ASHRAE Great Energy Predictor III

Team 8 : Kongdan Zhou, Jaeho Kim, Kyunggeun Do, Unsoo Kang

Programming for Data Science

2026.06.02

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Contents

1. INTRODUCTION
2. EDA
3. MODEL
4. RESULTS & DISCUSSION
5. CONCLUSION

Competition Background - ASHRAE Great Energy Predictor III

Who runs it, and why it exists

ASHRAE
the American Society of Heating, Refrigerating and Air-Conditioning Engineers, founded in 1894 — promotes sustainable technology for the built environment.

Great Energy Predictor III (GEPIII)
Third energy-prediction contest and the first since the mid-1990s
The Largest building-energy machine-learning competition ever held.



THE COMPETITION IN NUMBERS

4,370
participants

3,614
teams across 94 countries

39,403
predictions submitted

415+
public solution notebooks shared

Problem Definition

- **Task & Objective**

A large-scale time-series regression problem

Predict hourly **meter_reading** (energy use in kWh) for each building and meter. Train on one year of history and forecast the next two years.

WHAT COUNTS AS ONE PREDICTION

1 building × **1 hour**

→ *one row of energy use to forecast*

TIMELINE

TRAIN
2016 · 1 year

→

PREDICT
2017 – 2018 · 2 years

EVALUATION METRIC

RMSLE

Root Mean Squared Logarithm

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1) - \log(a_i+1))^2}$$

Penalises relative error, not absolute scale — outliers are dampened.

FOUR METER TYPES TO MODEL

Electricity	code 0
Chilled water	code 1
Steam	code 2
Hot water	code 3

Not every building has all four meters.

Problem Definition

- **Why Predict Energy?**

To prove an efficiency upgrade actually saved money and measure how much.

1 A building is upgraded

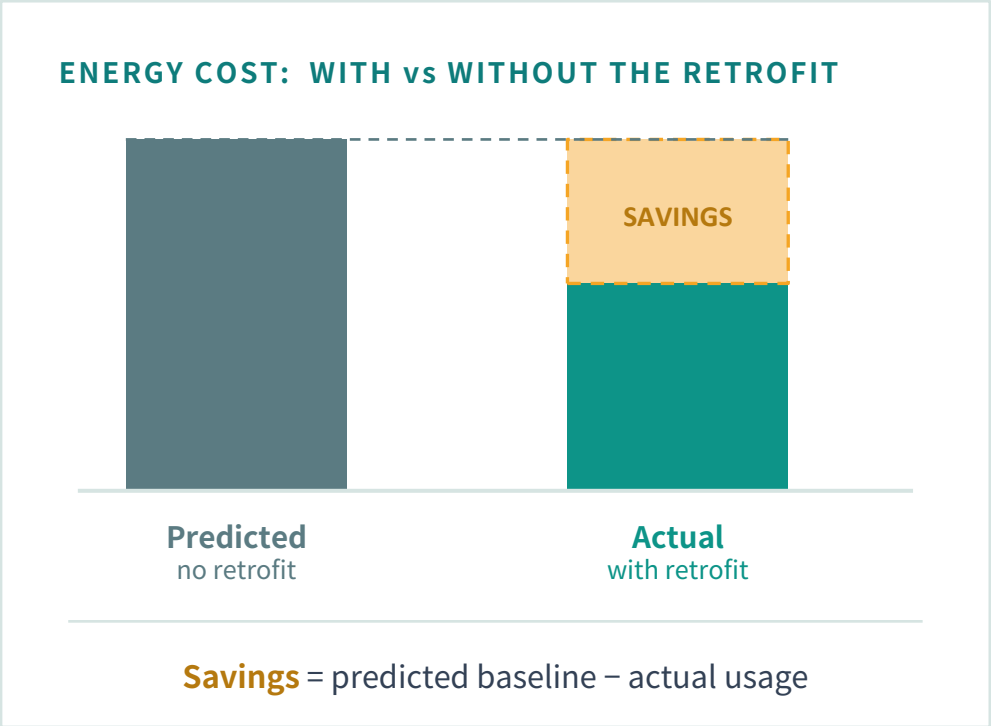
An owner invests in an energy retrofit (efficient HVAC, insulation, controls). Was the cost worth it?

