import pandas as pd

import numpy as np

import os

import FinanceDataReader as fdr

from sklearn.linear\_model import LinearRegression

from tqdm import tqdm

# Get Stock List

path = './open'

list\_name = 'Stock\_List.csv'

sample\_name = 'sample\_submission\_week4.csv'

stock\_list = pd.read\_csv(os.path.join(path,list\_name))

stock\_list['종목코드'] = stock\_list['종목코드'].apply(lambda x : str(x).zfill(6))

stock\_list

# Get Data & Modeling

start\_date = '20210104'

end\_date = '20210910'

start\_weekday = pd.to\_datetime(start\_date).weekday()

max\_weeknum = pd.to\_datetime(end\_date).strftime('%V')

Business\_days = pd.DataFrame(pd.date\_range(start\_date,end\_date,freq='B'), columns = ['Date'])

print(f'WEEKDAY of "start\_date" : {start\_weekday}')

print(f'NUM of WEEKS to "end\_date" : {max\_weeknum}')

print(f'HOW MANY "Business\_days" : {Business\_days.shape}', )

display(Business\_days.head())

# Baseline 모델의 구성 소개 ( Sample )

# X : (월 ~ 금) \* 33주간

# y : (다음주 월 ~ 금) \* 33주간

# y\_0 : 다음주 월요일

# y\_1 : 다음주 화요일

# y\_2 : 다음주 수요일

# y\_3 : 다음주 목요일

# y\_4 : 다음주 금요일

# 이번주 월~금요일의 패턴을 학습해 다음주 월요일 ~ 금요일을 각각 예측하는 모델을 생성

#

# 이 과정을 모든 종목(376개)에 적용

sample\_code = stock\_list.loc[0,'종목코드']

sample = fdr.DataReader(sample\_code, start = start\_date, end = end\_date)[['Close']].reset\_index()

sample = pd.merge(Business\_days, sample, how = 'outer')

sample['weekday'] = sample.Date.apply(lambda x : x.weekday())

sample['weeknum'] = sample.Date.apply(lambda x : x.strftime('%V'))

sample.Close = sample.Close.ffill()

sample = pd.pivot\_table(data = sample, values = 'Close', columns = 'weekday', index = 'weeknum')

sample.head()

model = LinearRegression()

x = sample.iloc[0:-2].to\_numpy()

x.shape

y = sample.iloc[1:-1].to\_numpy()

y\_0 = y[:,0]

y\_1 = y[:,1]

y\_2 = y[:,2]

y\_3 = y[:,3]

y\_4 = y[:,4]

y\_values = [y\_0, y\_1, y\_2, y\_3, y\_4]

x\_public = sample.iloc[-2].to\_numpy()

# 예측

predictions = []

for y\_value in y\_values :

model.fit(x,y\_value)

prediction = model.predict(np.expand\_dims(x\_public,0))

predictions.append(prediction[0])

predictions

# 실제 Public 값

sample.iloc[-1].values

# 전체 모델링

sample\_name = 'sample\_submission\_week4.csv'

sample\_submission = pd.read\_csv(os.path.join(path,sample\_name))

model = LinearRegression()

for code in tqdm(stock\_list['종목코드'].values):

data = fdr.DataReader(code, start = start\_date, end = end\_date)[['Close']].reset\_index()

data = pd.merge(Business\_days, data, how = 'outer')

data['weekday'] = data.Date.apply(lambda x : x.weekday())

data['weeknum'] = data.Date.apply(lambda x : x.strftime('%V'))

data.Close = data.Close.ffill()

data = pd.pivot\_table(data = data, values = 'Close', columns = 'weekday', index = 'weeknum')

x = data.iloc[0:-2].to\_numpy() # 2021년 1월 04일 ~ 2021년 8월 13일까지의 데이터로

y = data.iloc[1:-1].to\_numpy() # 2021년 1월 11일 ~ 2021년 8월 20일까지의 데이터를 학습한다.

y\_0 = y[:,0]

y\_1 = y[:,1]

y\_2 = y[:,2]

y\_3 = y[:,3]

y\_4 = y[:,4]

y\_values = [y\_0, y\_1, y\_2, y\_3, y\_4]

x\_public = data.iloc[-2].to\_numpy() # 2021년 8월 23일부터 8월 27일까지의 데이터를 예측할 것이다.

predictions = []

for y\_value in y\_values :

model.fit(x,y\_value)

prediction = model.predict(np.expand\_dims(x\_public,0))

predictions.append(prediction[0])

sample\_submission.loc[:,code] = predictions \* 2

sample\_submission.isna().sum().sum()

sample\_submission.columns

#결과:

# Index(['Day', '000060', '000080', '000100', '000120', '000150', '000240',

# '000250', '000270', '000660',

# ...

# '330860', '336260', '336370', '347860', '348150', '348210', '352820',

# '357780', '363280', '950130'],

# dtype='object', length=377)

columns = list(sample\_submission.columns[1:])

columns = ['Day'] + [str(x).zfill(6) for x in columns]

sample\_submission.columns = columns

sample\_submission.to\_csv('BASELINE\_Linear\_Week4.csv',index=False)

sample\_submission

from dacon\_submit\_api import dacon\_submit\_api

파일경로 = './BASELINE\_Linear\_Week3.csv'

개인\_Token = '' # 개인 토큰 발급 (https://dacon.io/account 에서 발급 가능)

대회ID = '235800' # 주식종가예측 대회 ID

팀이름 = '' # 현 대회에 참여중인 팀 명 (https://dacon.io/competitions/official/235800/team 에서 확인 가능)

submission\_메모\_내용 = 'sample\_submission'

result = dacon\_submit\_api.post\_submission\_file(

파일경로,

개인\_Token,

대회ID,

팀이름,

submission\_메모\_내용)