ASHRAE - Great Energy Predictor III

2nd place solution

Team

Team Name: **cHaOs**
Private Leaderboard Score: **1.232**
Private Leaderboard Place: **2nd / 3614**

Team Member 1: **Oleg Knaub**
Location: Amberg, Germany
Email: olegcyganenko@gmail.com
Education: Bachelor in Information Systems and Management
Current: Expert Data Scientist at Conrad Electronics
Profiles: Kaggle, LinkedIn, Xing

Team Member 2: **Rohan Rao**
Location: Bengaluru, India
Email: rohanrao88@gmail.com
Education: MSc Applied Statistics, IIT-Bombay
Current: Senior Data Scientist at H2O.ai
Profiles: Kaggle, Wikipedia, LinkedIn, Twitter

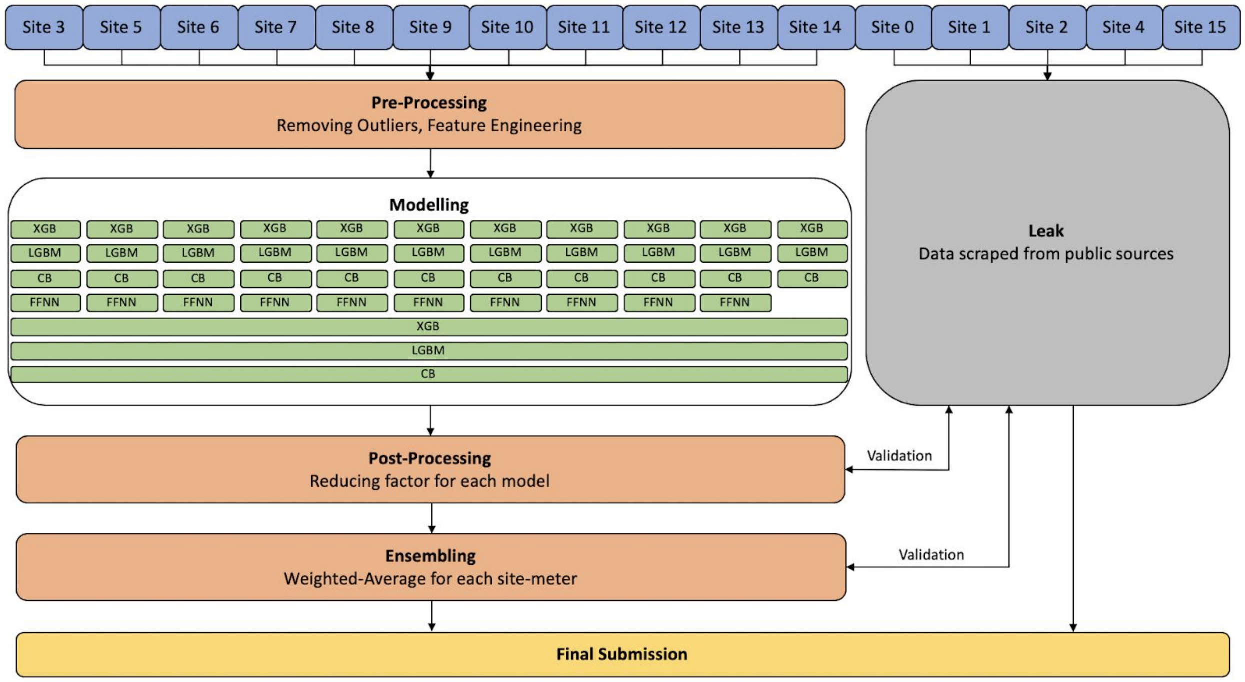
Team Member 3: **Anton Isakin**
Location: Nuremburg, Germany
Email: Anton.check@gmail.com
Education: Bachelor in Information Systems and Management
Current: ML Engineer at Siemens
Profiles: Kaggle

Team Member 4: **Yangguang Zang**
Location: Beijing, China
Email: 287071422@qq.com
Education: Statistics PhD from Chinese's Academy of Sciences
Current: Data Scientist at Netease
Profiles: Kaggle

Background

None of us have any prior industry experience in the energy domain. But all of us have done a few competitions on Kaggle across different types, so we are experienced at building machine learning solutions. Our team comprised of one Kaggle Grandmaster, one Kaggle Master and two Kaggle Experts, and each of us participated in this competition for the thrill and fun of solving real world problems using data.

