import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error

from sklearn.preprocessing import StandardScaler

# 加载数据

combined\_data = pd.read\_excel('Combined\_Station\_Data.xlsx')

# 假设清洗成本为每次清洗的固定成本

cleaning\_cost\_per\_cleaning = 500 # 假设清洗一次的成本为500元

# 假设电价为每kWh 0.6元

electricity\_price = 0.6 # 初始电价，每kWh的电价

# 选择输入特征和输出变量

features = ['Humidity', 'Irradiance\_w\_m2', 'CurrentTemperature', 'PR', 'ActualEnergy\_kWh', 'TheoreticalEnergy\_kWh']

target = 'ActualEnergy\_kWh'

# 数据预处理：去除缺失值

combined\_data = combined\_data.dropna(subset=features + [target])

# 特征标准化

scaler = StandardScaler()

combined\_data[features] = scaler.fit\_transform(combined\_data[features])

# 训练集与测试集划分

X = combined\_data[features]

y = combined\_data[target]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 训练一个随机森林回归模型

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# 模型预测

y\_pred = model.predict(X\_test)

# 计算均方误差（MSE）

mse = mean\_squared\_error(y\_test, y\_pred)

print(f'Mean Squared Error: {mse}')

# 模拟不同电价对总成本和发电量的影响

# 假设电价变动的范围：电价上涨10%，下降10%

electricity\_price\_changes = [electricity\_price \* 1.1, electricity\_price, electricity\_price \* 0.9]

cleaning\_frequencies = [1, 2, 3, 5, 7] # 清洗频率（单位：天）

total\_costs = {price: [] for price in electricity\_price\_changes}

total\_energy = {price: [] for price in electricity\_price\_changes}

# 设定权重

w1 = 0.3 # 清洗成本权重

w2 = 0.3 # 发电损失成本权重

w3 = 0.4 # 发电收益权重

for price in electricity\_price\_changes:

for cleaning\_freq in cleaning\_frequencies:

total\_loss\_cost = 0

total\_cleaning\_cost = 0

total\_generation = 0

for i in range(len(combined\_data)):

# 预测每个数据点的发电量

predicted\_energy = model.predict(X.iloc[i:i+1])[0] # 预测发电量

actual\_energy = combined\_data.ActualEnergy\_kWh.iloc[i] # 实际发电量

theoretical\_energy = combined\_data.TheoreticalEnergy\_kWh.iloc[i] # 理论发电量

# 计算发电损失

energy\_loss = (theoretical\_energy - actual\_energy) \* price # 考虑电价变化

total\_loss\_cost += energy\_loss

# 计算清洗成本

if i % cleaning\_freq == 0:

total\_cleaning\_cost += cleaning\_cost\_per\_cleaning # 累计清洗成本

# 累加发电量

total\_generation += predicted\_energy

# 计算综合成本：综合考虑清洗成本、发电损失成本和发电收益

total\_cost = w1 \* total\_cleaning\_cost + w2 \* total\_loss\_cost - w3 \* total\_generation

total\_costs[price].append(total\_cost)

total\_energy[price].append(total\_generation)

# 绘制折线图：不同电价下的总成本与发电量变化

fig, ax1 = plt.subplots(figsize=(10, 6))

for price in electricity\_price\_changes:

ax1.plot(cleaning\_frequencies, total\_costs[price], marker='o', label=f'Cost at ${price:.2f} per kWh')

ax1.set\_xlabel('Cleaning Frequency (days)')

ax1.set\_ylabel('Total Cost (Yuan)')

ax1.legend(loc='upper left')

ax1.set\_title('Effect of Electricity Price on Cleaning Cost')

ax2 = ax1.twinx() # 共享x轴

for price in electricity\_price\_changes:

ax2.plot(cleaning\_frequencies, total\_energy[price], marker='o', label=f'Energy at ${price:.2f} per kWh')

ax2.set\_ylabel('Total Energy (kWh)')

ax2.legend(loc='upper right')

fig.tight\_layout()

plt.show()

# 输出结果：不同电价下最优清洗频率

for price in electricity\_price\_changes:

min\_cost\_index = np.argmin(total\_costs[price])

optimal\_cleaning\_frequency = cleaning\_frequencies[min\_cost\_index]

print(f"At price ${price:.2f} per kWh, the optimal cleaning frequency is every {optimal\_cleaning\_frequency} days.")