Building Energy Prediciton Project

2024-10-02

1) Introduction / Overview

1.1) Inspiration

In late 2019, the ASHRAE - Great Energy Predictor III competition was held on the Kaggle platform. The overall objective was to find the most accurate modeling solutions for the building energy use prediction. The details of the competition is here: https://www.kaggle.com/competitions/ashrae-energy-prediction [1]

The competitors was 4,370 participants in 3,614 teams from 94 countries. They submitted 39,403 predictions. The top 5 winning solutions was published; including solution summary, code, and overview video on Github repository: https://github.com/buds-lab/ashrae-great-energy-predictor-3-solution-analysis [2]

1.2) Purpose

The purpose of the capstone project does not to reproduce winner model with new dataset. The reproduction code requires computer resource, therefore it is not suitable for the edX submission and peer review. In this case, the aim of this project is to simply apply one of the most popular model, and the LightGBM was used by all of the top 5 winners.

1.3) Dataset

The dataset for this capstone project is not the dataset provided for Kaggle competition, nevertheless they are similar features and format. There are two type of data in the similar datetime in 2018 and 2019.

The first is hourly weather data. The specific weather station is selected according to the building site and location. The data can be directly download using python packages called 'meteostat'. For more information please visit: https://meteostat.net/en/ [3].

The other is electrical metered data of a specific building, it is available here: https://sgrudata.github.io/ [4]. Those two dataset that necessary for the capstone project is gathered and ready to be downloaded from https://github.com/KornkamonTantiwanit/DataScience_Capstone [5].

2) Method / Analysis

2.1) Download Data

The three data files including (1) weather data in 2018 and 2019, (2) energy metered data in 2018 (Jul to Dec), and (3) energy metered data in 2019 (Jan to Dec). To be noted that the energy metered data in 2018 will be a holdout test dataset. Therefore, it will not be touched until the final evaluation at the end.

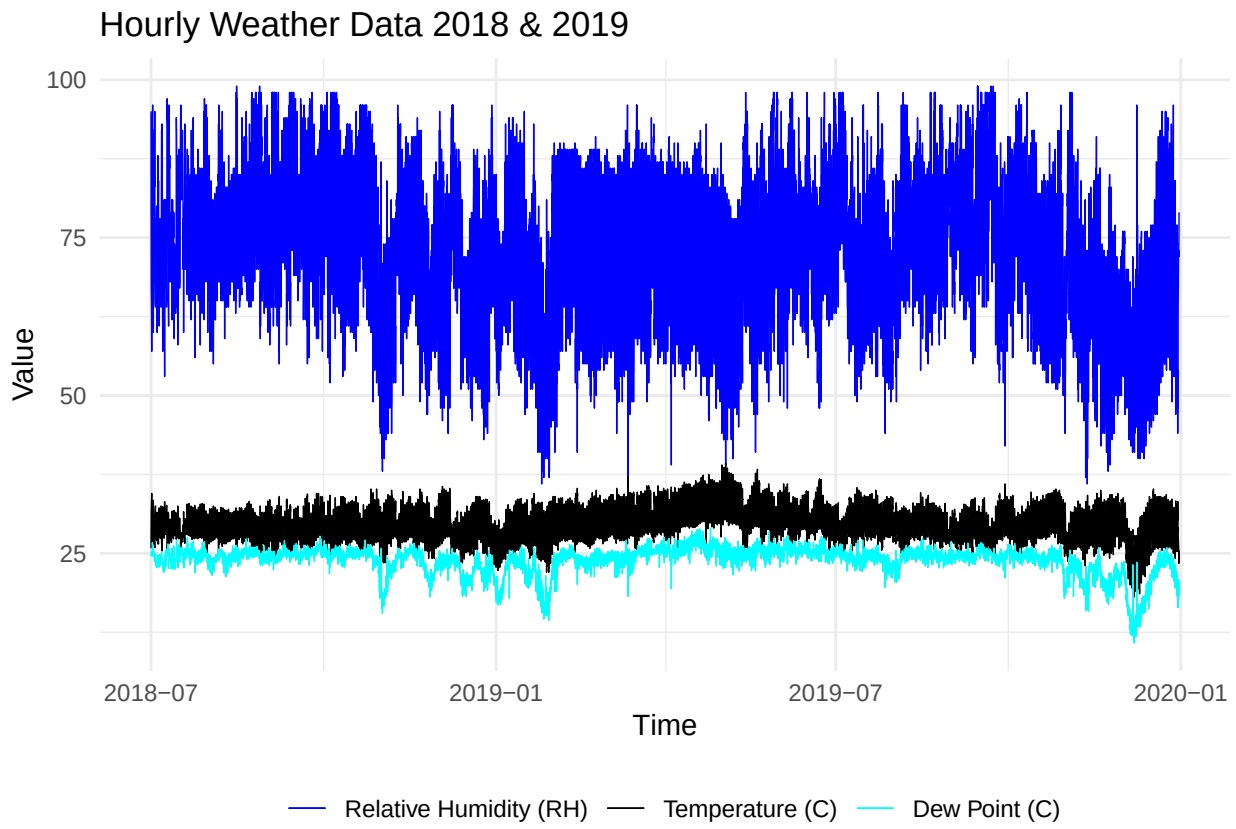
After all the files have been downloaded, the message will be shown as below.

```
## Download complete!
```

