# Value in Energy Data Seminar: POD Data Science Challenge

Team “Cameron” solution

Dr. Cameron Roach

16 June, 2021

# Introduction

This presentation covers the winning solution by team "Cameron" for the Presumed Open Data: Data Science Challenge. It will focus on:

1. training data
2. feature engineering
3. forecasting models
4. scheduling methodology.

A short example for one of the competition forecast dates will be presented at the end.

# Data used

| Data | Interval | In horizon? | Used? |
|---|---|---|---|
| Electricity demand | Half-hourly | No | Yes |
| Reanalysis temperature (6 locations) | Hourly | Yes | Yes |
| Reanalysis irradiance (6 locations) | Hourly | Yes | Yes |
| Solar PV generation | Half-hourly | No | Yes |
| Solar irradiance data | Half-hourly | No | No |
| PV module temperature | Half-hourly | No | No |

Solar irradiance and PV module temperature data collected at the solar farm were not used.

Reanalysis temperature and irradiance data were linearly interpolated to create half-hourly values.

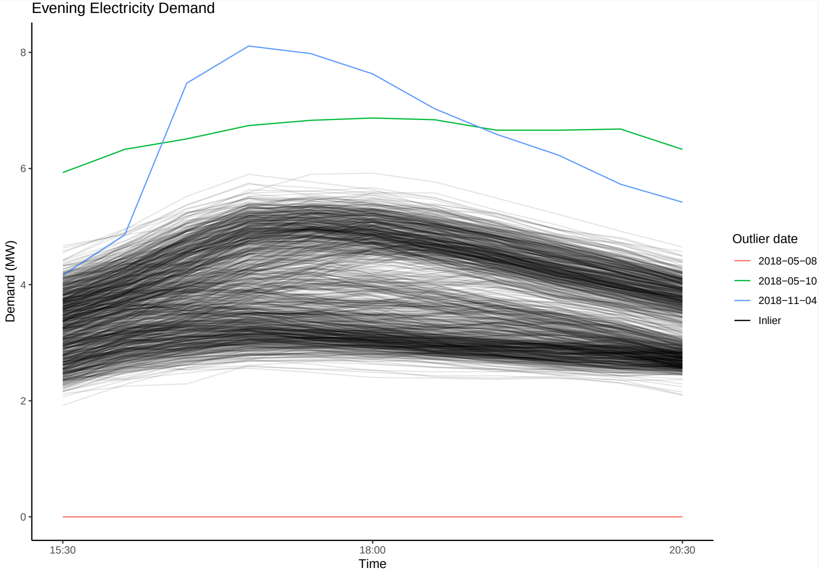
# Methodology

Focused on producing accurate forecasts for both PV generation and electricity demand. The process for battery scheduling involved the following steps:

1. Produced deterministic forecasts for **electricity demand**.
2. Produced deterministic/probabilistic forecasts for **PV generation**.
3. **Scaled PV generation forecast** to obtain battery charging schedule.
4. Scheduled battery discharging to produce **flat evening demand** (based on forecast).

# Electricity demand model

- Only fitted using evening demand between 3:30pm and 9:00pm.
- Three dates removed as they appeared to be outliers.



