PP Challenge

The challenge consists of two parts:

1. Prediction of the hourly electricity price for the last day of the data,
2. Finding the optimal charge/discharge setting to maximize the trading revenue.

1. Forecasting Problem

Data: The dataset is very small, containing only 96 samples to train machine learning models, and 24 samples to predict the hourly wholesale price of electricity. The dataset contains $Date$, $Hour$, and $RegionalPowerDemand$ information to predict the wholesale price of electricity. In all my data pipelines, I used only the $Hour$ and $PowerDemand$ columns to train models and discarded the $Date$ column as the dataset does not reveal any seasonal (weekly, monthly, yearly) patterns.

Models: I trained Xgboost and Lightgbm models. The Xgboost data pipeline transformed the $Hour$ column with the OneHotEncoder and standard-scaled the power demand whereas the lightgbm data pipeline transformed the $Hour$ column with the OrdinalEncoder.

To tune the models, I used the $Optuna$ hyper-parameter optimization framework. The key characteristic of Optuna is that it allows eager search spaces with Bayesian sampling, which is not possible with scikitlearn's parameter optimization tools.

In hyper-parameter tuning, I trained the models over randomly split 3 deterministic Folds. The objective function was the average of the root mean squared errors.

Results: The results of the parameter tuning are stored in xxx. The best Xgboost and Lightgbm models yielded the following rms errors:

Xgboost rmse: $3.56

Lightgbm rmse: $3.45

2. Optimization Problem

I predicted the wholesale hourly price of electricity with the best lightgbm model trained on the first part. To solve the optimization problem, I used GEKKO python package. The expected revenue from this exercise is found to be $3109.

Sets

To solve the optimization problem, I define the following set for each 1hr in a given day.

I = {i | i is an integer, and 0 $\leq i \leq$ 23 }.

Parameters

The parameters used in defining the optimization problem is as follows:

BC: Battery capacity = 200 MWh

R_max: Maximum battery charge/discharge rate. Positive for charge (+100 MW), negative for discharge (-100 MW).

P_i : Wholesale hourly electricity price.

t_i: Time bin = 1hour.

Decision Variables

R_i: Battery charge/discharge rate for a given time chunk of the day. Positive for charging mode, negative for discharging mode.

Objective

Given the hourly price of electricity and power input/output rates, I define the following objective function:

$$maximize(\sum_{i=0}^{23} P_i.\, R_i) \tag{1}$$

Constraints

The first two constaints ensure that the system will use only 1 charge/discharge cycle. The thrid constraint ensures that the battery system has enought enery to sell.

$$\sum R_i = 0 \tag{2}$$

$$|\sum R_i| * t_i = 2BC = 400\ MWh \tag{3}$$

$$R_i + \sum_{j=0}^{i-1} Rj <= 0 \tag{4}$$

Boundaries

The power input/output is constrained to be between -100 MW and +100 MW. However, in practice this parameter pegs can take only -100, 0, 100 to maximize the revenue.

$$-100\ MW < R_i < +100\ MW \tag{5}$$

Installation

```
git clone https://github.com/tdincer/PPChallenge.git
cd PPChallenge
pip3 install requirements.txt
```

Training Forecasting Models with Optuna

The forecasting models are in the $prediction$ folder.

To train xgboostregressor for 30 rounds with optuna:

```
python3 train.py -e xgb -30
```

To train lightgbmregressor for 30 rounds with optuna:

```
python3 train.py -e lgb -30
```

All scores for each training run are stored in the $study$ folder.

Inference

To get the predictions:

```
python3 inference.py
```

The command above produces the predictions in xgb_submission.csv and lgb_submission.csv files.

Optimization Code

The optimization code is located in the $optimization$ folder. To find the optimum battery charge/discharge strategy, go to optimization folder and run:

```
python3 scheduler.py
```