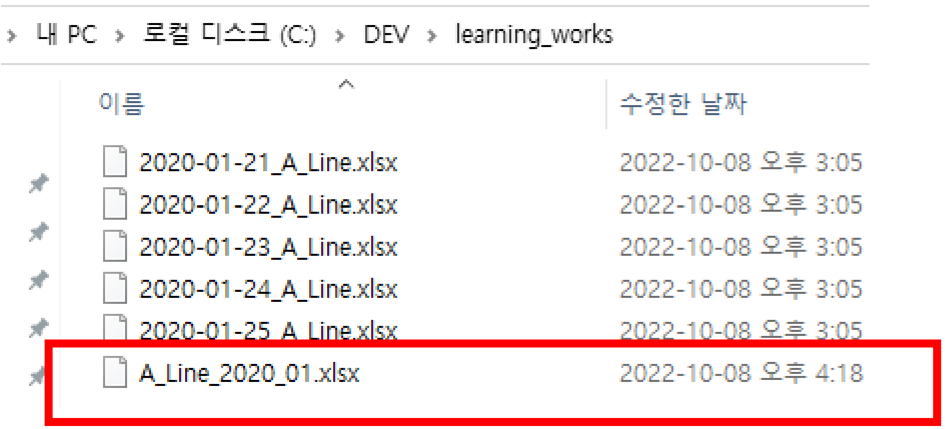
Data Ingestion, Data Acquisition

Unit 1

데이터 가져오기

가져 온 데이터 통합해 새 이름으로 저장하기

```
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()
```



› 내 PC › 로컬 디스크 (C:) › DEV › learning_works

이름	수정한 날짜
2020-01-21_A_Line.xlsx	2022-10-08 오후 3:05
2020-01-22_A_Line.xlsx	2022-10-08 오후 3:05
2020-01-23_A_Line.xlsx	2022-10-08 오후 3:05
2020-01-24_A_Line.xlsx	2022-10-08 오후 3:05
2020-01-25_A_Line.xlsx	2022-10-08 오후 3:05
A_Line_2020_01.xlsx	2022-10-08 오후 4:18

Unit 1

데이터 가져오기

통합한 데이터 다시 가져와 csv 파일로 저장하기

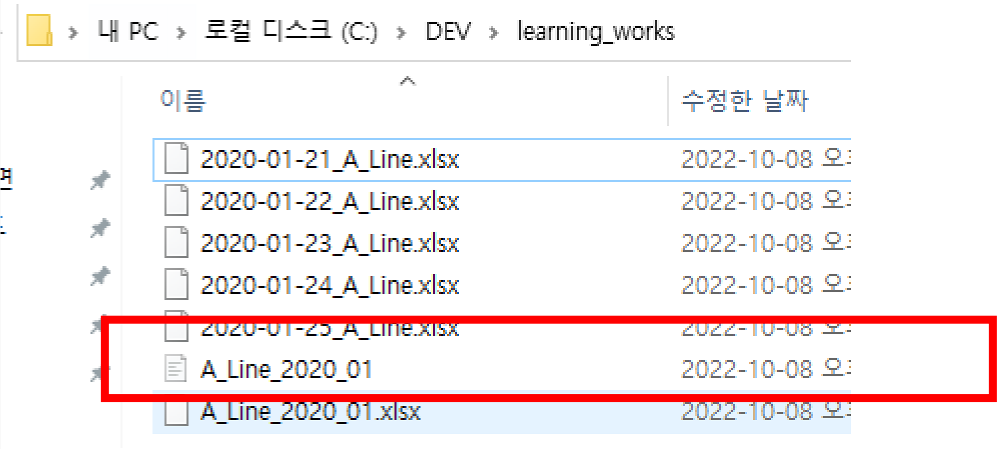
```
#불러올 파일의 경로를 filename 변수에 저장
filename = './A_Line_2020_01.xlsx'

import pandas as pd

#pandas read_excel로 불러오기
A_Line_2020_01 = pd.read_excel(filename)
A_Line_2020_01.head()
```

`

```
Date          Time                        Result            Periods           WRITING           BLE DEVICENAME      BLE MAC ADDRESS     FCTVER            MLBSERIAL
              FATPSERIAL                  DSNSERIAL         ETC               BLE RSSI          ATIVECURR           STANBYCURR          IR/Current        IR LED
              ACC_X                       ACC_Y             ACC_Z
0             2020-01-21                  19:06:34          OK                19.4              OK                  AbbeyFactoryTest    B010A059463E      Dec 15 2019
              GRS65006337020A24N00002                       WIP24211QUH100002 24211QUH100002    OK                  -33.0               2.1               2.9
              224.1                       1628              -8                -7                1151
1             2020-01-21                  19:06:34          OK                19.8              OK                  AbbeyFactoryTest    B010A0594640      Dec 15 2019
              GRS65006337020A24N00004                       WIP24211QUH100004 24211QUH100004    OK                  -34.0               2.1               2.4
              212.3                       741               9                 -8                1122
2             2020-01-21                  19:06:34          OK                19.6              OK                  AbbeyFactoryTest    B010A0594646      Dec 15 2019
              GRS65006337020A24N00005                       WIP24211QUH100005 24211QUH100005    OK                  -42.0               2.1               2.5
              217.4                       1133              -21               -45               968
3             2020-01-21                  19:06:34          OK                18.6              OK                  AbbeyFactoryTest    B010A0594642      Dec 15 2019
              GRS65006337020A24N00003                       WIP24211QUH100003 24211QUH100003    OK                  -33.0               2.0               1.9
              217.1                       1254              -5                -40               942
4    2020-01-21  19:06:34                 OK                18.9              OK                AbbeyFactoryTest    B010A0594648        Dec 15 2019
              GRS65006337020A24N00001                       WIP24211QUH100001 24211QUH100001    OK                  -37.0               2.0               2.5
              225.1                       2785              -35               -11               948

5    `
```

통합한 데이터 다시 가져와 csv 파일로 저장하기

```
A_Line_2020_01.to_csv('./A_Line_2020_01.csv', index=False)
```



› 내 PC › 로컬 디스크 (C:) › DEV › learning_works

이름	수정한 날짜
2020-01-21_A_Line.xlsx	2022-10-08 오
2020-01-22_A_Line.xlsx	2022-10-08 오
2020-01-23_A_Line.xlsx	2022-10-08 오
2020-01-24_A_Line.xlsx	2022-10-08 오
2020-01-25_A_Line.xlsx	2022-10-08 오
A_Line_2020_01	2022-10-08 오
A_Line_2020_01.xlsx	2022-10-08 오

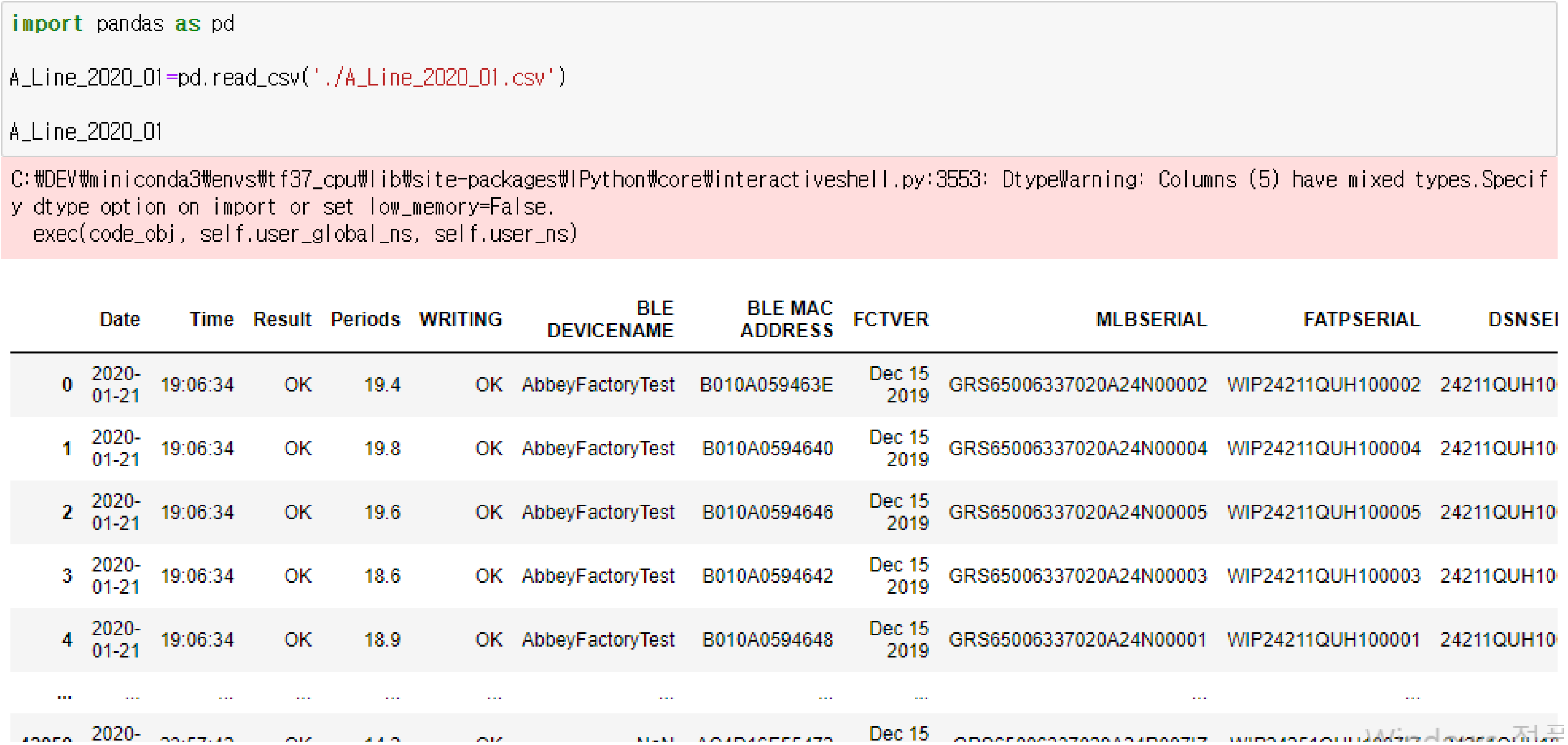
Unit 1

데이터 가져오기

통합한 데이터 다시 가져와 csv 파일로 저장하기

import pandas as pd

A_Line_2020_01=pd.read_csv('./A_Line_2020_01.csv')



```
import pandas as pd

A_Line_2020_01=pd.read_csv('./A_Line_2020_01.csv')

A_Line_2020_01
```

```
C:₩DEV₩miniconda3₩envs₩tf37_cpu₩lib₩site-packages₩IPython₩core₩interactiveshell.py:3553: DtypeWarning: Columns (5) have mixed types.Specif
y dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

	Date	Time	Result	Periods	WRITING	BLE DEVICENAME	BLE MAC ADDRESS	FCTVER	MLBSERIAL	FATPSERIAL	DSNSEI
0	2020-01-21	19:06:34	OK	19.4	OK	AbbeyFactoryTest	B010A059463E	Dec 15 2019	GRS65006337020A24N00002	WIP24211QUH100002	24211QUH10
1	2020-01-21	19:06:34	OK	19.8	OK	AbbeyFactoryTest	B010A0594640	Dec 15 2019	GRS65006337020A24N00004	WIP24211QUH100004	24211QUH10
2	2020-01-21	19:06:34	OK	19.6	OK	AbbeyFactoryTest	B010A0594646	Dec 15 2019	GRS65006337020A24N00005	WIP24211QUH100005	24211QUH10
3	2020-01-21	19:06:34	OK	18.6	OK	AbbeyFactoryTest	B010A0594642	Dec 15 2019	GRS65006337020A24N00003	WIP24211QUH100003	24211QUH10
4	2020-01-21	19:06:34	OK	18.9	OK	AbbeyFactoryTest	B010A0594648	Dec 15 2019	GRS65006337020A24N00001	WIP24211QUH100001	24211QUH10
...	...	...	...	...	...	...	...	...	...	...	
[illegible]	2020-	[illegible]	OK	[illegible]	OK	NaN	[illegible]	Dec 15	[illegible]	[illegible]	[illegible]

Subsection 3

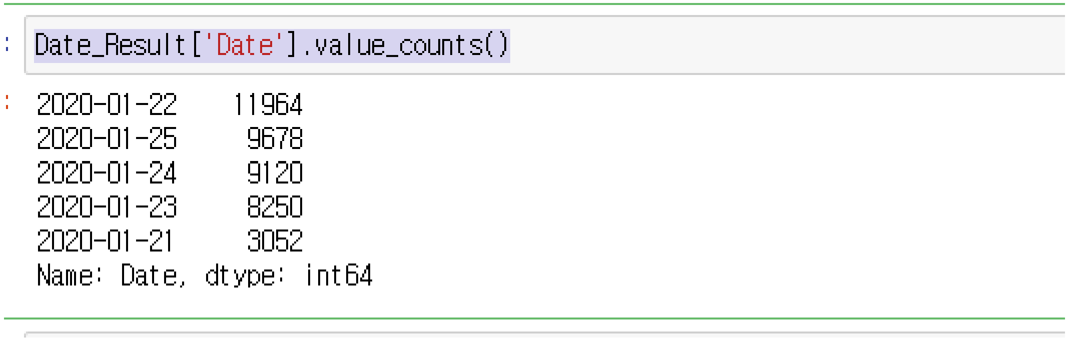
Machine Learning Algorithms을 위한 데이터 준비

특징 가공(Feature engineering)

Unit 1

Text와 Categorical 속성 처리