2 The baseline is invisible

You can't re-run the same year without the upgrade. What it WOULD have used is never observed.

3 So, we predict it

A model trained on past usage estimates that hidden baseline — consumption as if no retrofit happened.



WHY IT'S WORTH SOLVING

The more accurate the prediction, the more trustworthy the savings number. Reliable savings turn efficiency upgrades into bankable investments — so financiers fund them at scale, cutting cost, energy, and emissions.

Dataset Overview

- **Datasets**

TIMELINE · 3 YEARS OF HOURLY DATA

TRAIN · 2016
X + Y both given → learn the pattern

TEST · 2017 – 2018
X given, Y hidden → predict 2 years of energy use

Jan 2016 — Jan 2017 — Dec 2018

~20M	1,448	2,380	16	~41M
training readings	buildings	energy meters	sites / sources	points to predict

INPUTS (X) — features the model learns from

Building

`site_id · primary_use · square_feet · year_built · floor_count`

Weather

`air_temperature · dew_temperature · cloud_coverage · wind_speed · wind_direction · precip_depth_1_hr · sea_level_pressure`

Keys & time

`building_id · meter (0–3) · timestamp`

TARGET (Y)

meter_reading

energy consumption, in kWh

Known in train · **predicted** in test

Key Challenges & Our Goal

Key Challanges

Missing weather data

Many weather fields have gaps that must be imputed before modeling.

Scale & memory

20M+ rows strain memory; efficient types and pipelines are essential.

Anomalies & bad meters

Faulty readings and one dominant building can distort training.

Distribution shift

Forecasting 2 years from 1 demands robust generalisation.

Goal

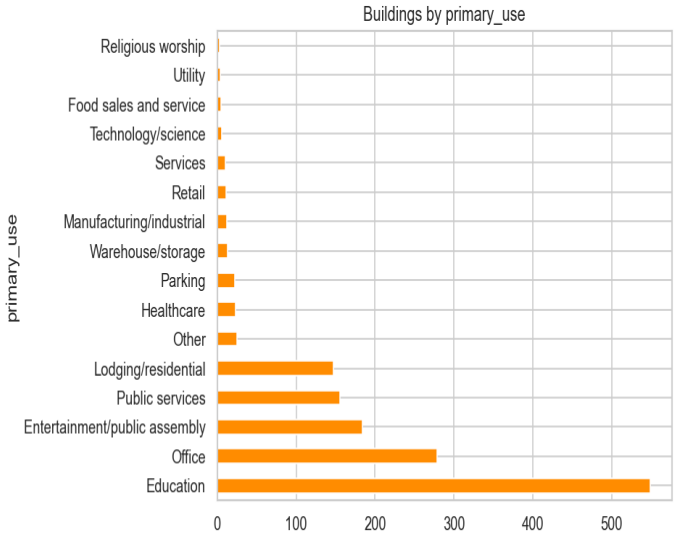
OUR GOAL

Build an accurate, robust model of building energy use across all four meter types by turning raw meter, building, and weather data into reliable baselines that make efficiency savings measurable

Exploratory Data Analysis (EDA)