Short Summary



Site 3
Site 5
Site 6
Site 7
Site 8
Site 9
Site 10
Site 11
Site 12
Site 13
Site 14
Site 0
Site 1
Site 2
Site 4
Site 15
Pre-Processing
Removing Outliers, Feature Engineering
Modelling
XGB XGB XGB XGB XGB XGB XGB XGB XGB XGB XGB
LGBM LGBM LGBM LGBM LGBM LGBM LGBM LGBM LGBM LGBM LGBM
CB CB CB CB CB CB CB CB CB CB CB
FFNN FFNN FFNN FFNN FFNN FFNN FFNN FFNN FFNN FFNN
XGB
LGBM
CB
Leak
Data scraped from public sources
Validation
Post-Processing
Reducing factor for each model
Validation
Ensembling
Weighted-Average for each site-meter
Final Submission

- Remove noise and outliers (Very important)
- Very few and basic features (For stability)
- Optimize models for each site+meter (For site-specific patterns)
- Ensemble of XGBoost, LightGBM, CatBoost, NeuralNetwork (To reduce variance)
- Post-processing (Very useful)
- Leak insertion (Sucks, but probably doesn't matter)
- Final Ensemble (approximate): 30% XGB-bagging + 50% LGBM-bagging + 15% CB-bagging + 5% FFNN

Many variations of XGB (XGBoost), LGBM (LightGBM), CB (CatBoost) were bagged: at site+meter level, at building+meter level, at building-type+meter level. Bagged XGB gave the best results among the boosting methods.

FFNN (Feed-Forward Neural Network) was used only for meter = 0.
It gave very poor results for other meters and didn't add value to ensemble.

The final ensemble scores almost the best on public LB, on leaked data as well as private LB, so hopefully it is robust and useful.

The overall solution (if optimally run in parallel can be built in 6-7hrs).

Detailed Solution

We have publicly shared our full solution on Kaggle Forum: [2nd place solution]()

Pre-Processing

A lot of the low values of the target variable seem to be noise (as discussed multiple times in the forums, specifically for site-0) and removing these rows from the training data gives a good boost in score which has been done by several other competitors too.

It was the most time consuming task as we visualized and wrote code to remove these rows for each of the 1449 buildings manually. We could have used a set of heuristics but that is not optimal due to some edge cases so we just decided to spend a few minutes on every building and remove the outliers.

Feature Engineering

Due to the size of the dataset and difficulty in setting up a robust validation framework, we did not focus much on feature engineering, fearing it might not extrapolate cleanly to the test data. Instead we chose to ensemble as many different models as possible to capture more information and help the predictions to be stable across years.

Our models barely use any lag features or complex features. We have less than 30 features in our best single model. This was one of the major decisions taken at the beginning of our work. From past experience it is tricky to build good features without a reliable validation framework.

Modelling

We bagged a bunch of boosting models XGB, LGBM, CB at various levels of data: Models for every site+meter, models for every building+meter, models for every building-type+meter and models using entire train data. It was very useful to build a separate model for each site so that the model could capture site-specific patterns and each site could be fitted with a different parameter set suitable for it. It also automatically solved for issues like timestamp alignment and feature measurement scale being different across sites so we didn't have to solve for them separately.