```
Result=pd.get_dummies(A_Line_2020_01["Result"])
Result.head(10)
data=pd.concat([A_Line_2020_01,Result],axis=1)
data.head()
Date_Result=data[['Date','FAIL']]
Date_Result.head()
Date_Result['Date'].value_counts()
```



```
Date_Result['Date'].value_counts()
```

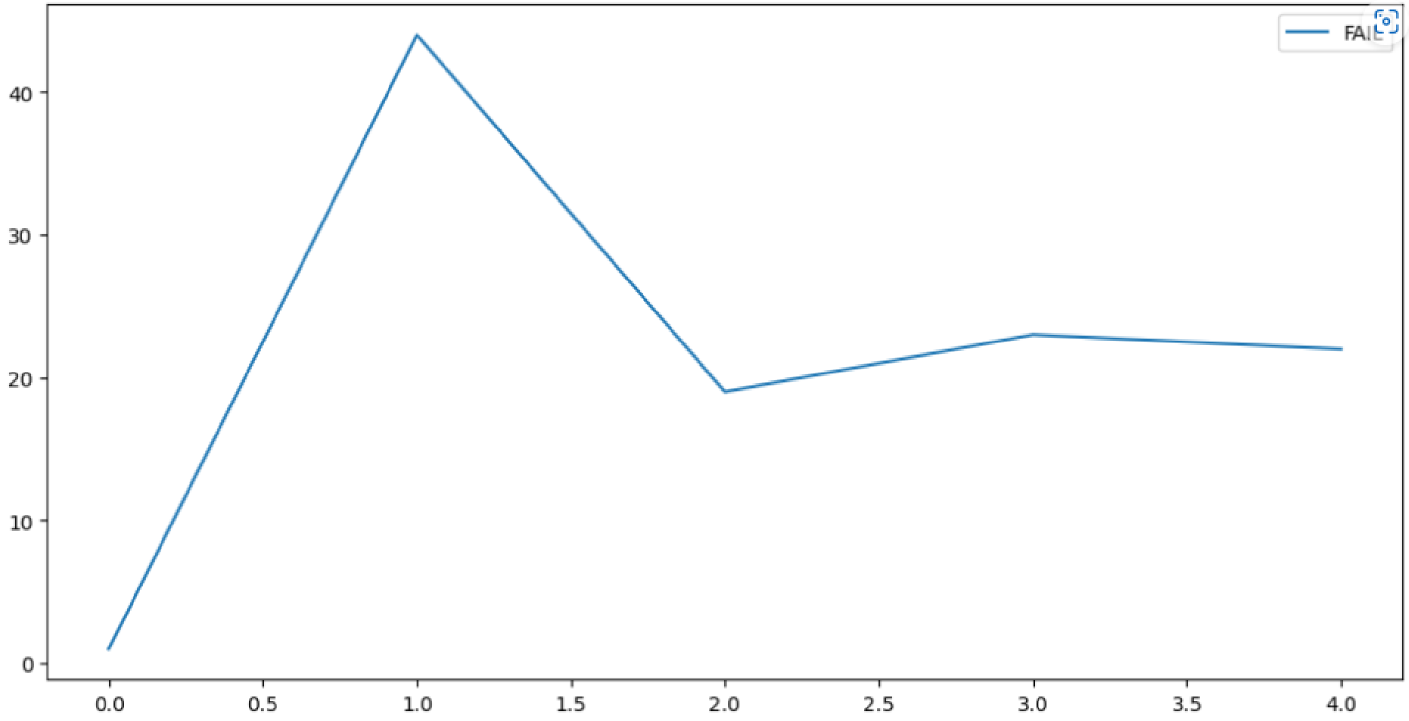
```
2020-01-22    11964
2020-01-25     9678
2020-01-24     9120
2020-01-23     8250
2020-01-21     3052
Name: Date, dtype: int64
```

Unit 1

특징 가공(Feature engineering)

Data Cleaning and Munging

```
Date_Result_gs=Date_Result.groupby('Date')['FAIL'].agg(**{'FAIL':'sum'}).reset_index()
Date_Result_gs
import matplotlib.pyplot as plt

Date_Result_gs.plot(figsize=(12,6))
plt.show()
```



Subsection 4

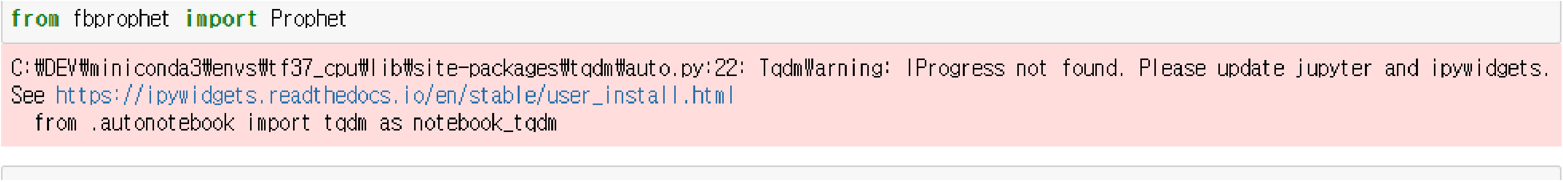
학습 솔루션 개발

Unit 1

Facebook Prophet

라이브러리 확인

from fbprophet import Prophet



```
from fbprophet import Prophet
```

```
C:₩DEV₩miniconda3₩envs₩tf37_cpu₩lib₩site-packages₩tqdm₩auto.py:22: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

Unit 1

Facebook Prophet 학습 진행

```
import plotly.graph_objs as go
import plotly.offline as py
from fbprophet import Prophet
from fbprophet.plot import plot_plotly, add_changepoints_to_plot

df_prophet = Date_Result_gs.rename(columns={
   'Date': 'ds',
   'FAIL': 'y'
})

df_prophet.tail()

m = Prophet(
   changepoint_prior_scale=0.5,
   changepoint_range=0.95,
   yearly_seasonality=False,
   weekly_seasonality=True,
   daily_seasonality=True,
   seasonality_mode='additive'
)

m.fit(df_prophet)

future = m.make_future_dataframe(periods=7)
forecast = m.predict(future)
```

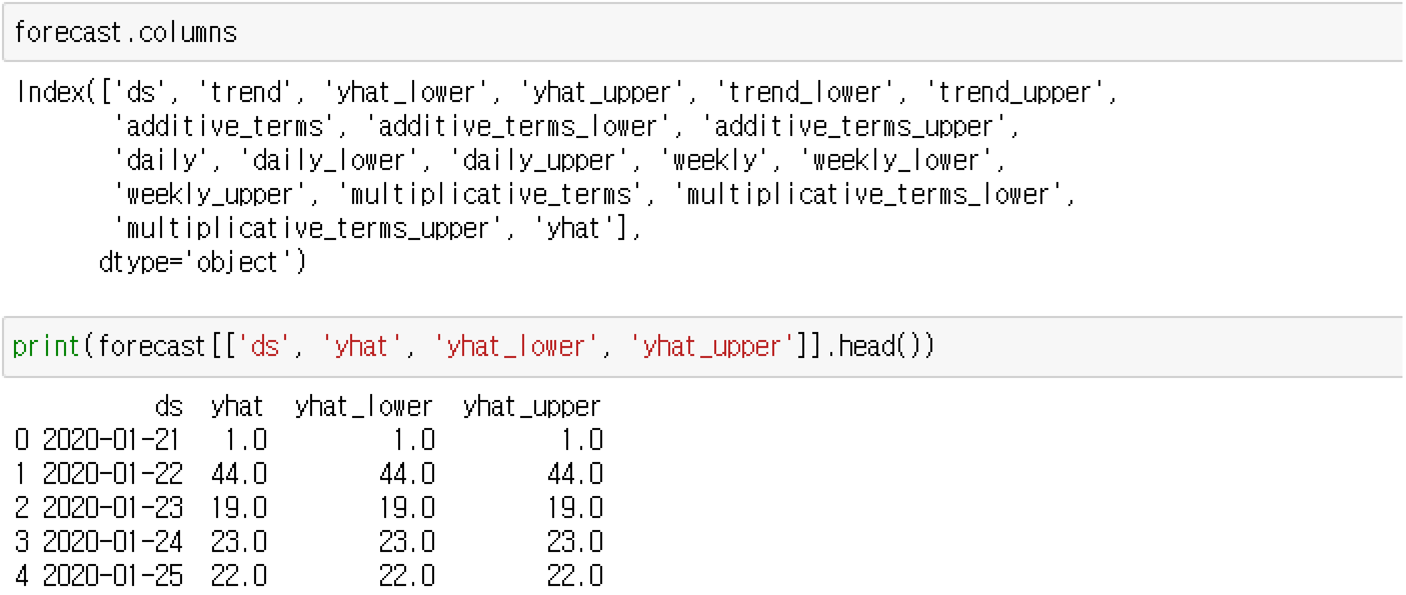
Facebook Prophet 학습 진행

```
import plotly.graph_objs as go
import plotly.offline as py
from fbprophet import Prophet
from fbprophet.plot import plot_plotly, add_changepoints_to_plot

df_prophet = Date_Result_gs.rename(columns={
   'Date': 'ds',
   'FAIL': 'y'
})

df_prophet.tail()

m = Prophet(
   changepoint_prior_scale=0.5,
   changepoint_range=0.95,
   yearly_seasonality=False,
   weekly_seasonality=True,
   daily_seasonality=True,
   seasonality_mode='additive'
)

m.fit(df_prophet)

future = m.make_future_dataframe(periods=7)
forecast = m.predict(future)
```

Facebook Prophet

Unit 1

Facebook Prophet 학습 결과



```
forecast.columns
```

```
Index(['ds', 'trend', 'yhat_lower', 'yhat_upper', 'trend_lower', 'trend_upper',
       'additive_terms', 'additive_terms_lower', 'additive_terms_upper',
       'daily', 'daily_lower', 'daily_upper', 'weekly', 'weekly_lower',
       'weekly_upper', 'multiplicative_terms', 'multiplicative_terms_lower',
       'multiplicative_terms_upper', 'yhat'],
      dtype='object')
```

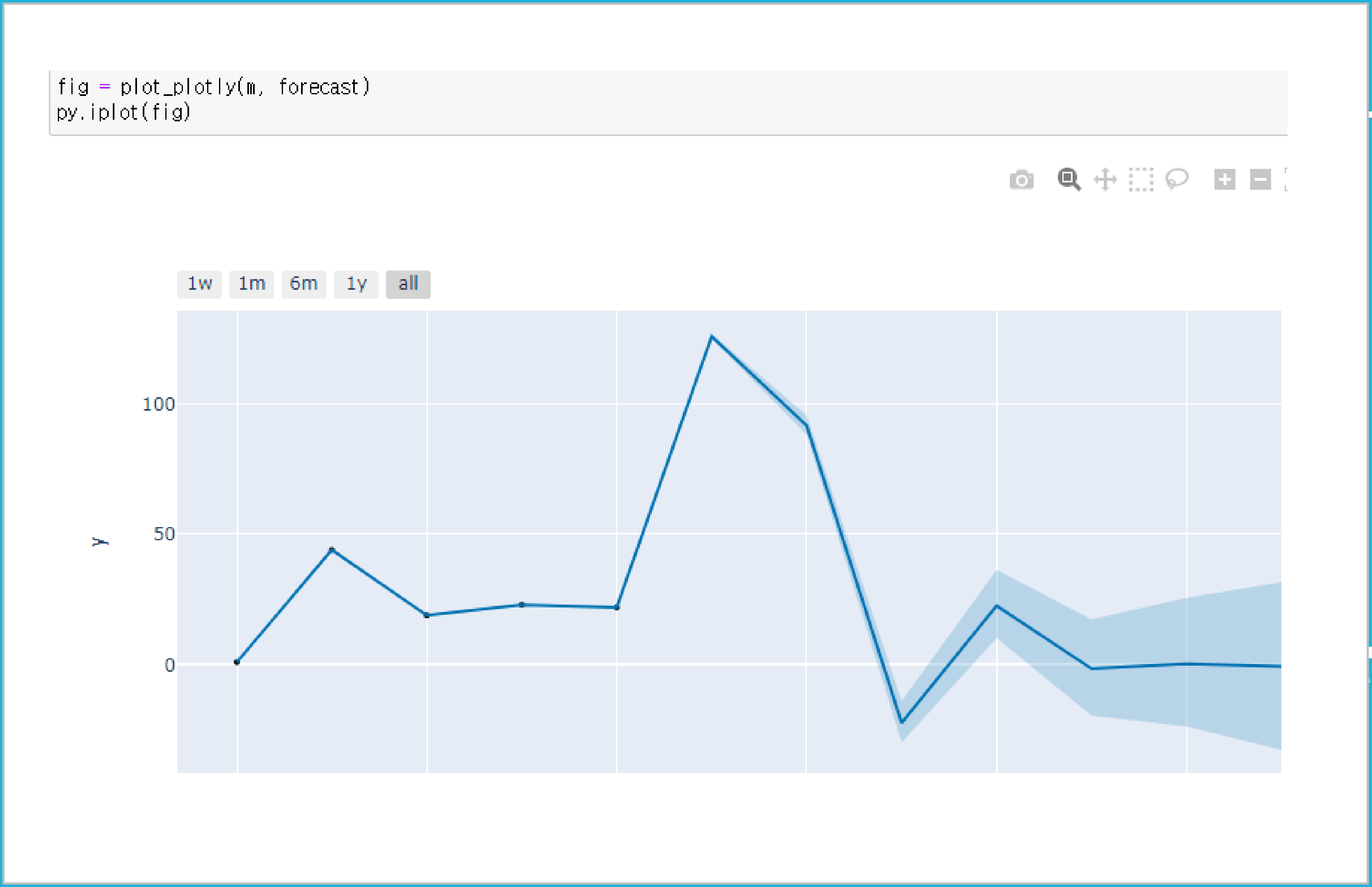
```
print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].head())
```

```
          ds  yhat  yhat_lower  yhat_upper
0 2020-01-21   1.0         1.0         1.0
1 2020-01-22  44.0        44.0        44.0
2 2020-01-23  19.0        19.0        19.0
3 2020-01-24  23.0        23.0        23.0
4 2020-01-25  22.0        22.0        22.0
```