Quick facts

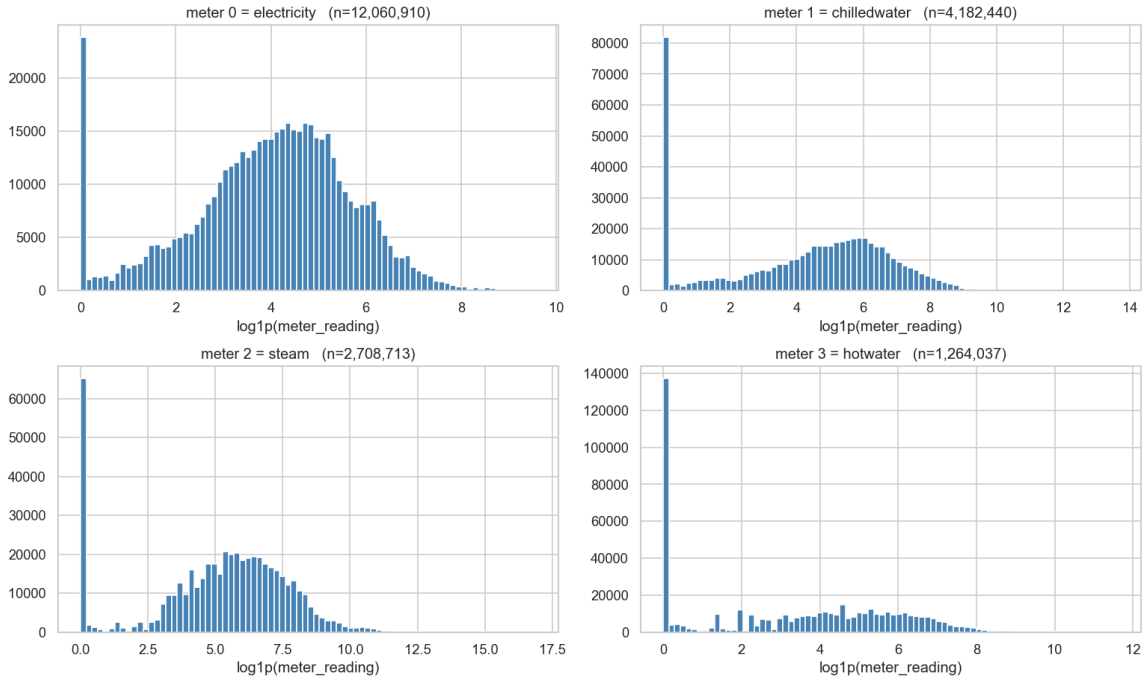
- 16 sites (mostly US universities)
- 4 meter types: electricity, chilled water, steam, hot water
- 16 primary use categories
- Time zones: 47% US/Eastern, 27% US/Central, 15% US/Mountain
- Coverage: 1,413 buildings have electricity meters; only 145 have hot water



Class imbalance

- Education: 549 (38%)
- Office: 279 (19%)
- Religious/Utility: <5
- Site sizes vary 5x
- Models will be biased toward Education + US Eastern climate