2.2) Read CSV and Visualized Data

Two data file was read and visualization, while the one file is untouched as holdout test data. The first file is weather data in 2018 and 2019. The head and plot of hourly data are shown below.

time	temp	dwpt	rhum
2018-07-01 00:00:00	25.8	24.9	95.0
2018-07-01 01:00:00	27.7	24.8	85.7
2018-07-01 02:00:00	29.6	24.7	76.3
2018-07-01 03:00:00	31.5	24.6	67.0
2018-07-01 04:00:00	32.0	24.8	66.0
2018-07-01 05:00:00	32.5	25.1	65.0



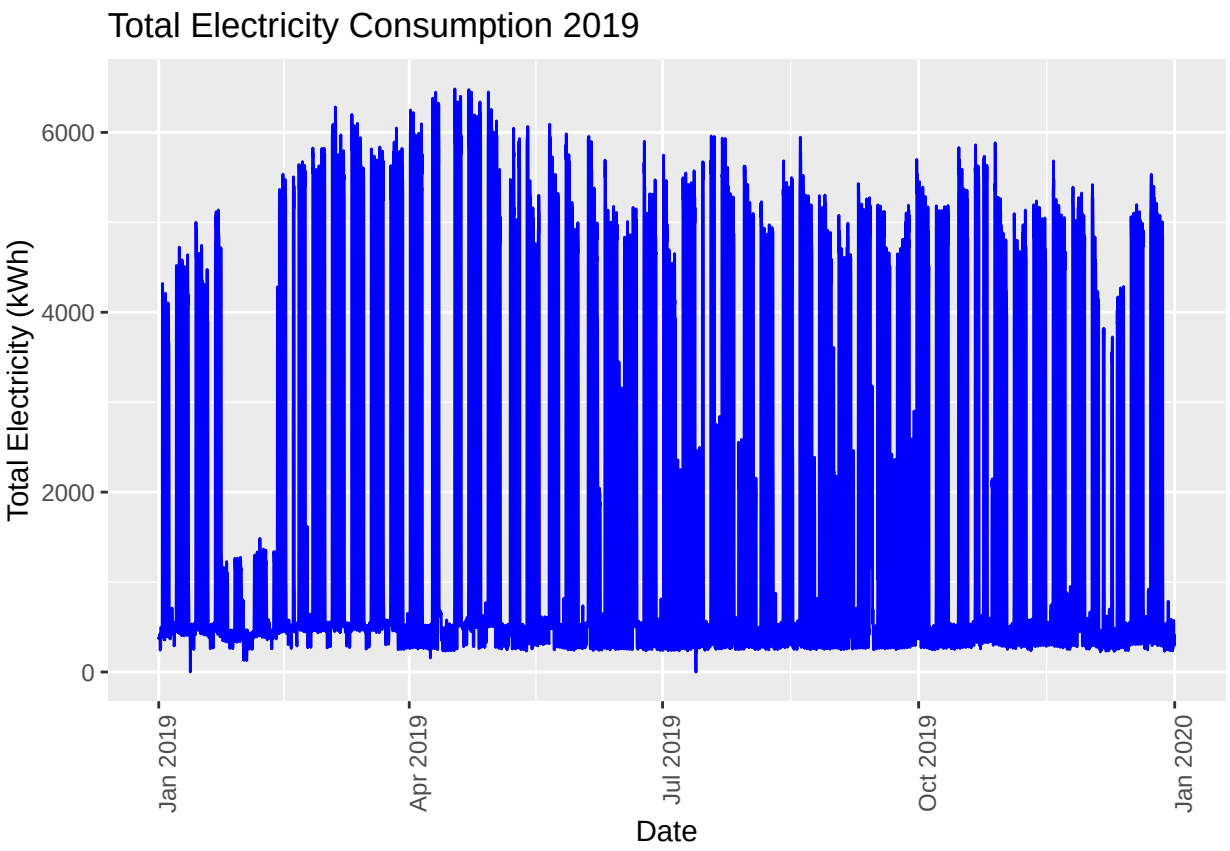
The second file is electrical metered data that was read and visualized only for training data in 2019 and leave 2018 untouch as holdout test data. The original format has multiple column as follows, moreover it is recorded in one minute data as (Wh).

```
##  [1] "Date"          "z1_AC1(kW)"    "z1_Light(kW)" "z1_Plug(kW)"  "z1_S1(degC)"
##  [6] "z1_S1(RH%)"    "z1_S1(lux)"    "z2_AC1(kW)"   "z2_AC2(kW)"   "z2_AC3(kW)"
## [11] "z2_AC4(kW)"    "z2_AC5(kW)"    "z2_AC6(kW)"   "z2_AC7(kW)"   "z2_AC8(kW)"
## [16] "z2_AC9(kW)"    "z2_AC10(kW)"   "z2_AC11(kW)"  "z2_AC12(kW)"  "z2_AC13(kW)"
## [21] "z2_AC14(kW)"   "z2_Light(kW)" "z2_Plug(kW)"  "z2_S1(degC)"  "z2_S1(RH%)"
## [26] "z2_S1(lux)"    "z3_Light(kW)" "z3_Plug(kW)"  "z3_S1(degC)"  "z3_S1(RH%)"
## [31] "z3_S1(lux)"    "z4_AC1(kW)"    "z4_Light(kW)" "z4_Plug(kW)"  "z4_S1(degC)"
## [36] "z4_S1(RH%)"    "z4_S1(lux)"
```