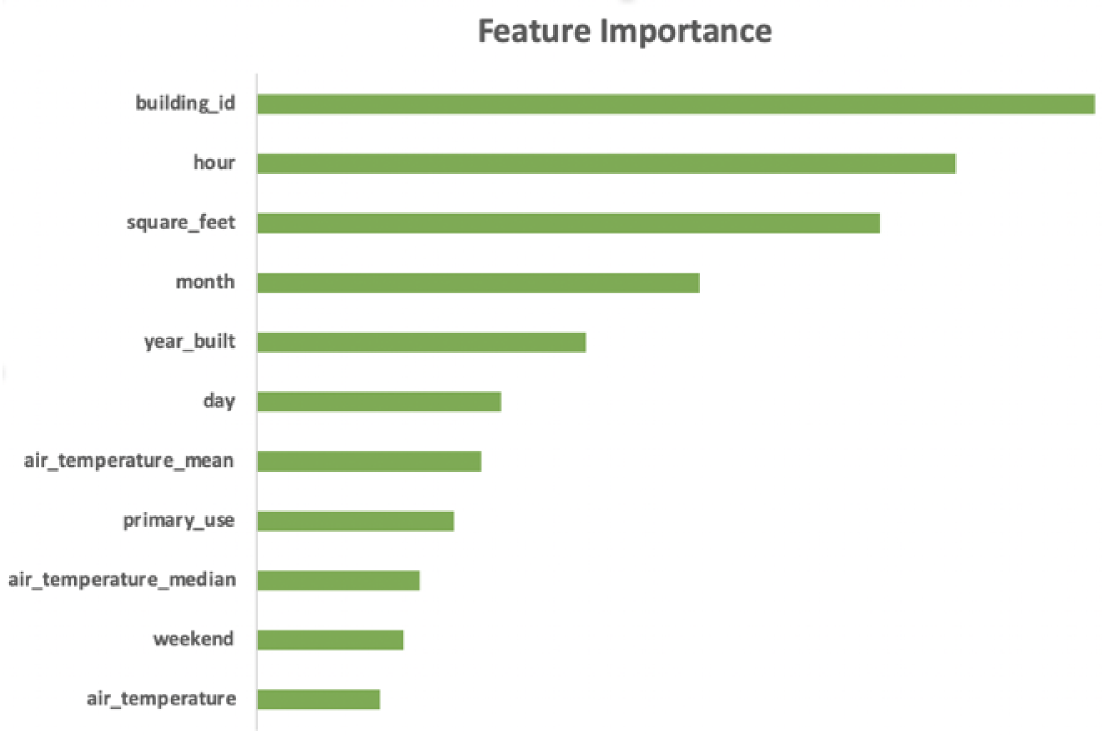
Ensembling models at different levels were useful to improve score.

Site-level FFNN was used only for meter = 0. Each site had a different NN architecture. It gave very poor results for other meters and didn't add value to ensemble.

Also, FFNN was very poor for site-14 so we didn't use it and hence that tile is missing from the models section in the architecture diagram.

For tuning of all models and hyperparameters, we used a combination of 4-fold and 5-fold CV on month from training data as well as validation on leaked data.

Feature Importance



Most of the top features are either the raw features or simple derived features.

Post-Processing

We have shared our post-processing experiments on Kaggle Forum: Why does postprocessing work? 2nd place magic

Since we remove a lot of low value observations from training data, it artificially increases the mean of the target variable and hence the model's raw predictions on test data also has an inflated mean. Since RMSE is optimal at true mean value, reducing the mean of predictions of test data by a reducing factor helps bring it down to its true mean, thus improving score.

We tried a range of post-processing values and finally ended up using 0.8 - 0.85 for most models.

Ensembling

Our best single type model was XGB but LGBM was very close and CB was not very bad either. All scored in the range of 1.04 - 1.06 on the public LB without leak.

Since FFNN was built only for meter = 0, we ensembled differently for every site+meter combination using a weighted average where the weights were determined using a combination of CV score, LB score, Leak score and intuition.

Final Ensemble (approximate) for meter = 0: 30% XGB-bagging + 50% LGBM-bagging + 15% CB-bagging + 5% FFNN

Final Ensemble (approximate) for meters 1, 2, 3: 30% XGB-bagging + 50% LGBM-bagging + 20% CB-bagging

The final ensemble scores almost the best on public LB, on leaked data as well as private LB, so hopefully it is robust and useful.

External Data

Due to the publicly available energy data of sites 0, 1, 2, 4, 15, we used them to validate our models and use them for final predictions. Kaggle have removed these sites from the final evaluation and we did not use any other external data.

Productionization

The steps to run the code of our solution is provided in the READMe. It takes over 48hrs to run the entire solution.

We strongly recommend to use the LightGBM models (in the *lgb.ipynb* notebook) if you'd like to productionize this solution. Those models are relatively simpler but almost give the same final score as our best ensemble submission on the private test set.

Hence, we too believe the models are stable and strong.