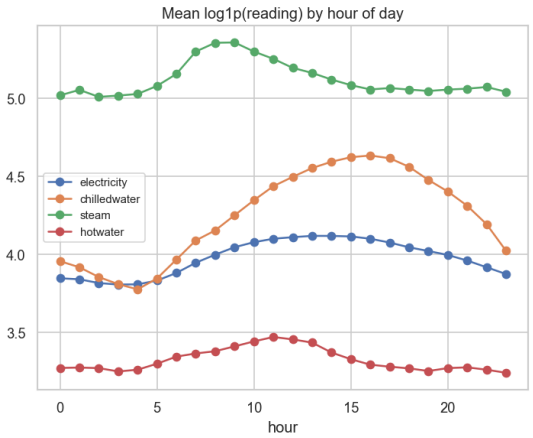
Target variable

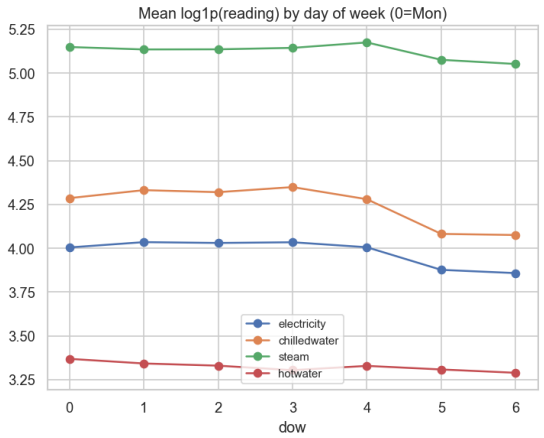


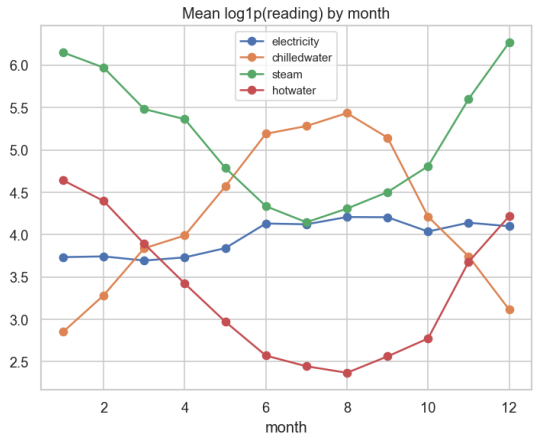
- Extremely skewed: min=0, median=79, max=21.9M
- 9.3% are exact zeros - no negatives, no NaN
- log1p(meter_reading) is mandatory (matches RMSLE)
- expm1 + clip(min=0) on the inverse

EDA – What Drives Energy Use

Each meter has a distinct time signature

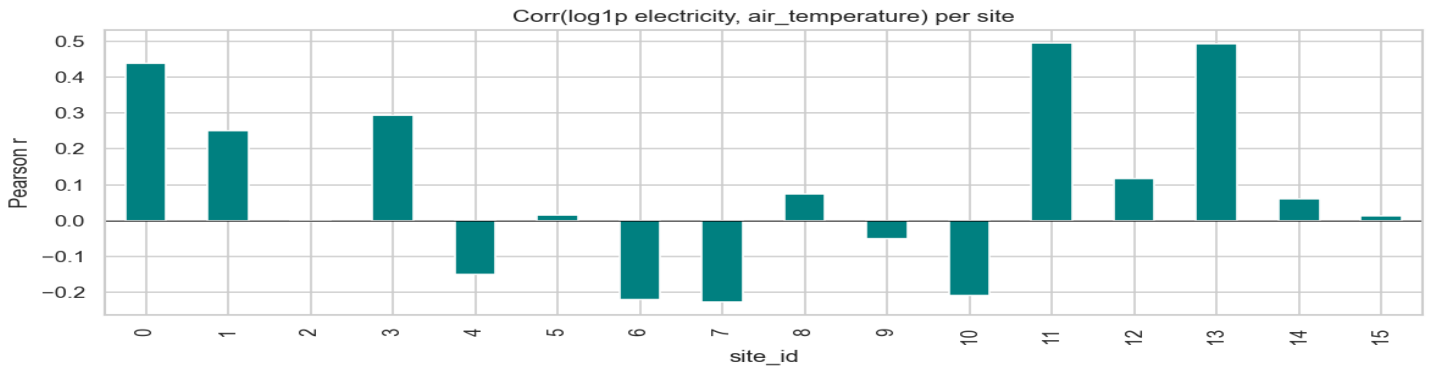






- Chilledwater: peak 3 PM, weekday-driven, summer max (cooling)
- Steam / hotwater: flat hourly, weekend-insensitive, winter max (heating)
- Electricity: mild business-hours peak, year-round flat
- Implication: train one model per meter type