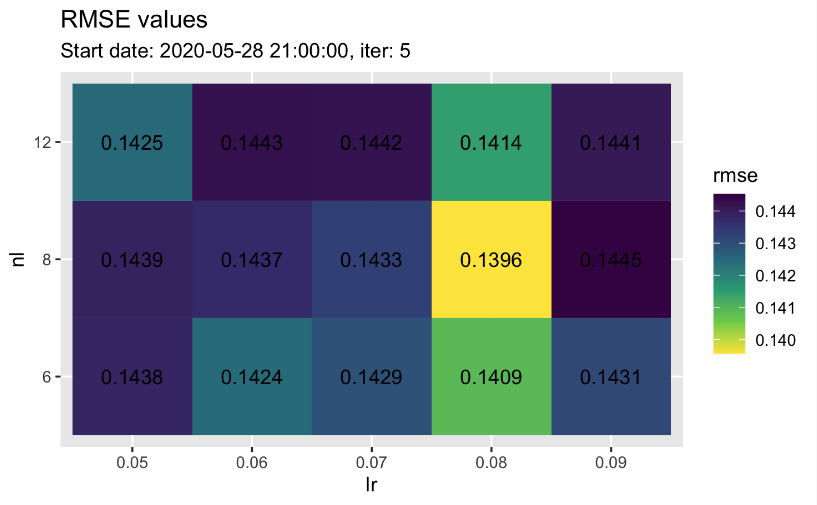
# Methodology

Trained model to predict half-hourly demand using **gradient boosted decision trees**. Implemented using `lightgbm`.

The demand model used the following features:

- half-hour period of day
- day of year
- day of week
- public holidays
- lagged demand (1-week)
- temperature and lags (1, 2, 6, 12, 24, 48 and 96 half-hourly lags) for all 6 reanalysis locations
- solar irradiance and lags (1, 2, 6, 12, 24, 48 and 96 half-hourly lags) for all 6 reanalysis locations
- previous day's maximum, minimum and average temperatures across all 6 reanalysis locations
- trend.

Highly parameterized model! Needed to take care to avoid over-fitting. Used time series cross validation to tune **number of leaves** and **learning rate** parameters. Typically used 5 “week-long” folds and RMSE to assess accuracy, but experimented throughout the competition.



# Methodology

**Probabilistic PV generation model**

The PV generation model was again fit using **gradient boosted decision trees** implemented using `lightgbm`.

Originally focused on point forecasts for PV. Experimented with quantile forecasts towards the end of the competition.

The following features were used by the PV generation model:

- half-hour period of day
- month of year
- day of year
- lagged PV generation (1-week)
- temperature and lags (1, 2, . . . , 6 half-hourly lags) for all 6 reanalysis locations
- solar irradiance and lags (1, 2, . . . , 6 half-hourly lags) for all 6 reanalysis locations.

# Methodology

**Scheduling battery (charging)**

- Scaled PV prediction so that total energy was equal to 6MWh over the allowed charging times. This served as the battery charging schedule.
  - Less charging occurs during times that are expected to have low PV generation, more charging occurs when we expect more PV generation.
- No penalty for importing from grid, so always aimed to fully charge battery.

# Methodology

**Scheduling battery (discharging)**

Ideally, we want demand to be completely flat to ensure the peak has been reduced as much as possible. Needed to calculate what the new peak would be:

$$\sum_{k=32}^{42} \hat{L}_{d,k} = L_d^* \times \underbrace{|\{32, \ldots, 42\}|}_{=11} + 12 \quad \Rightarrow \quad L_d^* = \frac{\sum_{k=32}^{42} \hat{L}_{d,k} - 12}{11},$$

where $L_d^*$ is the new expected peak between periods 32 to 42 for day $d$ and $\hat{L}_{d,k}$ is our forecast demand.

# Methodology

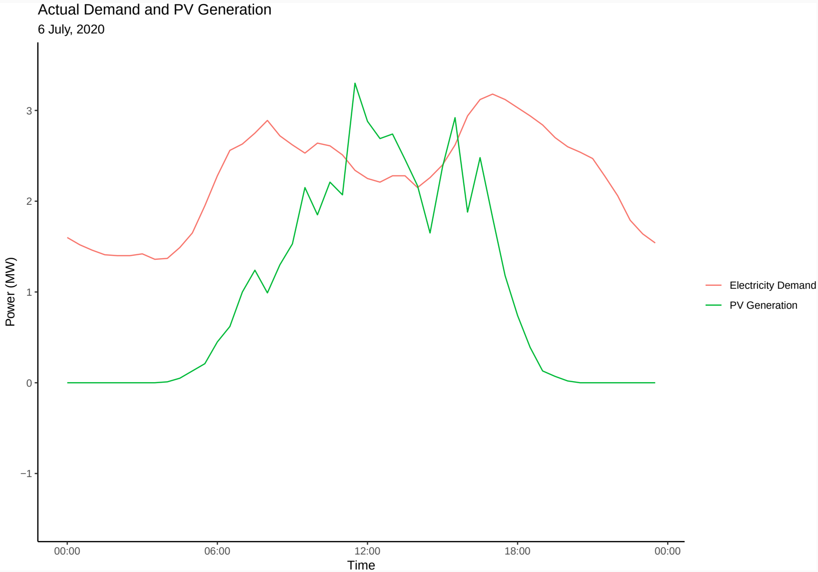
From here, we can simply take the difference between the forecast electricity demand and $L_d^*$ to obtain the required discharge for each half-hourly period on day $d$:

$$B_{d,k} = L_d^* - \hat{L}_{d,k},$$

for $k \in \{32, \ldots, 42\}$.

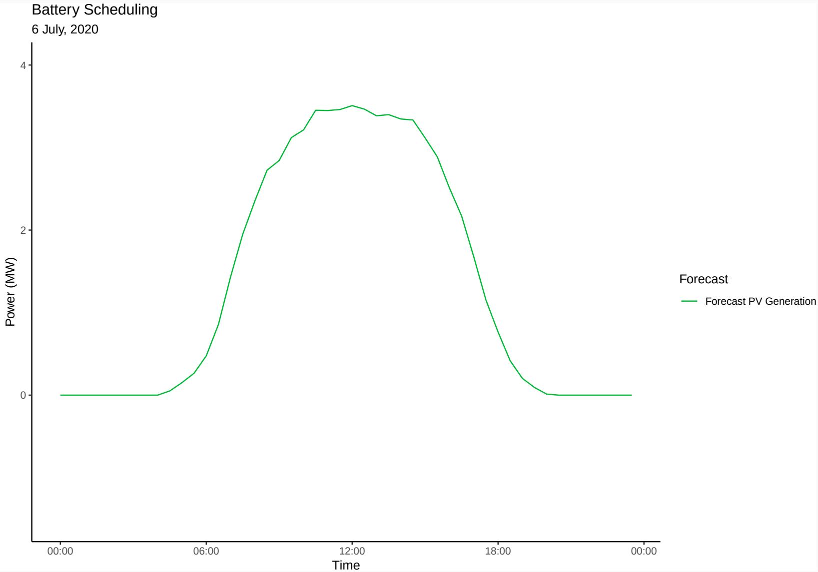
Note, there were some checks to ensure battery constraints weren't violated, but these didn't actually come into play over the duration of the competition — $B_{d,k}$ values were consistently less than 2.5MW in magnitude.

# Example



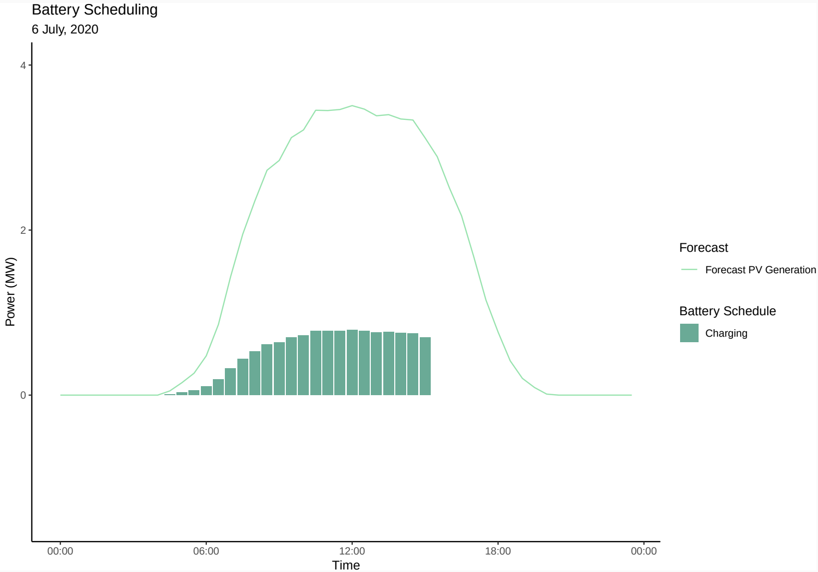
Actual Demand and PV Generation
6 July, 2020
Power (MW)
3
2
1
0
−1
00:00
06:00
12:00
18:00
00:00
Time
Electricity Demand
PV Generation