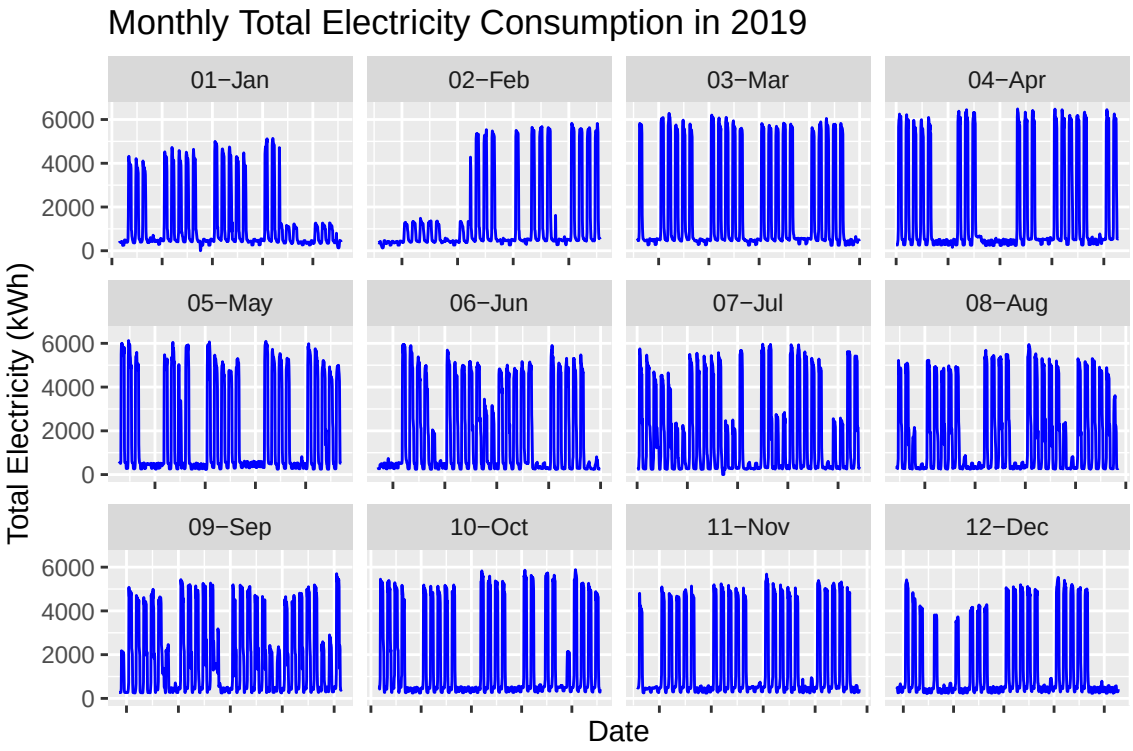
Date	z1_AC1(kW)	z1_Light(kW)	z1_Plug(kW)	z1_S1(degC)	z1_S1(RH%)	z1_S1(lux)	z2_AC1(kW)
2019-01-01 00:00:00	0	0.31	0.09	NA	NA	NA	0.00
2019-01-01 00:01:00	0	0.31	0.09	NA	NA	NA	0.00
2019-01-01 00:02:00	0	0.31	0.09	NA	NA	NA	0.00
2019-01-01 00:03:00	0	0.31	0.09	NA	NA	NA	0.85
2019-01-01 00:04:00	0	0.31	0.09	NA	NA	NA	0.94
2019-01-01 00:05:00	0	0.31	0.09	NA	NA	NA	0.93