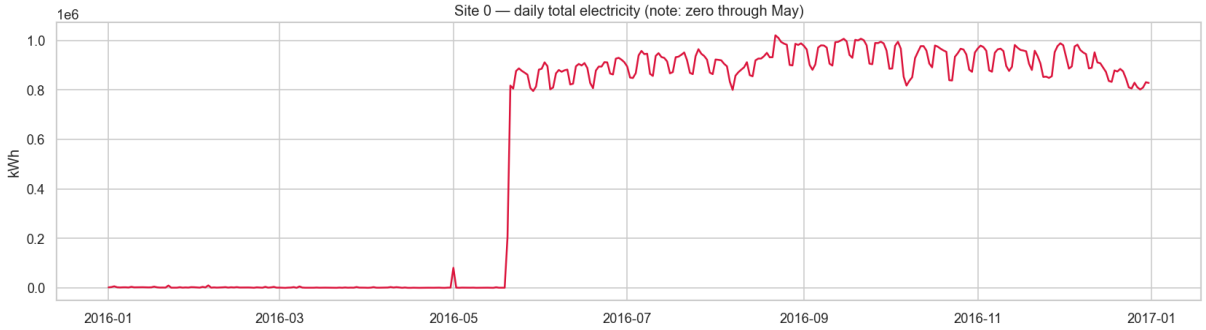
No single weather-energy relationship - corr(electricity, temp) per site



- Site 11/13: r = +0.50 (cooling-dominated)
- Site 7: r = -0.23 (electric heating-dominated)
- Most sites near zero - HVAC washes out
- site_id = implicit climate variable
- Need site x weather interactions OR per-site models

EDA – Critical Data Quality Fix

Issue 1: Site 0 has two compounding bugs



Bug 1: Missed data collection

- Jan 1 - May 20, 2016: meter=0 was 0
- Collection system not yet live
- Fix: drop rows where building <= 104 & meter==0 & ts < 2016-05-20

Bug 2: Wrong unit

- Site 0 electricity is in kBTU, not kWh
- Inflated by factor 3.412
- Fix: multiply remaining values by 0.2931
- Combined: ~0.05-0.10 RMSLE gain

Issue 2: Weather timestamps in UTC, meters in local time

- Direct merge aligns 'PM heat' with 'overnight cool' for US sites
- Off by 4-8 hours per site - kills weather features

Smart fix (adopted from reference notebook)

Auto-derive offsets from data instead of hardcoding:

- Per site, rank hourly temps within each day
- Find the hour with the highest average rank (= daily peak)
- Daily peak should fall at hour 14 (2 PM) in local time
- Offset = peak_hour - 14 -> shift weather timestamp by that

Code snippet (from reference, condensed)

```
peak_hour = df.groupby(['site_id', date])['air_temp']
        .rank().groupby('site_id', 'hour').mean()
        .idxmax()
offset_per_site = peak_hour - 14
weather.timestamp -= timedelta(hours=offset)
```

Algorithm: LightGBM

1. LightGBM-based Gradient Boosting Decision Tree regression model

- Sequentially adds multiple decision trees to **correct** the prediction errors of previous stages.
- Effectively learns the non-linear relationships and interactions among building, weather, and time variables.
- Reduces overfitting through L1/L2 regularization, subsampling, feature fraction, and early stopping.

Boosting update

$$F_m(x_i) = F_{m-1}(x_i) + \eta f_m(x_i), \quad f_m \in \mathcal{F}$$

$\mathcal{F}$: set of decision trees, η: *learning rate*

Regularized objective

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{m=1}^{M} \Omega(f_m)$$

$l(\cdot)$: prediction loss, $\Omega(f_m)$: tree complexity penalty



GroupKFold

2. 2-fold GroupKFold

- The training data is split using a **2-fold GroupKFold** method based on monthly groups.
- The data is partitioned to ensure that data belonging to the same group does not appear in both the training and validation sets simultaneously.
- The final prediction value is calculated by averaging the predictions of the two models trained in each fold.