Unit 1

Facebook Prophet

Facebook Prophet 학습 결과



```
fig = plot_plotly(m, forecast)
py.iplot(fig)
```

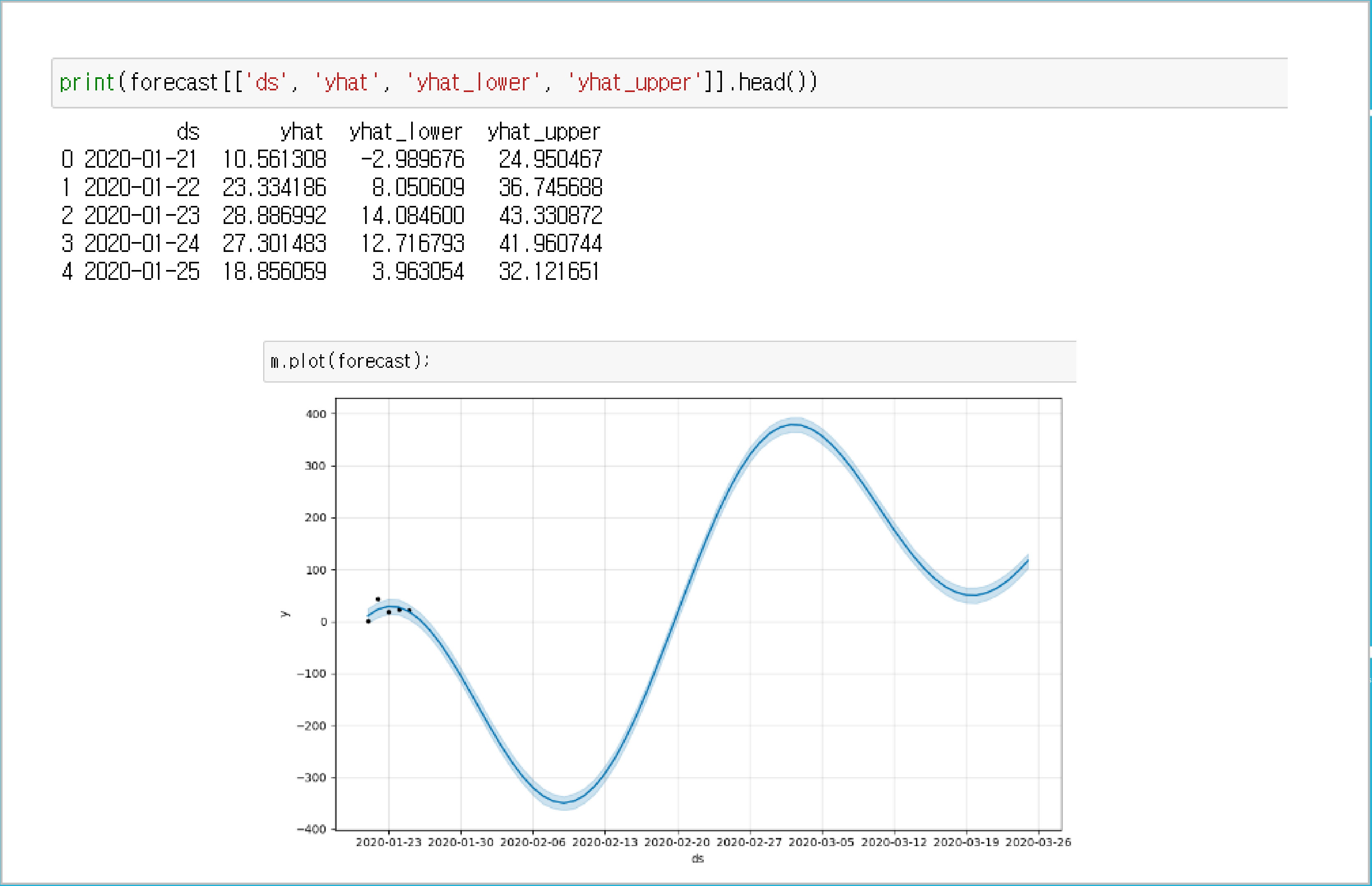
Prophet 모듈을 이용한 예측

```
df_prophet = Date_Result_gs.rename(columns={
   'Date': 'ds',
   'FAIL': 'y'
})

df_prophet.tail()

m=Prophet(yearly_seasonality=True)  #주기성이 연단위라고 설정
m.fit(df_prophet);

future=m.make_future_dataframe(periods=60)
forecast=m.predict(future)                    #60일간의 데이터 예측
```

Unit 1

Facebook Prophet

Prophet 모듈을 이용한 예측 결과



```
print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].head())
```

```
          ds       yhat  yhat_lower  yhat_upper
0 2020-01-21  10.561308   -2.989676   24.950467
1 2020-01-22  23.334186    8.050609   36.745688
2 2020-01-23  28.886992   14.084600   43.330872
3 2020-01-24  27.301483   12.716793   41.960744
4 2020-01-25  18.856059    3.963054   32.121651
```

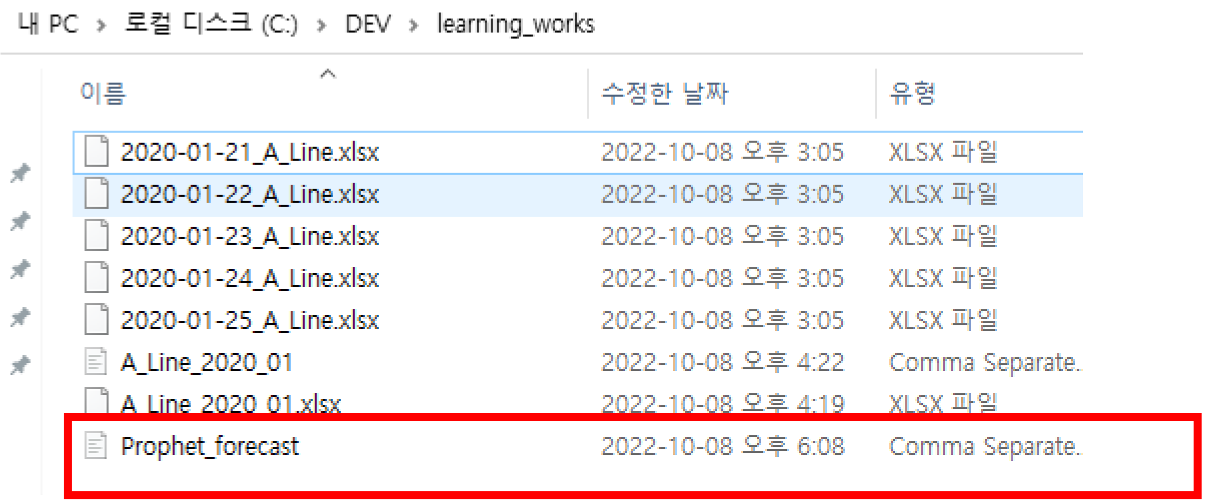
```
m.plot(forecast);
```

Unit 1

Facebook Prophet

Prophet 모듈을 이용한 예측 결과 저장

```
Prophet_forecast=forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']]
Prophet_forecast.to_csv("./Prophet_forecast.csv",index=False)
```

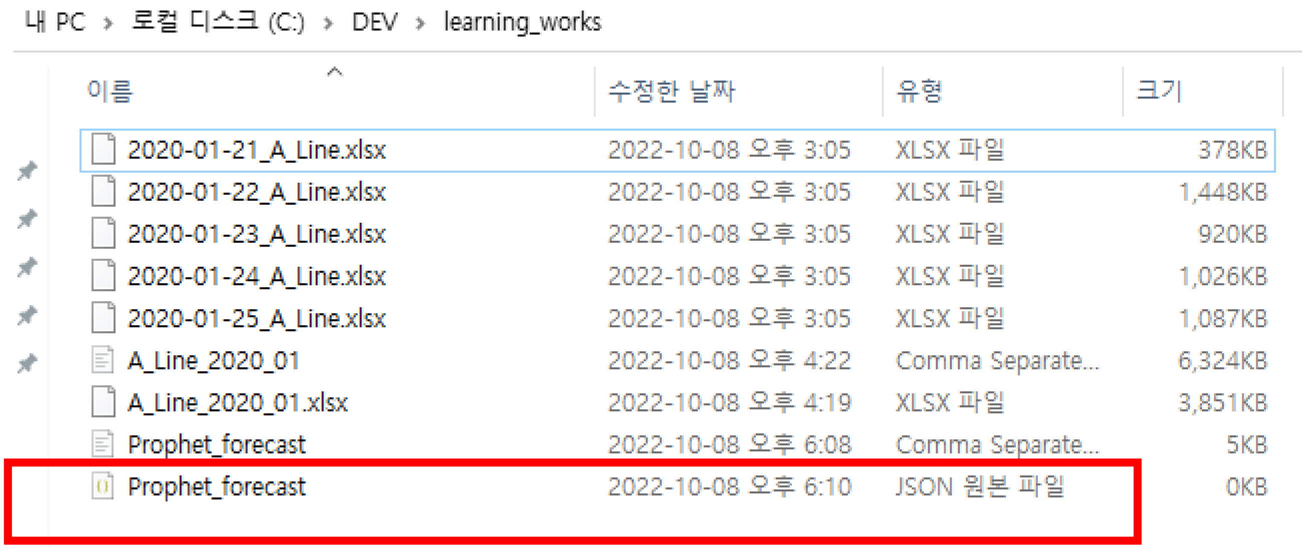


내 PC › 로컬 디스크 (C:) › DEV › learning_works

이름	수정한 날짜	유형
2020-01-21_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
2020-01-22_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
2020-01-23_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
2020-01-24_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
2020-01-25_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
A_Line_2020_01	2022-10-08 오후 4:22	Comma Separate.
A_Line_2020_01.xlsx	2022-10-08 오후 4:19	XLSX 파일
Prophet_forecast	2022-10-08 오후 6:08	Comma Separate.

Prophet 모듈을 이용한 예측 결과 저장

```
import csv
import json

csvfile=open('./Prophet_forecast.csv','r')
jsonfile=open('./Prophet_forecast.json','w')

fieldnames=("Date","FAIL")
reader=csv.DictReader(csvfile, fieldnames)

out=json.dumps([row for row in reader])
jsonfile.write(out)
```



내 PC › 로컬 디스크 (C:) › DEV › learning_works

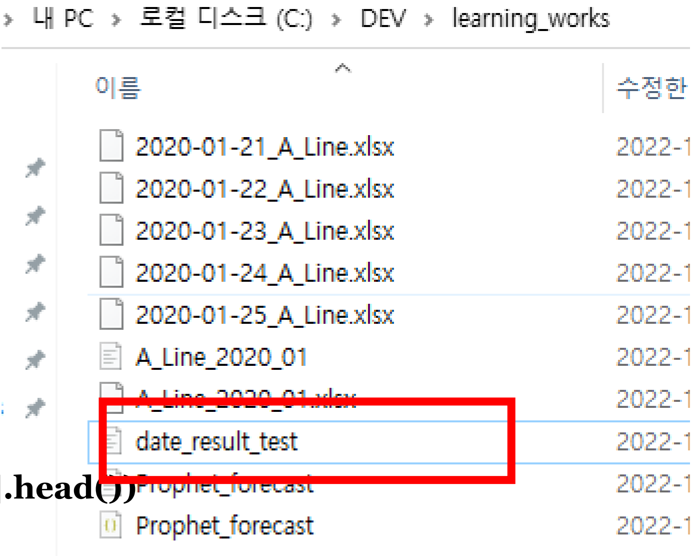
이름	수정한 날짜	유형	크기
2020-01-21_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	378KB
2020-01-22_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	1,448KB
2020-01-23_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	920KB
2020-01-24_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	1,026KB
2020-01-25_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일	1,087KB
A_Line_2020_01	2022-10-08 오후 4:22	Comma Separate...	6,324KB
A_Line_2020_01.xlsx	2022-10-08 오후 4:19	XLSX 파일	3,851KB
Prophet_forecast	2022-10-08 오후 6:08	Comma Separate...	5KB
Prophet_forecast	2022-10-08 오후 6:10	JSON 원본 파일	0KB

Facebook Prophet

Unit 1

학습된 모델 저장하기

```
pip install joblib



import pickle
import joblib

PBA_model = pickle.dumps(m)

m_from_pickle = pickle.loads(PBA_model)

test_data=pd.read_csv('./date_result_test.csv')

forecast=m_from_pickle.predict(test_data)

print("예측: {}".format(forecast))
print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].head())
```