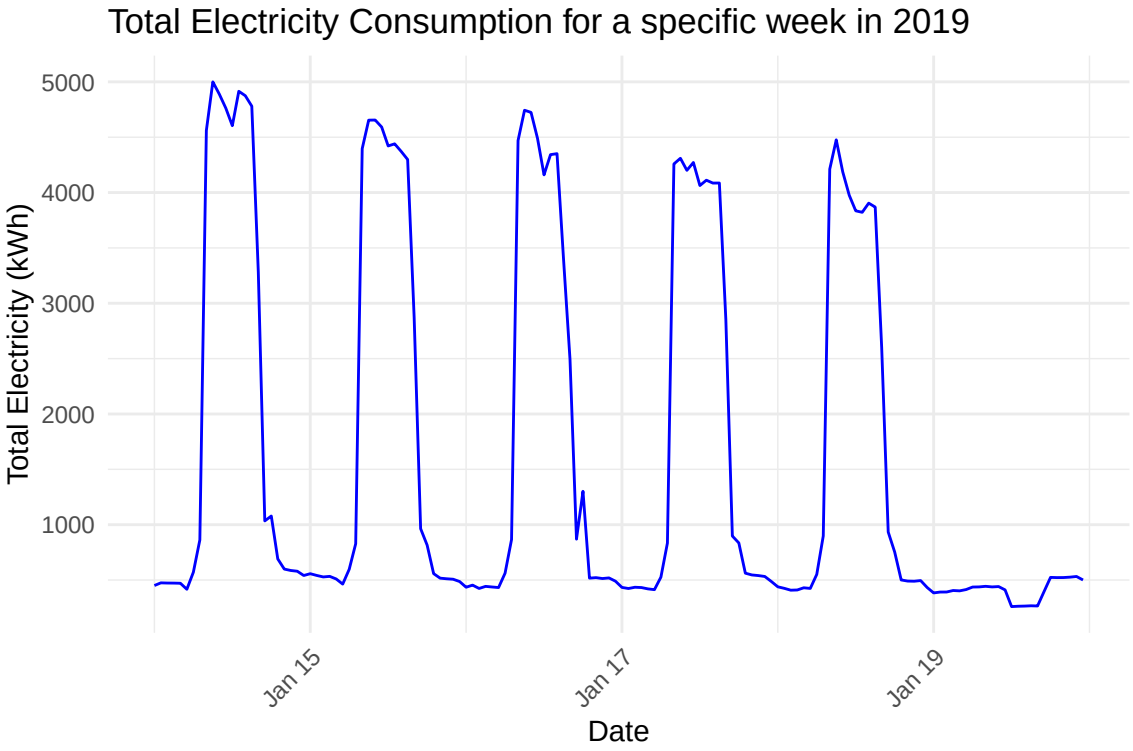
The one minute data is then selected for energy-related column in (Wh) and summed up to hourly data in (kWh). The head and visualization is below:

Date	total_Elec_kWh
2019-01-01 00:00:00	363.49
2019-01-01 01:00:00	384.56
2019-01-01 02:00:00	381.62
2019-01-01 03:00:00	373.07
2019-01-01 04:00:00	379.32
2019-01-01 05:00:00	387.69



The zoomed plot shows the pattern of monthly and a selected weekly for electrical consumption in 2019 respectively.