Target transformation

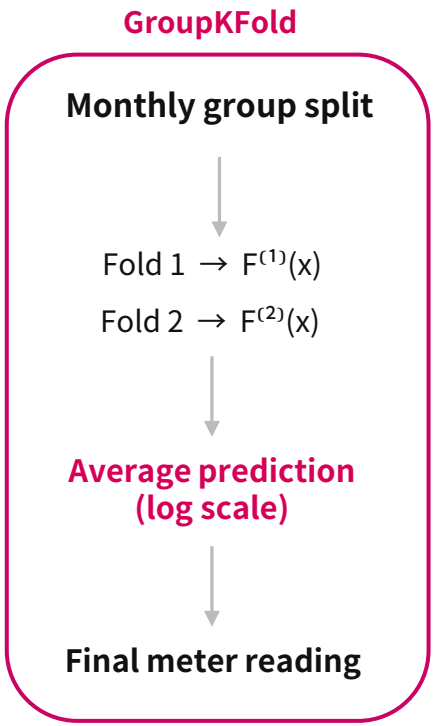
$$y_i = \log(1 + \text{meter reading}_i)$$

$$\widehat{\text{meter reading}}_i = \exp(\hat{y}_i) - 1$$

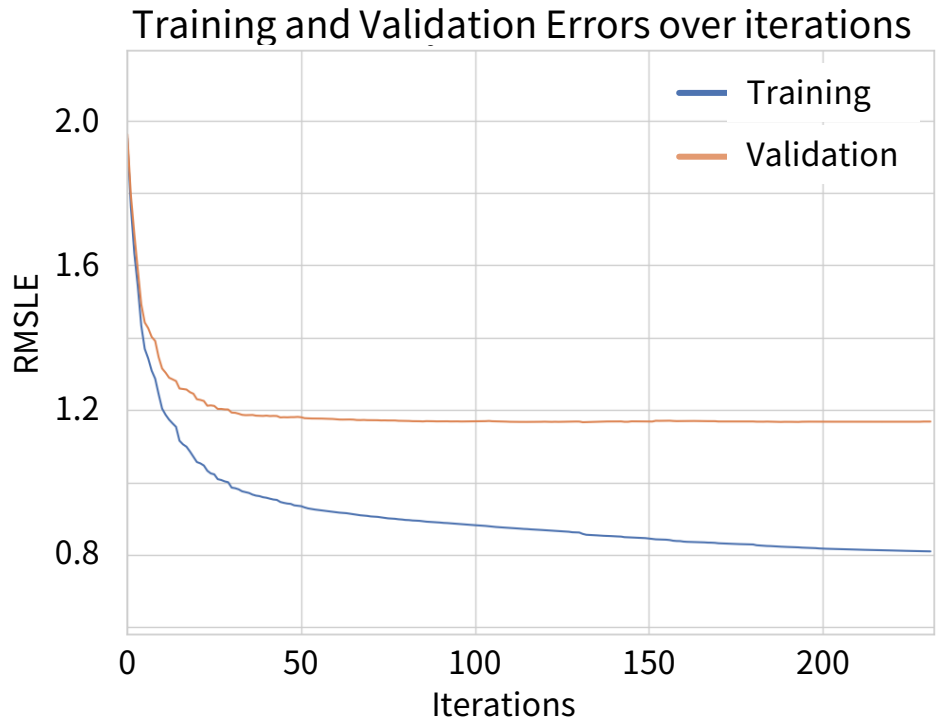
Final prediction