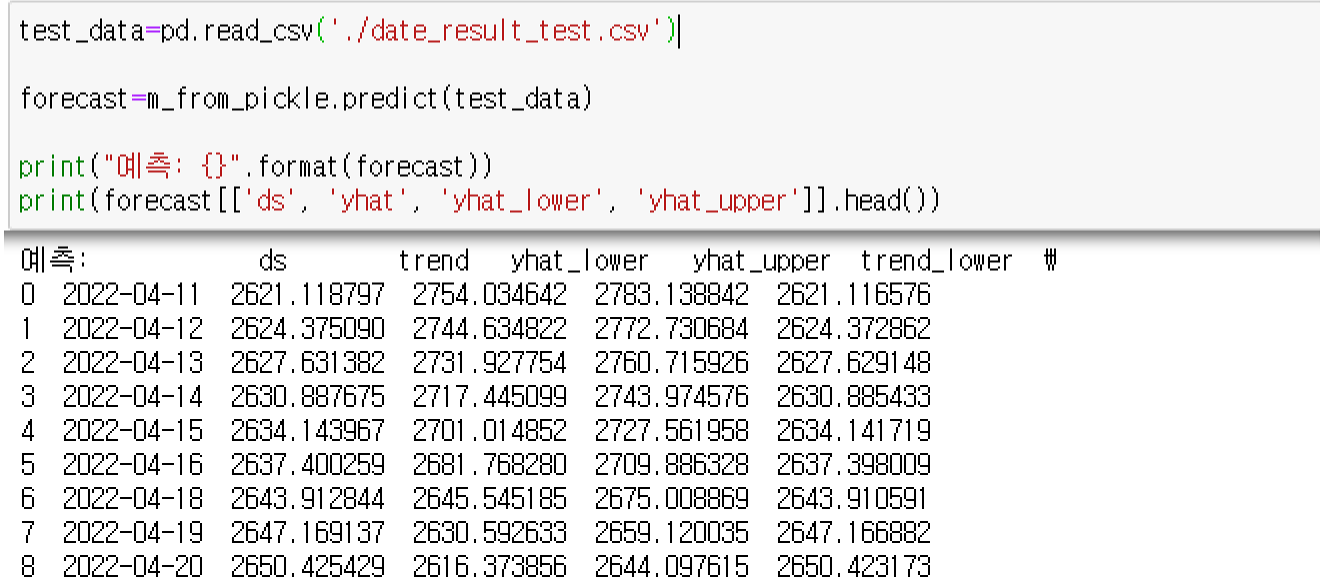
› 내 PC › 로컬 디스크 (C:) › DEV › learning_works

이름	수정한
2020-01-21_A_Line.xlsx	2022-1
2020-01-22_A_Line.xlsx	2022-1
2020-01-23_A_Line.xlsx	2022-1
2020-01-24_A_Line.xlsx	2022-1
2020-01-25_A_Line.xlsx	2022-1
A_Line_2020_01	2022-1
A_Line_2020_01.xlsx	2022-1
date_result_test	2022-1
Prophet_forecast	2022-1
Prophet_forecast	2022-1

Facebook Prophet

Unit 1

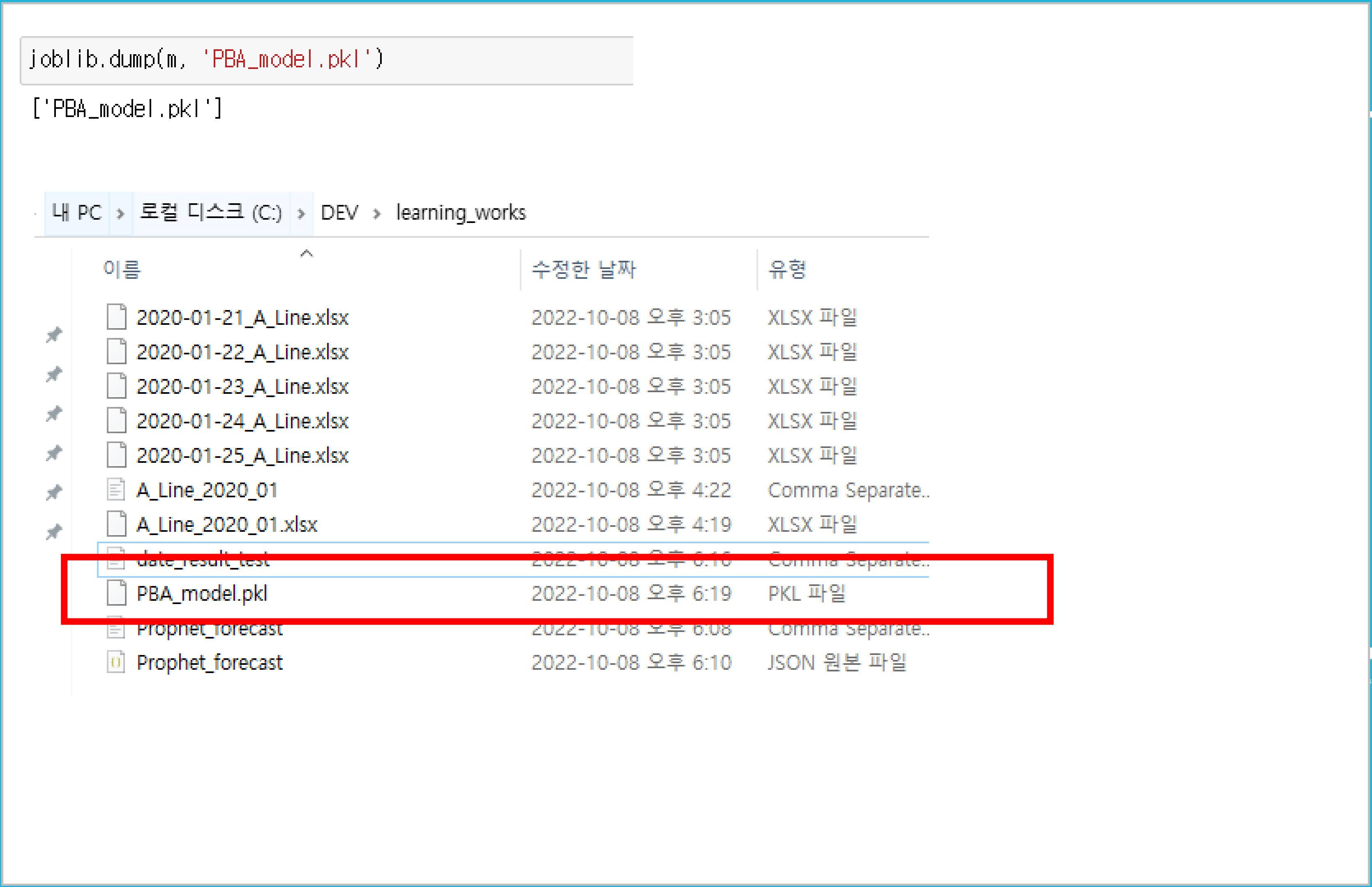
학습된 모델 저장하기



```
test_data=pd.read_csv('./date_result_test.csv')

forecast=m_from_pickle.predict(test_data)

print("예측: {}".format(forecast))
print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].head())
```

```
예측:            ds        trend   yhat_lower   yhat_upper  trend_lower  ₩
0  2022-04-11  2621.118797  2754.034642  2783.138842  2621.116576
1  2022-04-12  2624.375090  2744.634822  2772.730684  2624.372862
2  2022-04-13  2627.631382  2731.927754  2760.715926  2627.629148
3  2022-04-14  2630.887675  2717.445099  2743.974576  2630.885433
4  2022-04-15  2634.143967  2701.014852  2727.561958  2634.141719
5  2022-04-16  2637.400259  2681.768280  2709.886328  2637.398009
6  2022-04-18  2643.912844  2645.545185  2675.008869  2643.910591
7  2022-04-19  2647.169137  2630.592633  2659.120035  2647.166882
8  2022-04-20  2650.425429  2616.373856  2644.097615  2650.423173
```

학습된 모델 파일에 저장



```
joblib.dump(m, 'PBA_model.pkl')
```

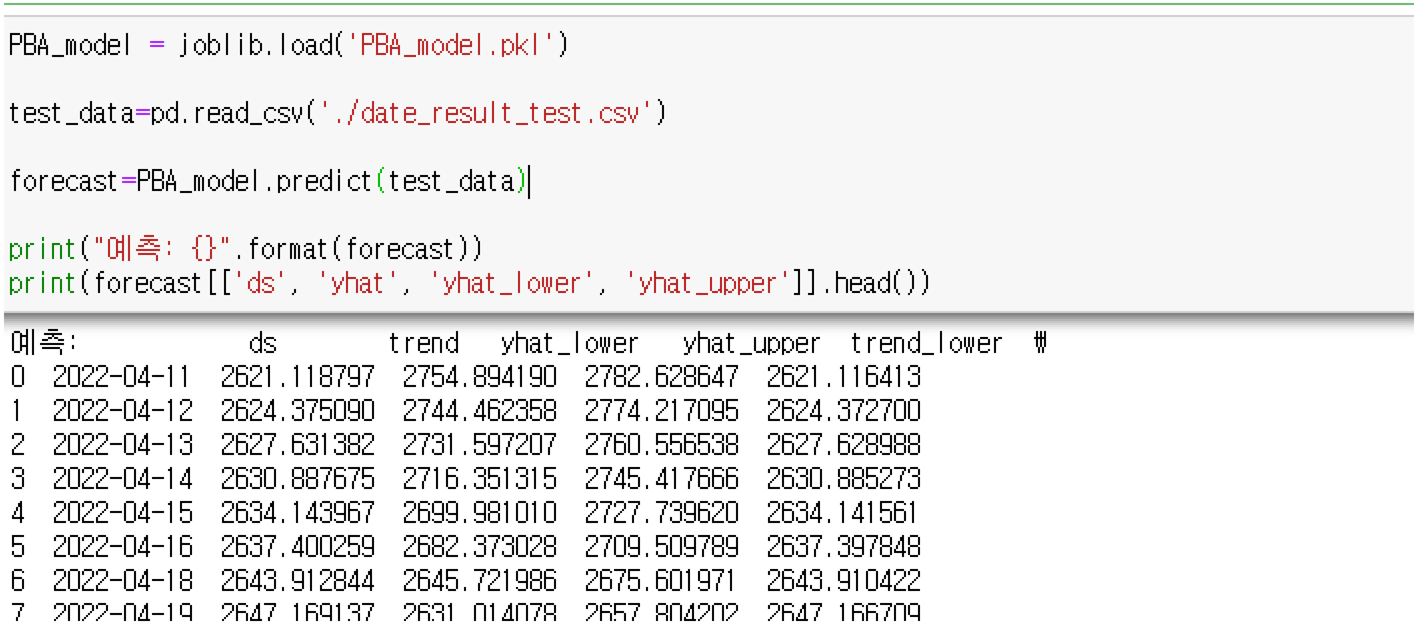
```
['PBA_model.pkl']
```

이름	수정한 날짜	유형
2020-01-21_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
2020-01-22_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
2020-01-23_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
2020-01-24_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
2020-01-25_A_Line.xlsx	2022-10-08 오후 3:05	XLSX 파일
A_Line_2020_01	2022-10-08 오후 4:22	Comma Separate..
A_Line_2020_01.xlsx	2022-10-08 오후 4:19	XLSX 파일
date_result_test	2022-10-08 오후 6:16	Comma Separate..
PBA_model.pkl	2022-10-08 오후 6:19	PKL 파일
Prophet_forecast	2022-10-08 오후 6:08	Comma Separate..
Prophet_forecast	2022-10-08 오후 6:10	JSON 원본 파일

Unit 1

Facebook Prophet

저장한 파일을 불러와서 predict



```
PBA_model = joblib.load('PBA_model.pkl')

test_data=pd.read_csv('./date_result_test.csv')

forecast=PBA_model.predict(test_data)

print("예측: {}".format(forecast))
print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].head())
```

```
예측:           ds        trend    yhat_lower    yhat_upper  trend_lower  ₩
0  2022-04-11  2621.118797  2754.894190  2782.628647  2621.116413
1  2022-04-12  2624.375090  2744.462358  2774.217095  2624.372700
2  2022-04-13  2627.631382  2731.597207  2760.556538  2627.628988
3  2022-04-14  2630.887675  2716.351315  2745.417666  2630.885273
4  2022-04-15  2634.143967  2699.981010  2727.739620  2634.141561
5  2022-04-16  2637.400259  2682.373028  2709.509789  2637.397848
6  2022-04-18  2643.912844  2645.721986  2675.601971  2643.910422
7  2022-04-19  2647.169137  2631.014078  2657.804202  2647.166709
```

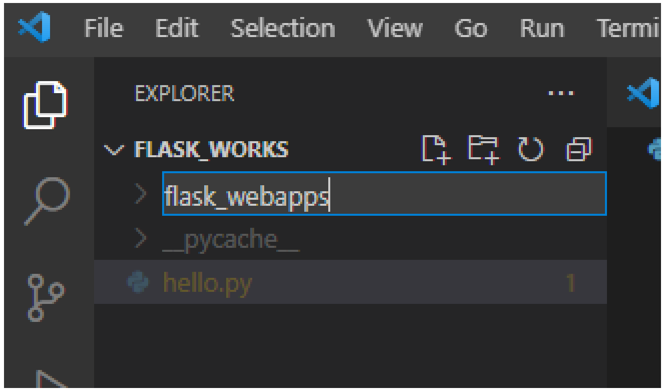
Subsection 5

flask_webapps

Unit 1

작업폴더 생성

flask_webapps



flask_webapps

```
(flask_37) C:\DEV\flask_works\flask_webapps>tree /f
폴더 PATH의 목록입니다.
볼륨 일련 번호는 BED0-C858입니다.
C:.
│  main.py
│
├─model
│      PBA_model.pkl
│
├─static
│      style.css
│
├─templates
│      index.html
│
└─test
        date_result_test.csv
        Prophet_forecast.csv
        Prophet_forecast.json


(flask_37) C:\DEV\flask_works\flask_webapps>
```

