import pandas as pd

import numpy as np

from sklearn.datasets import load\_iris

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import GridSearchCV, train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import TimeSeriesSplit

# GridSearchCV의 param\_grid 설정

params = {

'depth': [2, 4, 6, 8, 10],

'learning\_rate': [0.01, 0.025, 0.05, 0.075, 0.1],

'iterations': [1000, 2000, 3000, 4000, 5000],

'n\_estimators' : [100, 500, 1000, 5000]

}

tscv = TimeSeriesSplit(n\_splits=2, max\_train\_size=8, test\_size=2, gap=0)

grid = GridSearchCV(estimator=model, param\_grid=params, cv=tscv, n\_jobs=1)

best\_params = []

for y\_value in y\_values:

grid.fit(x, y\_value)

best\_params.append(grid.best\_params\_)

print(" Results from Grid Search ")

print("\n The best estimator across ALL searched params:\n", grid.best\_estimator\_)

print("\n The best score across ALL searched params:\n", grid.best\_score\_)

print("\n The best parameters across ALL searched params:\n", grid.best\_params\_)

# {'depth': 2, 'iterations': 1000, 'learning\_rate': 0.01}

model = LGBMRegressor(depth=2, learning\_rate =0.01, iterations = 1000)

predictions = []

for y\_value in y\_values:

#요일별로 학습

model.fit(x,y\_value, early\_stopping\_rounds=500)

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

predictions.append(prediction[0])

predictions