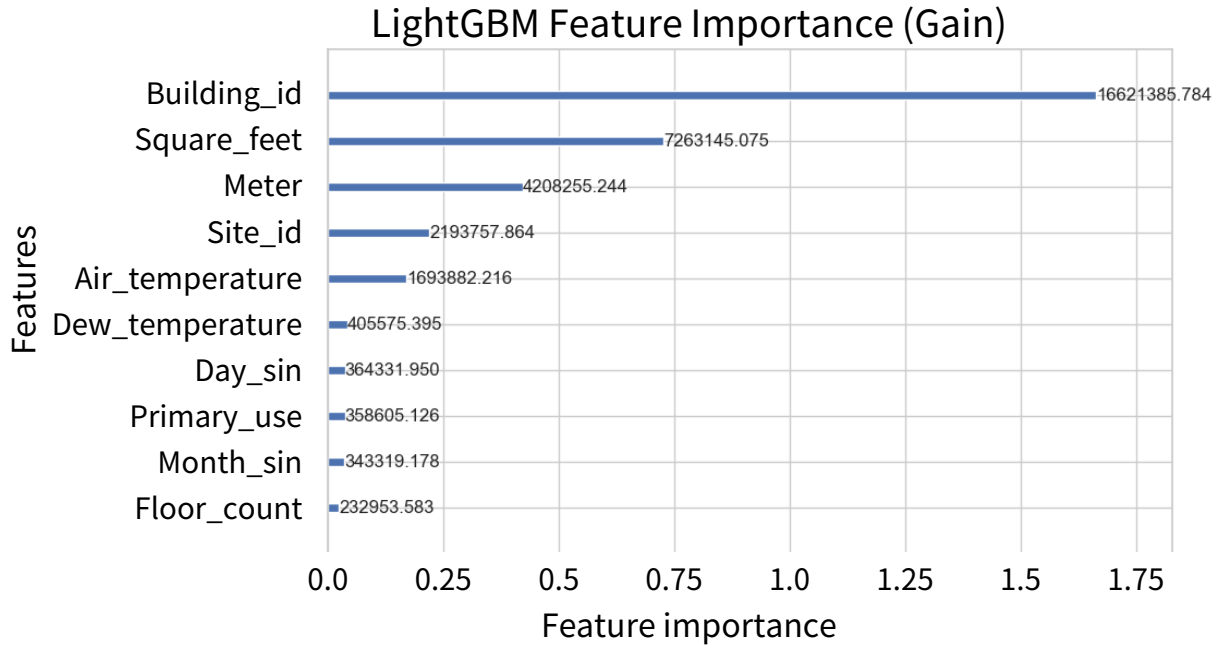
$$\hat{y}_i = \frac{1}{2}\left(F^{(1)}(x_i) + F^{(2)}(x_i)\right)$$

$$\widehat{\text{meter reading}}_i = \exp\left[\frac{1}{2}\left(F^{(1)}(x_i) + F^{(2)}(x_i)\right)\right] - 1$$