Unit 1

library

Library 확인 및 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()

(flask_37) C:\DEV\flask_works>conda install -c anaconda joblib
```

Unit 1

library

Library 확인 및 설치

```
(flask_37) C:\DEV\flask_works\flask_webapps>pip install -U scikit-learn==0.21.3
…

(flask_37) C:\DEV\flask_works\flask_webapps>pip uninstall numpy

(flask_37) C:\DEV\flask_works\flask_webapps>pip install numpy==1.19.5

(flask_37) C:\DEV\flask_works\flask_webapps>
pip install numpy --ignore-installed numpy==1.19.5

*** numpy 재 설치 에러날 때
(flask_37) C:\DEV\flask_works\flask_webapps>pip install --force-reinstall --no-deps
numpy==1.21.6

(flask_37) C:\DEV\flask_works\flask_webapps>pip install numpy --ignore-installed
numpy==1.19.5

  (flask_37) C:\DEV\flask_works\flask_webapps>pip uninstall scipy

(flask_37) C:\DEV\flask_works\flask_webapps>pip install scipy==1.1.0

(flask_37) C:\DEV\flask_webapps\Flask-ML_DL-Example>


*** scipy 재 설치 에러날 때
(flask_37) C:\DEV\flask_works\flask_webapps>pip install scipy --ignore-installed scipy==1.1.0
```

Unit 2

작성 코드

main.py

main.py

```python
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')
```

Unit 2

작성 코드

Templates / index.html

```
<html>

<head>
  <title>GE_PBA_B_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>RemoteSolution 2022-04-19_GE_PBA_B_검사서 Predictor</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>
```

작성 코드

Unit 2

Static / style.css

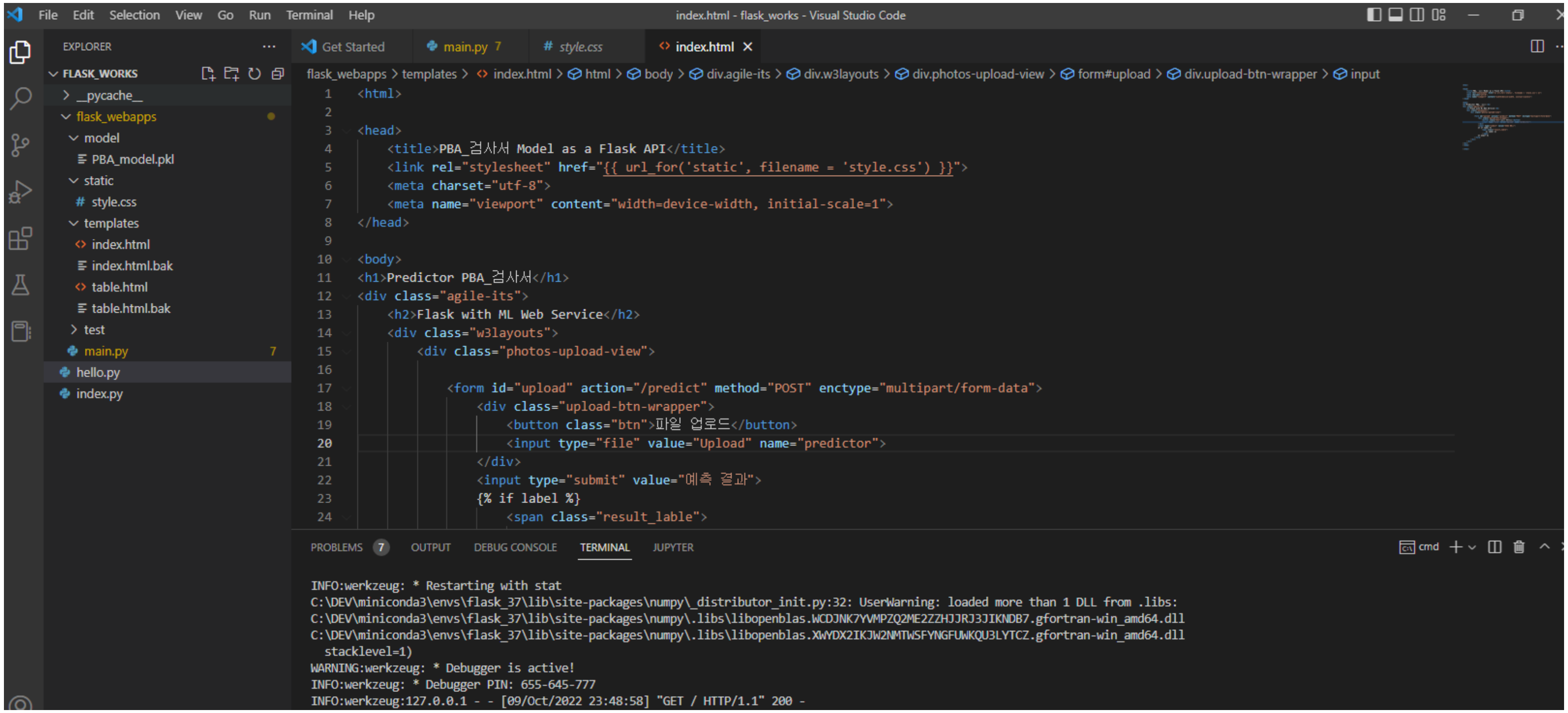
```
/*-- Reset-Code --*/
   html,body,div,span,applet,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,a,abbr,acronym,address,big,cite,code,del,dfn,em,img,ins,kbd,q,s,samp,small,strike,strong,sub,sup,tt,var,b,
u,i,dl,dt,dd,ol,nav ul,nav
li,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,article,aside,canvas,details,embed,figure,figcaption,footer,header,hgroup,menu,nav,output,ruby,section,summary,ti
me,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;

}...
```

생략

Unit 3

실행

CLI 명령어



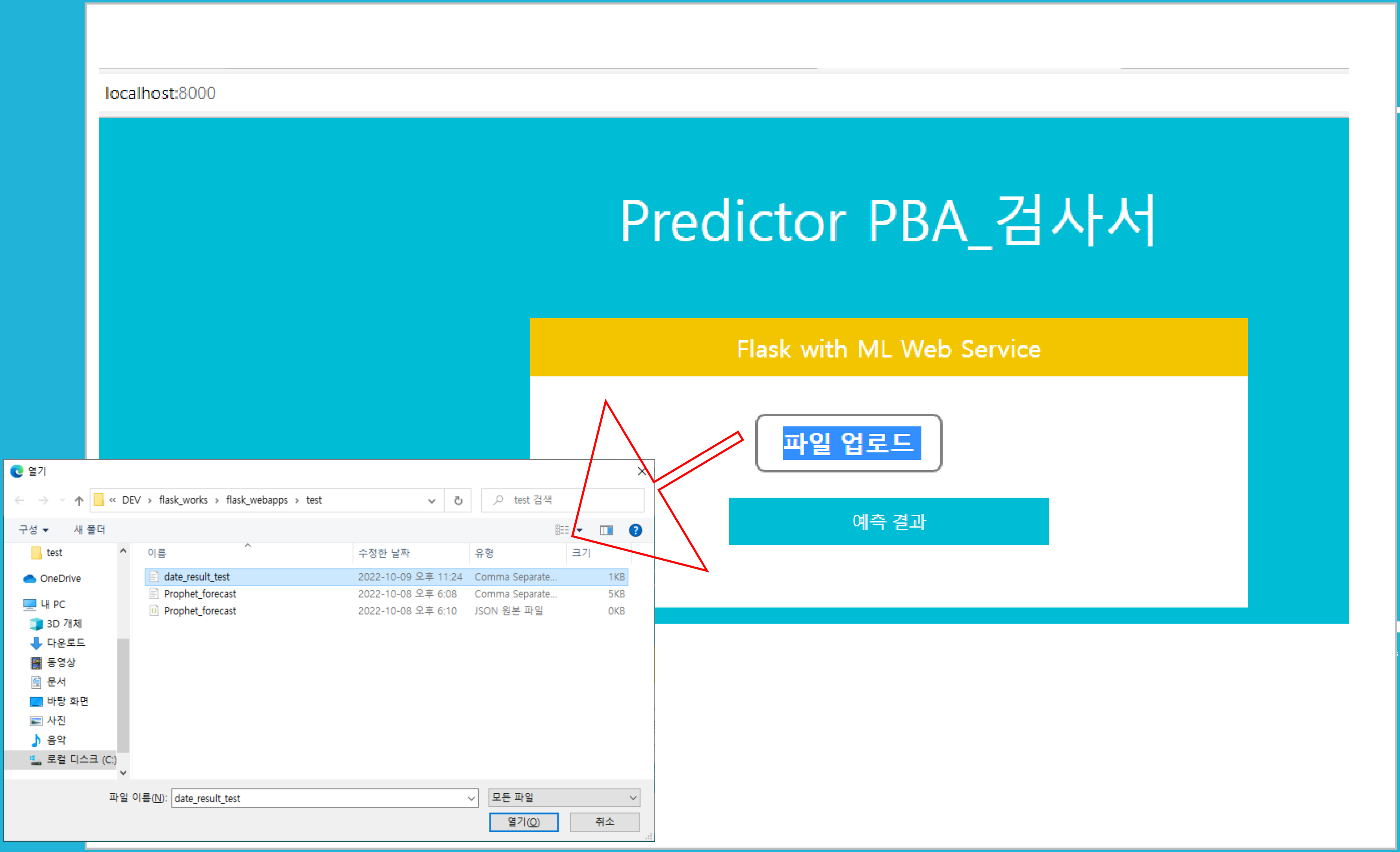
```html
<html>

<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">
```

```
INFO:werkzeug: * Restarting with stat
C:\DEV\miniconda3\envs\flask_37\lib\site-packages\numpy\_distributor_init.py:32: UserWarning: loaded more than 1 DLL from .libs:
C:\DEV\miniconda3\envs\flask_37\lib\site-packages\numpy\.libs\libopenblas.WCDJNK7YVMPZQ2ME2ZZHJJRJ3JIKNDB7.gfortran-win_amd64.dll
C:\DEV\miniconda3\envs\flask_37\lib\site-packages\numpy\.libs\libopenblas.XWYDX2IKJW2NMTWSFYNGFUWKQU3LYTCZ.gfortran-win_amd64.dll
  stacklevel=1)
WARNING:werkzeug: * Debugger is active!
INFO:werkzeug: * Debugger PIN: 655-645-777
INFO:werkzeug:127.0.0.1 - - [09/Oct/2022 23:48:58] "GET / HTTP/1.1" 200 -
```

Unit 3

실행

입력 화면과 결과 화면



localhost:8000

Predictor PBA_검사서

Flask with ML Web Service

열기

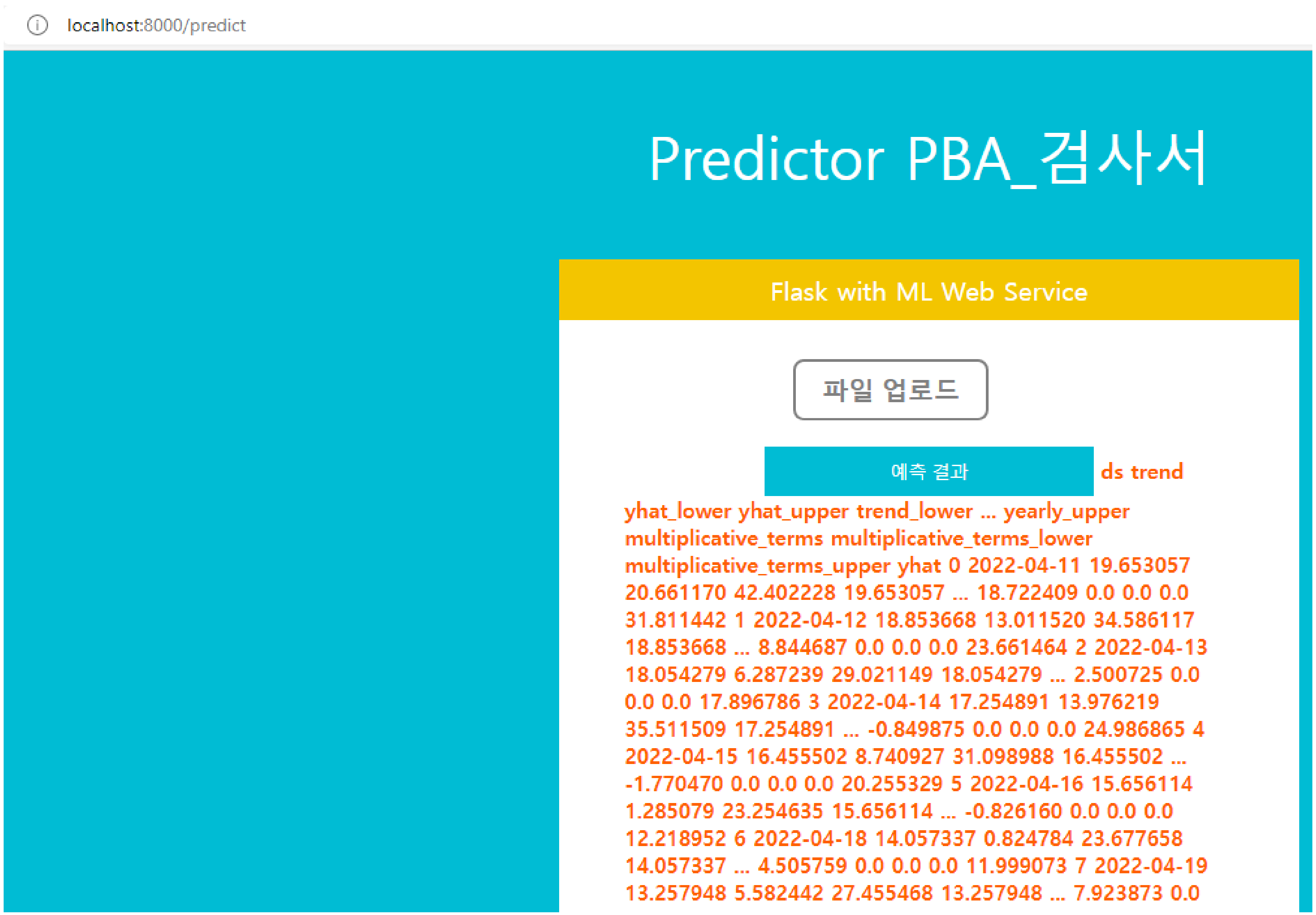
« DEV › flask_works › flask_webapps › test

이름	수정한 날짜	유형	크기
date_result_test	2022-10-09 오후 11:24	Comma Separate...	1KB
Prophet_forecast	2022-10-08 오후 6:08	Comma Separate...	5KB
Prophet_forecast	2022-10-08 오후 6:10	JSON 원본 파일	0KB

파일 이름(N): date_result_test

모든 파일

입력 화면과 결과 화면



localhost:8000/predict

Predictor PBA_검사서