# Example

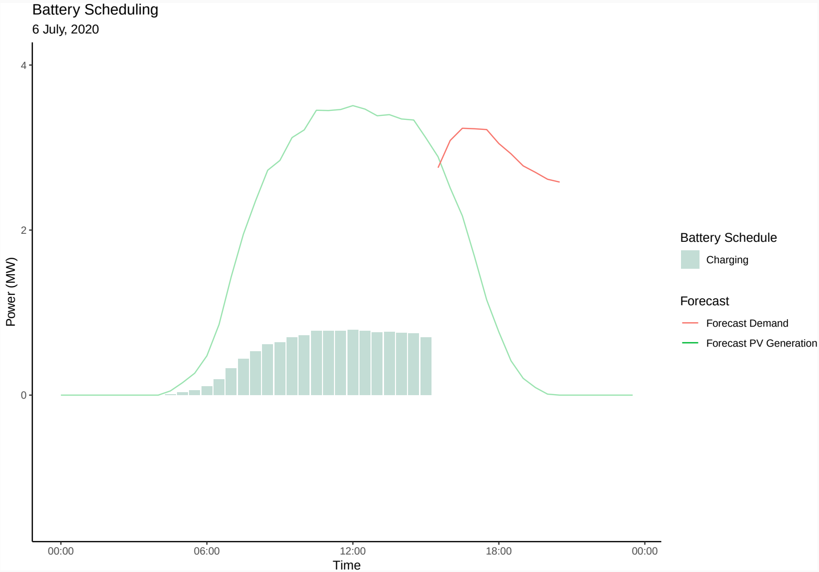


Battery Scheduling
6 July, 2020
Power (MW)
4
2
0
00:00
06:00
12:00
18:00
00:00
Time
Forecast
Forecast PV Generation

# Example

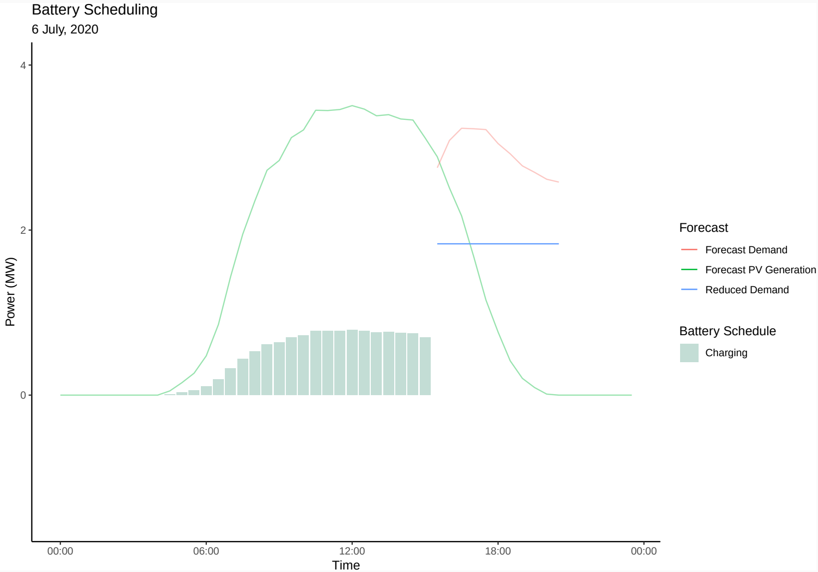

Battery Scheduling
6 July, 2020
4
2
0
Power (MW)
00:00
06:00
12:00
18:00
00:00
Time
Forecast
Forecast PV Generation
Battery Schedule
Charging