Training results and feature importance

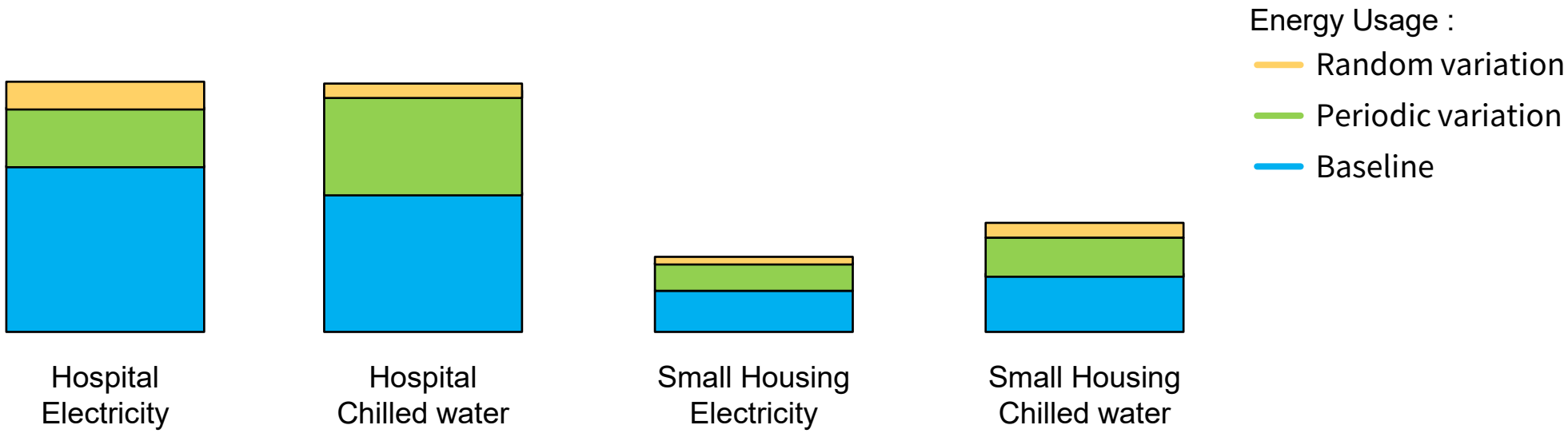




- The training was set to **stop early** when the validation score do not improve for 100 rounds of training.
- **Final iteration** to stop training was **254th iteration**, with **total training time of 3 min and 7 sec**.

- **Building-specific features** scored high importance.
- The **size of the building** represented by **square_feet** scored the 2nd highest importance.
- Month and Days **features which we added had importance** in predictions.

Why building-specific information is important?



- We are predicting the energy usage of **a known** building in future, so **classifying the ID first can be essential**.

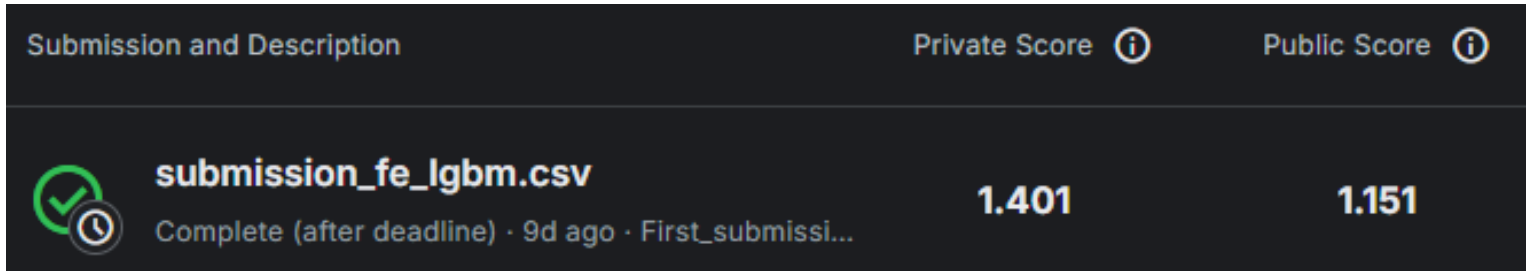
- While “Meter” represents the **type of energy**, the **pattern** of energy use **can vary** by the “Meter” **even for the same building.**

- “Square_feet” and the “external temperatures” show how the **physical environment of heat transfer within the building matters**.

- The **engineered features**, “Days and Months”, **scored decent importance**. This shows how our features improves training results.

Submission results

Kaggle submission result



- Our ranking : 2267th place (3614 participants)

- Ranking leaders : score about 1.24 RMSE for private data, utilizing ensemble of various prediction models. The prediction models include Neural network, XGBoost, Catboost, LightGBM …

- Bagging a bunch of models can help each model to learn site (building) specific patterns, leading to improved accuracy of prediction.

Conclusion

1. Data cleansing and Feature engineering

- With EDA, data distribution was examined and rows of data which seem to be not collected were fixed.
- Engineered features for timestamp, which were shown to have importance in the training.

2. Model and training

- Constructed Gradient Boosting Decision Tree regression model

- Trained with a 2-fold GroupKFold method based on monthly groups

3. Scores and what we learned

- Our model achieved a median score. Training results show that building specific information is important in energy prediction. Physical information such as temperature and size of the building also possess importance. A decision tree based GBM can provide a decent prediction when the features are adequately engineered and properly treated. Incorporating multiple prediction models will improve the precision.