```
ds trend
yhat_lower yhat_upper trend_lower ... yearly_upper
multiplicative_terms multiplicative_terms_lower
multiplicative_terms_upper yhat 0 2022-04-11 19.653057
20.661170 42.402228 19.653057 ... 18.722409 0.0 0.0 0.0
31.811442 1 2022-04-12 18.853668 13.011520 34.586117
18.853668 ... 8.844687 0.0 0.0 0.0 23.661464 2 2022-04-13
18.054279 6.287239 29.021149 18.054279 ... 2.500725 0.0
0.0 0.0 17.896786 3 2022-04-14 17.254891 13.976219
35.511509 17.254891 ... -0.849875 0.0 0.0 0.0 24.986865 4
2022-04-15 16.455502 8.740927 31.098988 16.455502 ...
-1.770470 0.0 0.0 0.0 20.255329 5 2022-04-16 15.656114
1.285079 23.254635 15.656114 ... -0.826160 0.0 0.0 0.0
12.218952 6 2022-04-18 14.057337 0.824784 23.677658
14.057337 ... 4.505759 0.0 0.0 0.0 11.999073 7 2022-04-19
13.257948 5.582442 27.455468 13.257948 ... 7.923873 0.0
```

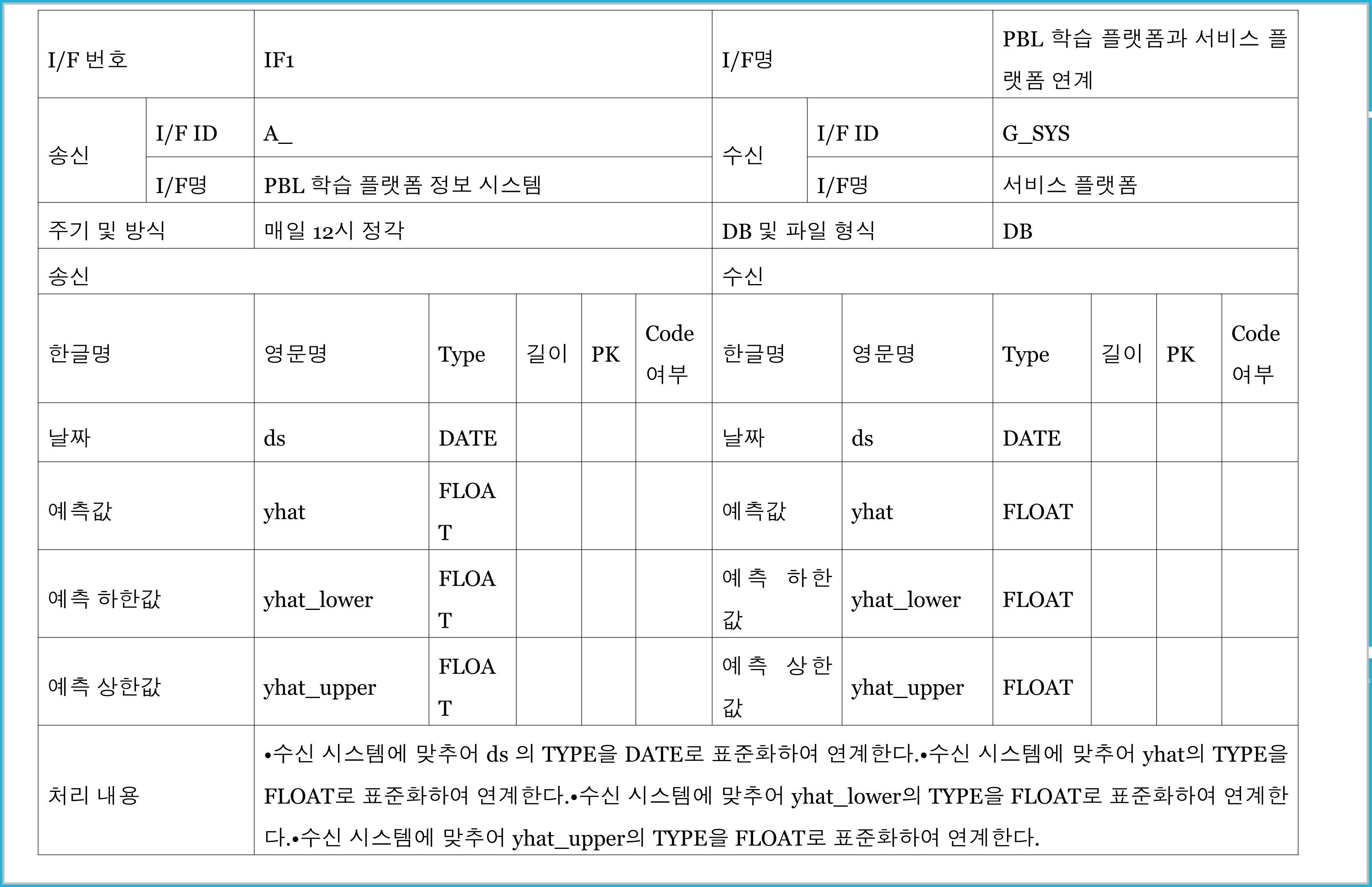
Subsection 6

인터페이스 명세서와 파일

Unit 1

인터페이스 명세서와 파일

인터페이스 명세서

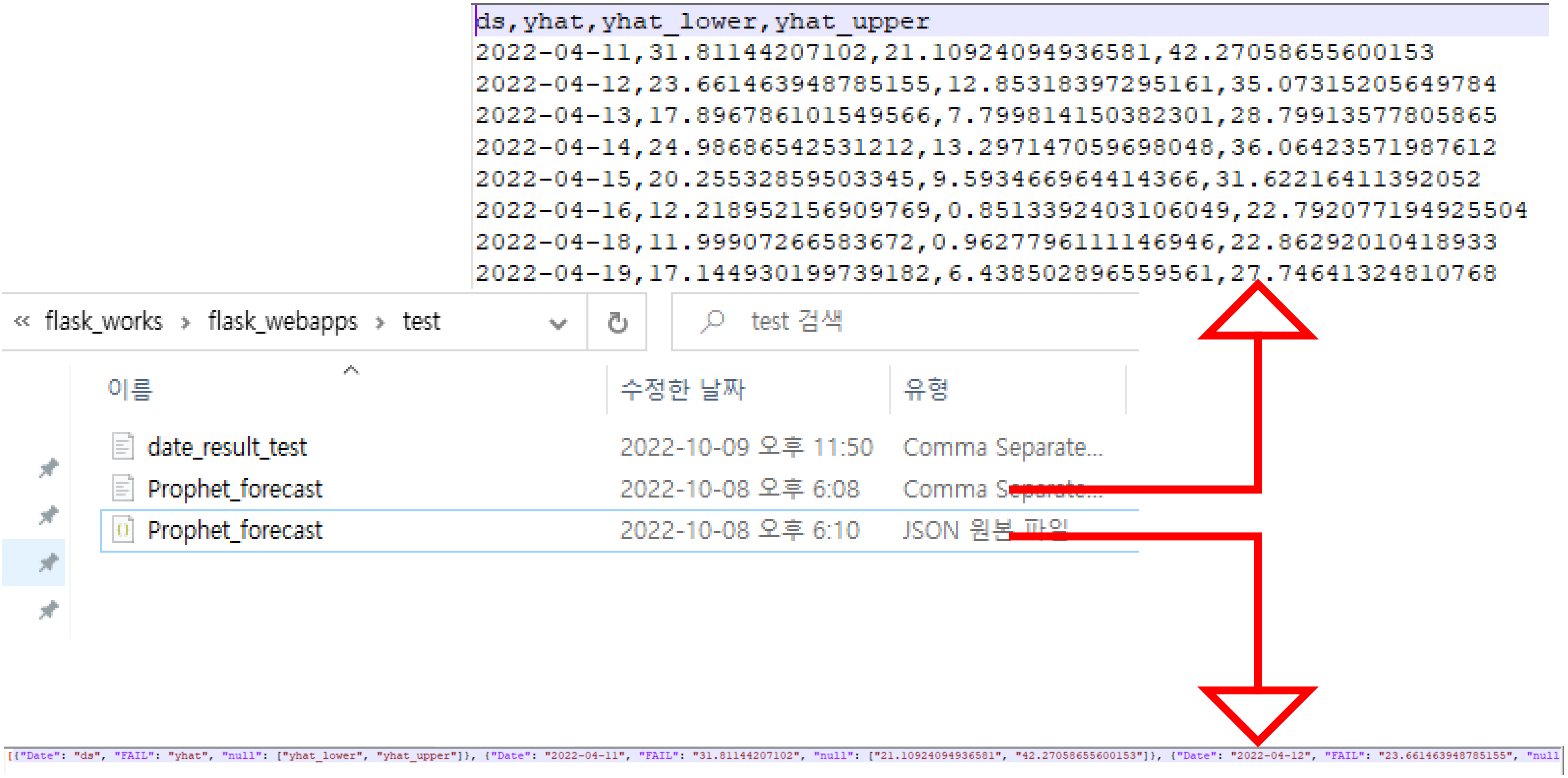


I/F 번호		IF1				I/F명			PBL 학습 플랫폼과 서비스 플랫폼 연계		
송신	I/F ID	A_				수신	I/F ID		G_SYS		
	I/F명	PBL 학습 플랫폼 정보 시스템					I/F명		서비스 플랫폼		
주기 및 방식		매일 12시 정각				DB 및 파일 형식			DB		
송신						수신					
한글명		영문명	Type	길이	PK	Code 여부	한글명	영문명	Type	길이	PK	Code 여부
날짜		ds	DATE				날짜	ds	DATE			
예측값		yhat	FLOAT				예측값	yhat	FLOAT			
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.										

Unit 1

인터페이스 명세서와 파일

인터페이스 명세서



```
ds,yhat,yhat_lower,yhat_upper
2022-04-11,31.81144207102,21.10924094936581,42.27058655600153
2022-04-12,23.661463948785155,12.85318397295161,35.07315205649784
2022-04-13,17.896786101549566,7.799814150382301,28.79913577805865
2022-04-14,24.98686542531212,13.297147059698048,36.06423571987612
2022-04-15,20.25532859503345,9.593466964414366,31.62216411392052
2022-04-16,12.218952156909769,0.8513392403106049,22.792077194925504
2022-04-18,11.99907266583672,0.9627796111146946,22.86292010418933
2022-04-19,17.144930199739182,6.438502896559561,27.74641324810768
```

이름	수정한 날짜	유형
date_result_test	2022-10-09 오후 11:50	Comma Separate...
Prophet_forecast	2022-10-08 오후 6:08	Comma Separate...
Prophet_forecast	2022-10-08 오후 6:10	JSON 원본 파일

[{"Date": "ds", "FAIL": "yhat", "null": ["yhat_lower", "yhat_upper"]}, {"Date": "2022-04-11", "FAIL": "31.81144207102", "null": ["21.10924094936581", "42.27058655600153"]}, {"Date": "2022-04-12", "FAIL": "23.661463948785155", "null

Subsection 7

테이블 목록 보기

구현 코드

Unit 1

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"))
```