# Example



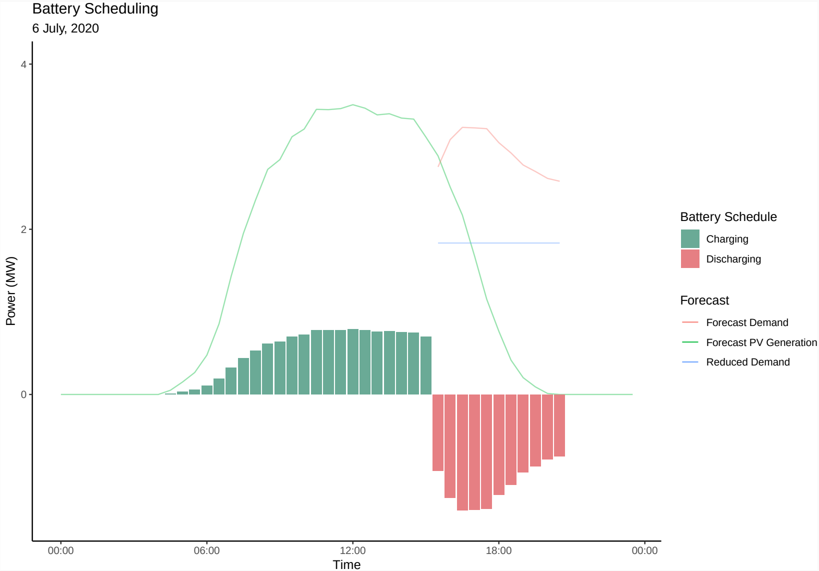
Battery Scheduling
6 July, 2020
Power (MW)
4
2
0
00:00
06:00
12:00
18:00
00:00
Time
Battery Schedule
Charging
Forecast
Forecast Demand
Forecast PV Generation

# Example

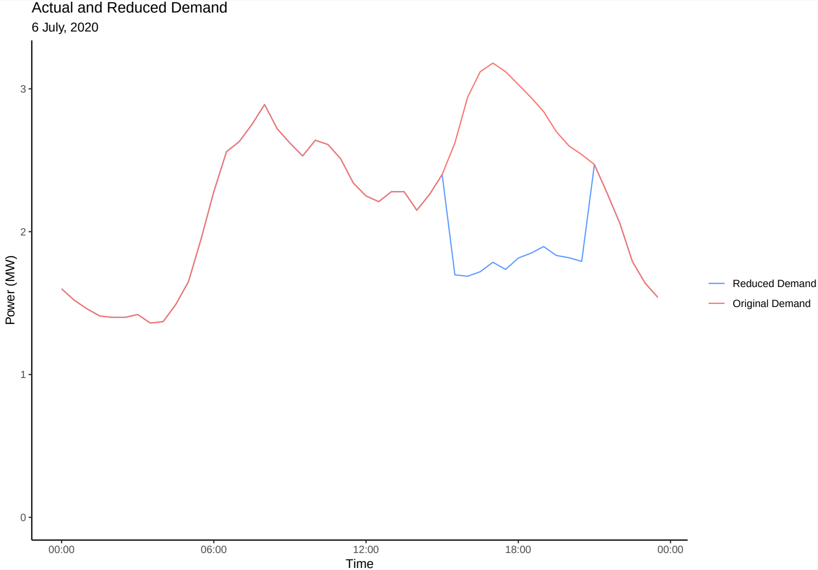


Battery Scheduling
6 July, 2020
Power (MW)
4
2
0
00:00
06:00
12:00
18:00
00:00
Time
Forecast
Forecast Demand
Forecast PV Generation
Reduced Demand
Battery Schedule
Charging

# Example


Battery Scheduling
6 July, 2020
Power (MW)
4
2
0
00:00
06:00
12:00
18:00
00:00
Time
Battery Schedule
Charging
Discharging
Forecast
Forecast Demand
Forecast PV Generation
Reduced Demand

# Example



Actual and Reduced Demand
6 July, 2020
Power (MW)
3
2
1
0
00:00
06:00
12:00
18:00
00:00
Time
Reduced Demand
Original Demand

# Code

All code used for this entry has been implemented as an R package.

Available from github.com/camroach87/pod-energy-comp.

Code for this presentation is available from github.com/camroach87/presentations.

# Thank you! Questions?