구현 코드

Unit 1

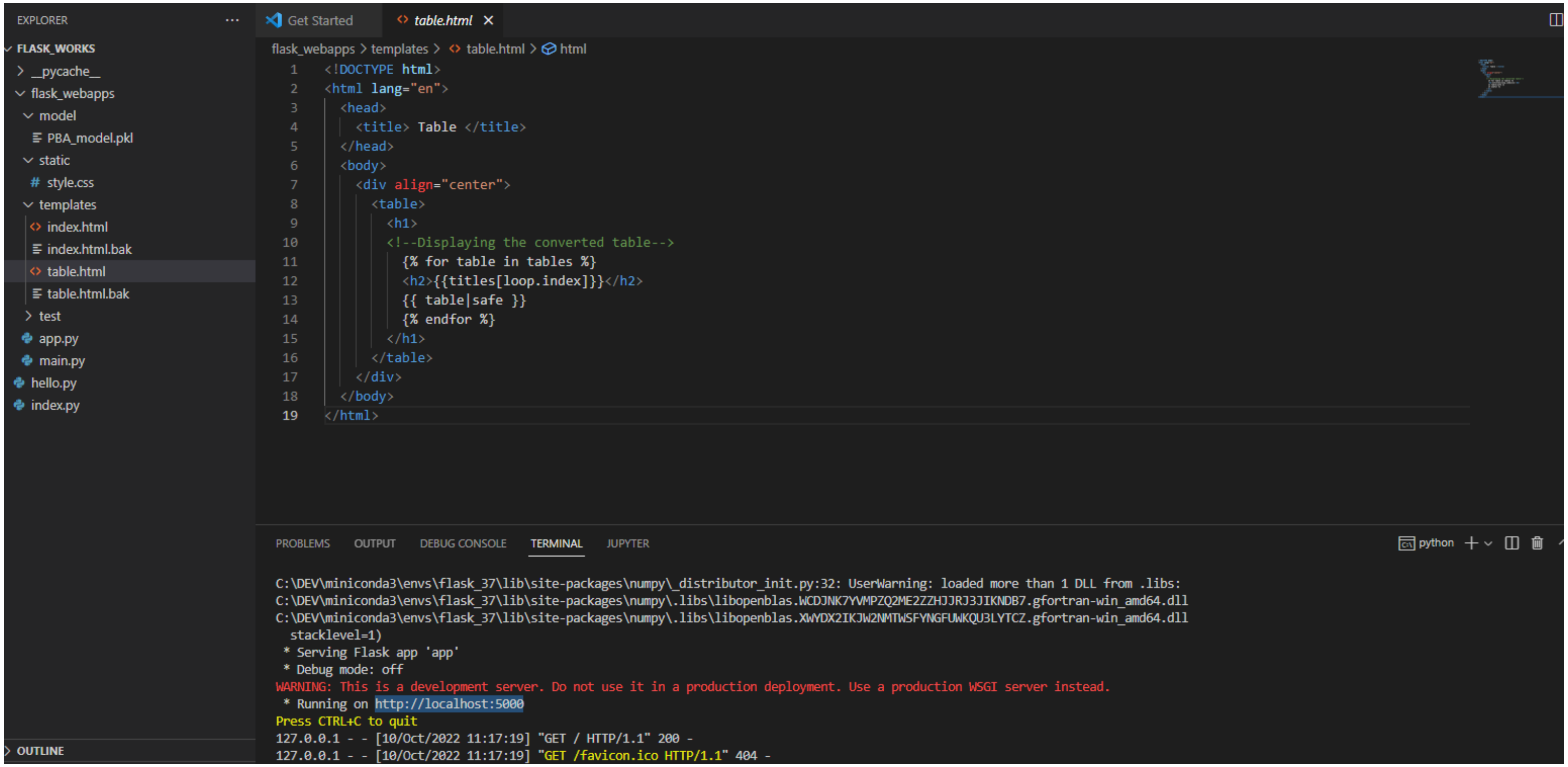
table.html

```
<!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>
```

Unit 2

실행하기

CLI 명령어



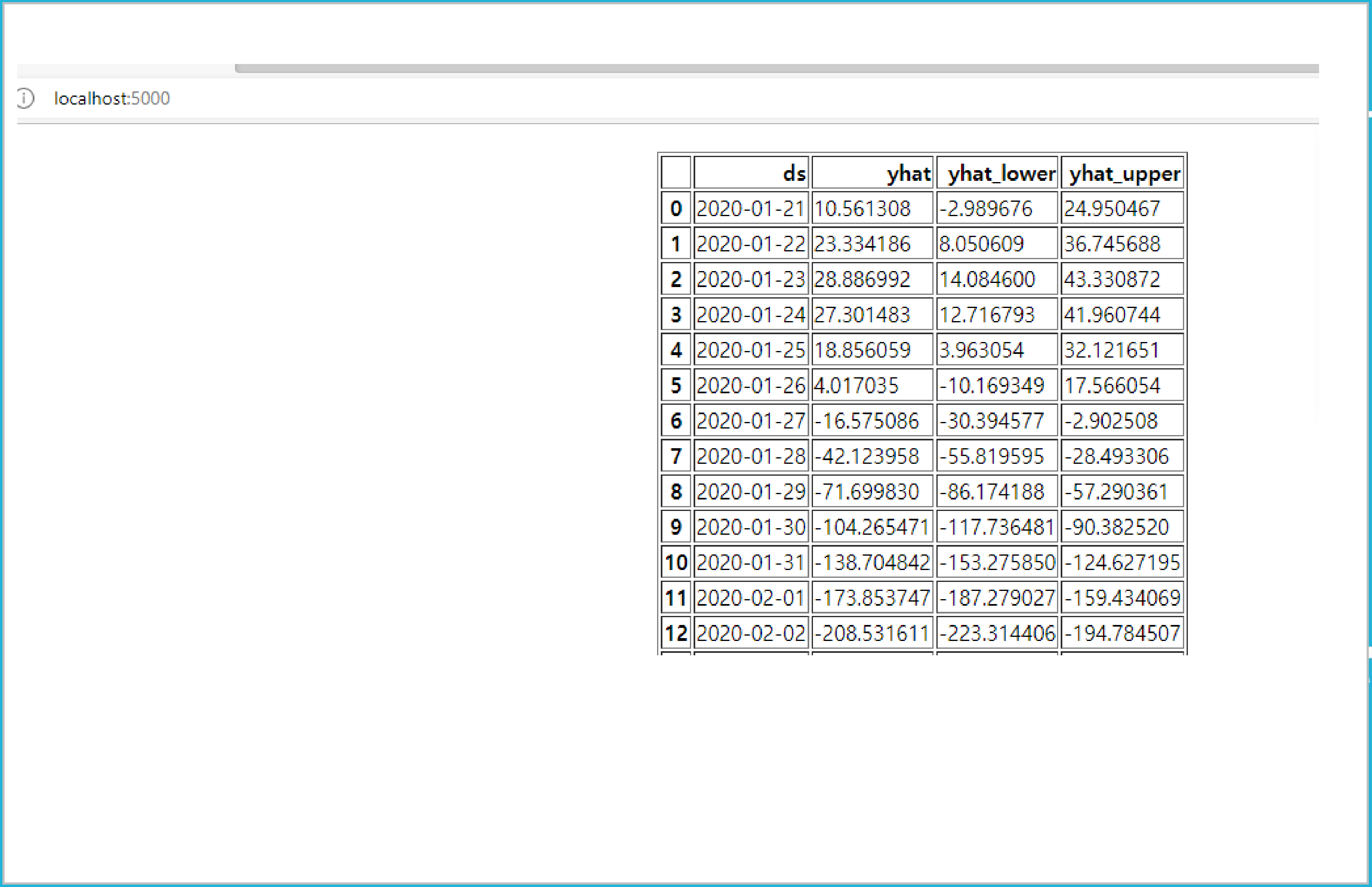
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <title> Table </title>
  </head>
  <body>
    <div align="center">
      <table>
        <h1>
        <!--Displaying the converted table-->
          {% for table in tables %}
          <h2>{{titles[loop.index]}}</h2>
          {{ table|safe }}
 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

```
C:\DEV\miniconda3\envs\flask_37\lib\site-packages\numpy\_distributor_init.py:32: UserWarning: loaded more than 1 DLL from .libs:
C:\DEV\miniconda3\envs\flask_37\lib\site-packages\numpy\.libs\libopenblas.WCDJNK7YVMPZQ2ME2ZZHJJRJ3JIKNDB7.gfortran-win_amd64.dll
C:\DEV\miniconda3\envs\flask_37\lib\site-packages\numpy\.libs\libopenblas.XWYDX2IKJW2NMTWSFYNGFUWKQU3LYTCZ.gfortran-win_amd64.dll
  stacklevel=1)
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://localhost:5000
Press CTRL+C to quit
127.0.0.1 - - [10/Oct/2022 11:17:19] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Oct/2022 11:17:19] "GET /favicon.ico HTTP/1.1" 404 -
```

Unit 2

실행하기

실행 결과



localhost:5000

	ds	yhat	yhat_lower	yhat_upper
0	2020-01-21	10.561308	-2.989676	24.950467
1	2020-01-22	23.334186	8.050609	36.745688
2	2020-01-23	28.886992	14.084600	43.330872
3	2020-01-24	27.301483	12.716793	41.960744
4	2020-01-25	18.856059	3.963054	32.121651
5	2020-01-26	4.017035	-10.169349	17.566054
6	2020-01-27	-16.575086	-30.394577	-2.902508
7	2020-01-28	-42.123958	-55.819595	-28.493306
8	2020-01-29	-71.699830	-86.174188	-57.290361
9	2020-01-30	-104.265471	-117.736481	-90.382520
10	2020-01-31	-138.704842	-153.275850	-124.627195
11	2020-02-01	-173.853747	-187.279027	-159.434069
12	2020-02-02	-208.531611	-223.314406	-194.784507

Subsection 8

CSV 데이터 sqlite DB에 저장

sqlite DB에 저장

Unit 1

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)

Unit 1

sqlite DB에 저장

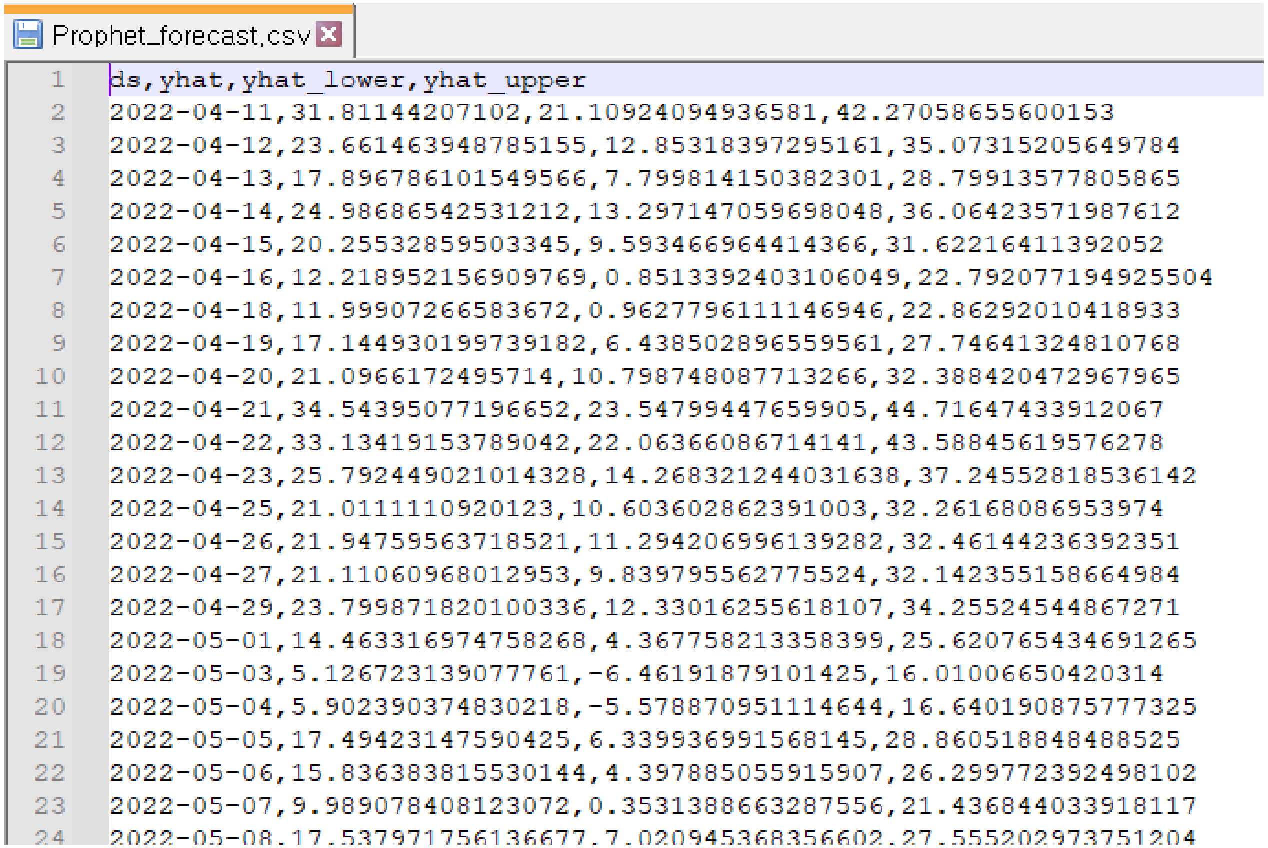
폴더 구조

```
(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>
```

Unit 1

sqlite DB에 저장

CSV 파일



```
ds,yhat,yhat_lower,yhat_upper
2022-04-11,31.81144207102,21.10924094936581,42.27058655600153
2022-04-12,23.661463948785155,12.85318397295161,35.07315205649784
2022-04-13,17.896786101549566,7.799814150382301,28.79913577805865
2022-04-14,24.98686542531212,13.29714705969804 8,36.06423571987612
2022-04-15,20.25532859503345,9.593466964414366,31.62216411392052
2022-04-16,12.218952156909769,0.8513392403106049,22.792077194925504
2022-04-18,11.99907266583672,0.9627796111146946,22.86292010418933
2022-04-19,17.144930199739182,6.438502896559561,27.74641324810768
2022-04-20,21.0966172495714,10.79874808771326 6,32.38842047296796 5
2022-04-21,34.54395077196652,23.54799447659905,44.71647433912067
2022-04-22,33.13419153789042,22.06366086714141,43.58845619576278
2022-04-23,25.792449021014328,14.268321244031638,37.2455281853614 2
2022-04-25,21.0111110920123,10.603602862391003,32.26168086953974
2022-04-26,21.94759563718521,11.294206996139282,32.46144236392351
2022-04-27,21.11060968012953,9.839795562775524,32.14235515866498 4
2022-04-29,23.799871820100336,12.33016255618107,34.2552454486727 1
2022-05-01,14.463316974758268,4.367758213358399,25.620765434691265
2022-05-03,5.126723139077761,-6.461918791014 25,16.01006650420314
2022-05-04,5.902390374830218,-5.578870951114644,16.640190875777325
2022-05-05,17.49423147590425,6.339936991568145,28.860518848488525
2022-05-06,15.836383815530144,4.397885055915907,26.299772392498102
2022-05-07,9.989078408123072,0.3531388663287556,21.436844033918117
2022-05-08,17.537971756136677,7.020945368356602,27.555202973751204
```

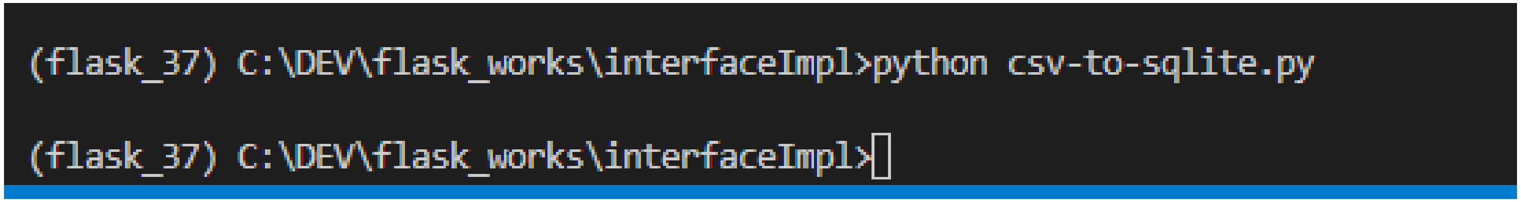
Unit 1

sqlite DB에 저장

csv-to-sqlite.py

먼저 기존에 존재하는 interfaceImpl.sqlite를 삭제한 후 수행합니다.

```
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>
```

sqlite DB에 저장

Unit 1

기본 명령어를 통한 내용 조회

```
(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>
```

Unit 2

CSV 파일 읽어들여서 highcharts + Flask 그래프 구현

app.py

```
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)
```

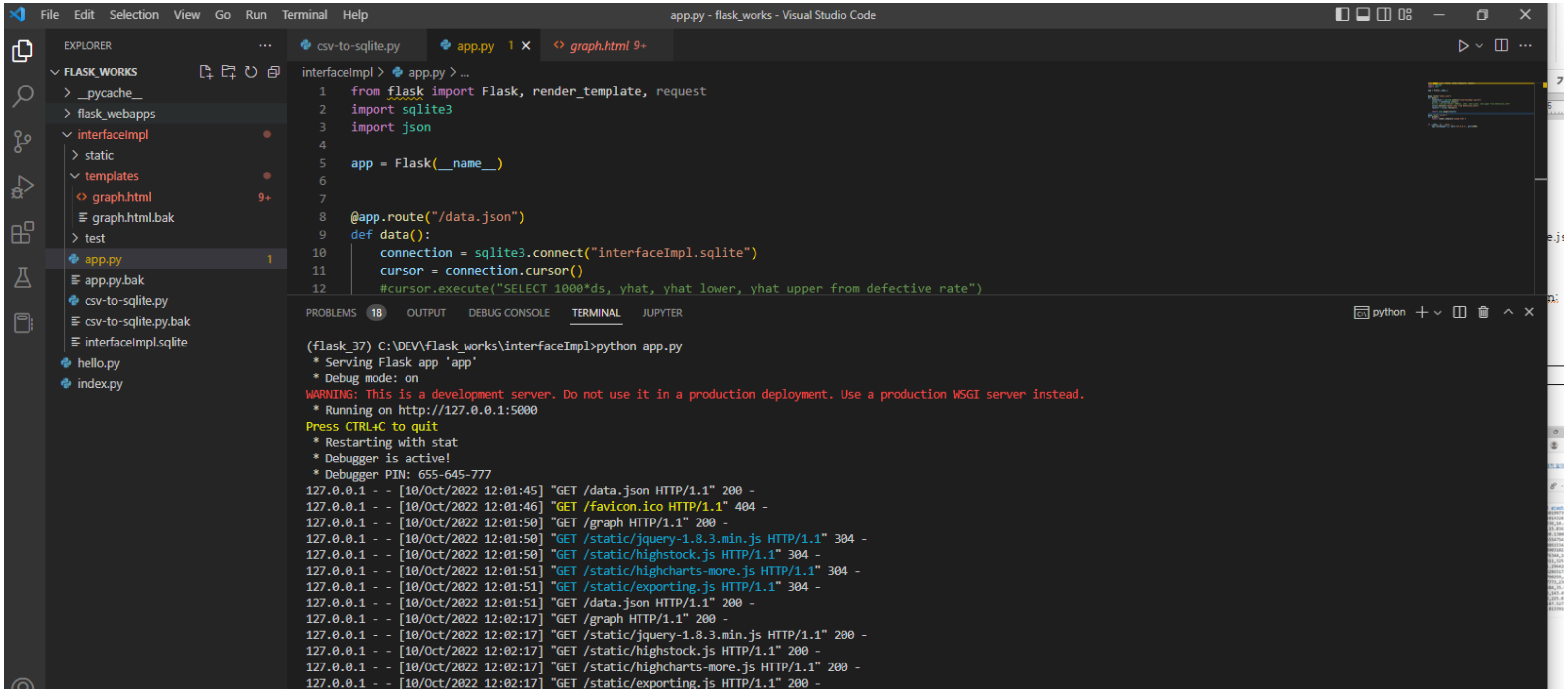
graph.html

```
<!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 : [...],
       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>
```

Unit 3

샐행

CLI 명령어



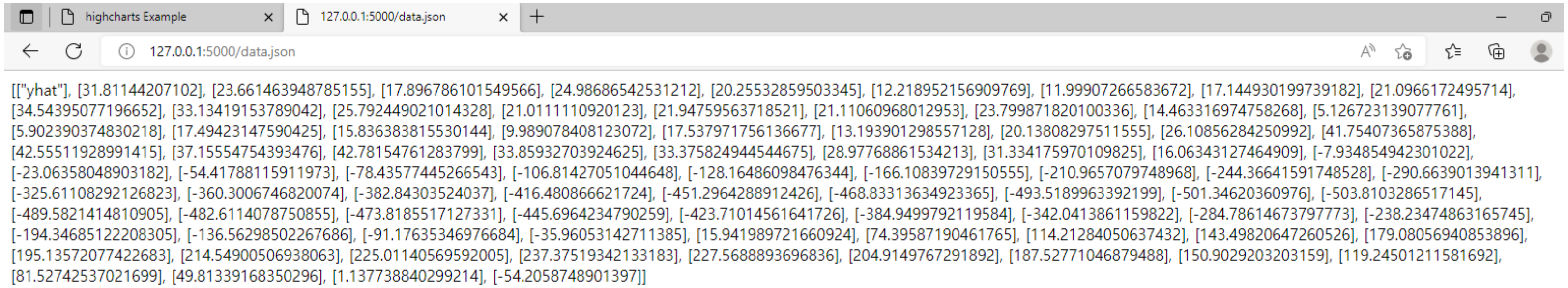
```python
from flask import Flask, render_template, request
import sqlite3
import json

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")
```

```
(flask_37) C:\DEV\flask_works\interfaceImpl>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 655-645-777
127.0.0.1 - - [10/Oct/2022 12:01:45] "GET /data.json HTTP/1.1" 200 -
127.0.0.1 - - [10/Oct/2022 12:01:46] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [10/Oct/2022 12:01:50] "GET /graph HTTP/1.1" 200 -
127.0.0.1 - - [10/Oct/2022 12:01:50] "GET /static/jquery-1.8.3.min.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Oct/2022 12:01:50] "GET /static/highstock.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Oct/2022 12:01:51] "GET /static/highcharts-more.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Oct/2022 12:01:51] "GET /static/exporting.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Oct/2022 12:01:51] "GET /data.json HTTP/1.1" 200 -
127.0.0.1 - - [10/Oct/2022 12:02:17] "GET /graph HTTP/1.1" 200 -
127.0.0.1 - - [10/Oct/2022 12:02:17] "GET /static/jquery-1.8.3.min.js HTTP/1.1" 200 -
127.0.0.1 - - [10/Oct/2022 12:02:17] "GET /static/highstock.js HTTP/1.1" 200 -
127.0.0.1 - - [10/Oct/2022 12:02:17] "GET /static/highcharts-more.js HTTP/1.1" 200 -
127.0.0.1 - - [10/Oct/2022 12:02:17] "GET /static/exporting.js HTTP/1.1" 200 -
```

Unit 3

샐행

가져온 데이터

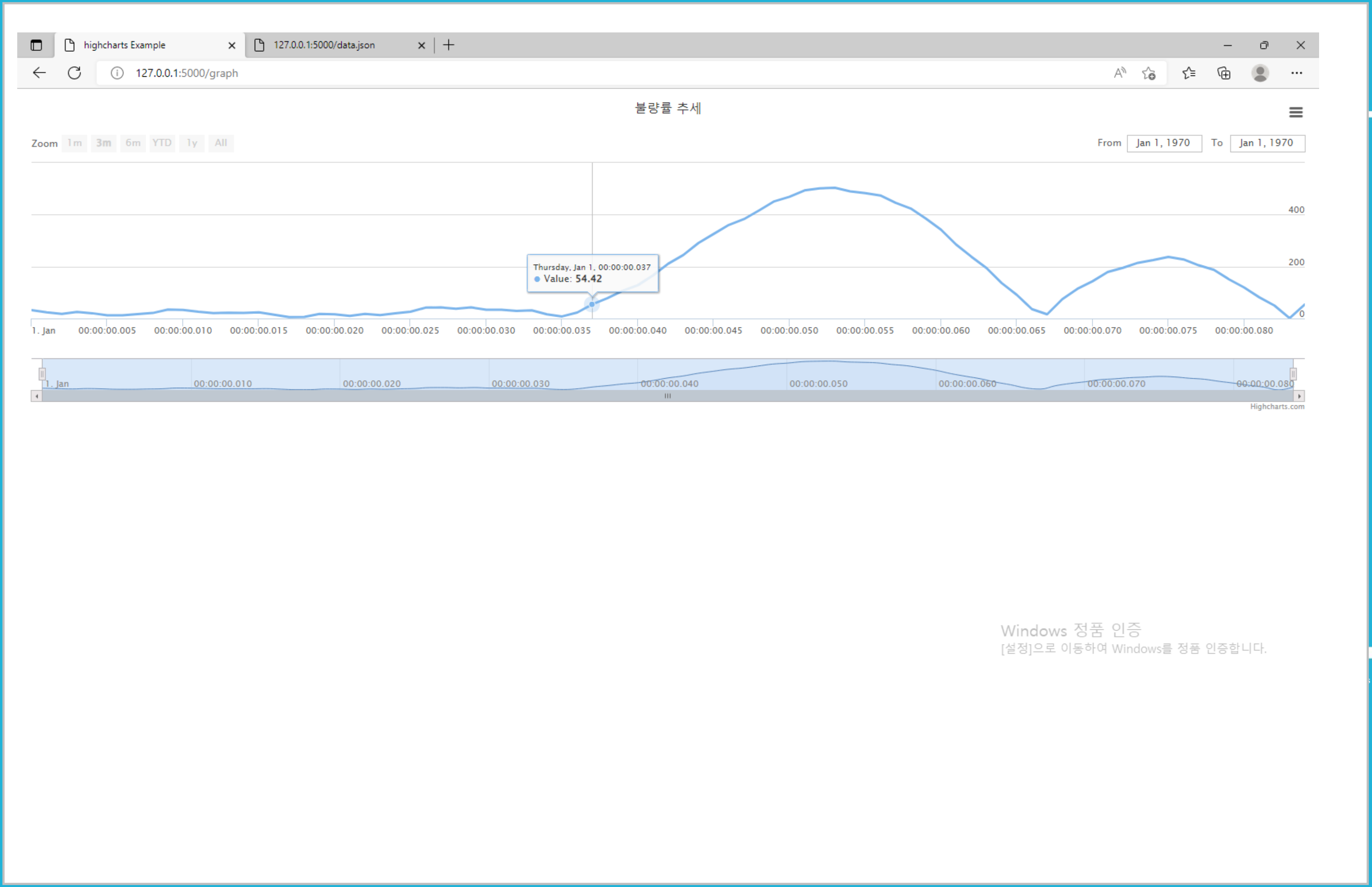


[["yhat"], [31.81144207102], [23.661463948785155], [17.896786101549566], [24.98686542531212], [20.25532859503345], [12.218952156909769], [11.99907266583672], [17.144930199739182], [21.0966172495714], [34.54395077196652], [33.13419153789042], [25.792449021014328], [21.0111110920123], [21.94759563718521], [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]]

Unit 3

샐행

결과 화면



Thursday, Jan 1, 00:00:00.037
● Value: 54.42