2.3) Combined Data for Training

The hourly weather data and hourly electrical usage in 2019 are combined as a training data, as shown below:

Date	total_Elec_kWh	temp	dwpt	rhum
2019-01-01 00:00:00	363.49	23.3	20.6	85
2019-01-01 01:00:00	384.56	23.6	18.0	71
2019-01-01 02:00:00	381.62	24.8	18.0	66
2019-01-01 03:00:00	373.07	26.4	17.2	57
2019-01-01 04:00:00	379.32	26.9	18.5	60
2019-01-01 05:00:00	387.69	27.9	18.9	58

Then split training data in to train set and test set according to the code below:

```
set.seed(123)
data_split <- initial_split(combined_data, prop = 0.8)
train_set <- training(data_split)
test_set <- testing(data_split)
```

2.4) Feature Engineering

The datetime is treated as a numeric value for training feature purposes, as following:
- Time of day: working hours is 08:30-16:30
- Weekday: Monday, Tuesday, Wednesday, Thursday, Friday
- Weekend: Saturday, Sunday
- Public holiday: 13 national holidays are 1-Jan, 6-Apr, 13-16 Apr, 1-May, 4-May, 28-July, 12-Aug, 13-Sep, 5-Dec, 10-Dec, 31-Dec

2.5) Model Training and Evaluation

This project is explore two models. The first is the most popular model among the top 5 winner of the ASHRAE - Great Energy Predictor III competition. The LightGBM, a tree-based model was applied from all winners. Moreover, the Random Forest which is one of the early developed tree-based model is also study.

Firstly is the LightGBM. The grid search uses for finding the best model's parameters. The best model is that has the lowest RMSE and the highest r-squared on the test set.

This is my setup for grid search parameters. The values of minimum RMSE and maximum R2 on test set is shown. These values will be use for model comparison.

```
# Tuning grid LightGBM
tune_grid <- expand.grid(learning_rate = c(0.01, 0.05, 0.1),
                         num_leaves = c(20, 30, 40),
                         max_depth = c(5, 10, 15))
```

```
## Best RMSE on test set (LightGBM): 1087.875
## R-squared on test set (LightGBM): 0.683
```

The second model is Random Forest (RF). It is true that RF can not handling missing values. Therefore, the missing values is checked for both train set and test set. If missing values are found, it will not be included and omitted. After the missing values in train set and test set are cleaned, the message will print out as follow:

```
## No missing values in train_fea_clean
## No missing values in test_fea_clean
```

Again, the grid search uses for finding the best model's parameters. Similar to the LightGBM, the best model has the lowest RMSE and the highest r-squared on the test set.

This is my setup for grid search parameters. The values of minimum RMSE and maximum R2 on test set is shown. These values will be use for model comparison.

```
# Tuning grid Random Forest
tune_grid <- expand.grid(mtry = c(2, 3, 4, 5),
                         ntree = c(100, 200, 500))
```

```
## Best RMSE on test set (Random Forest): 1086.279
## R-squared on test set (Random Forest): 0.684
```

2.6) Model Comparison and Selection

The RMSE and R2 from the best tune models are compared as the following:

Table 5: Comparison of RMSE and R-squared

Model	RMSE	R_squared
LightGBM	1087.875	0.683
Random Forest	1086.279	0.684

The RMSE and R2 of the two models almost similar. In this case, the LightGBM is selected due to its ability to handling missing data. This ability would be specially benefits to the real world data, which usually have missing values.

2.7) Understand the best LightGBM Model

To understand the best model, the feature importance is studied. The Gain value that measures the reduction in the loss function (error) is considered. it is found that the working hours the most importance feature that could increase or decrease the accuracy of the model.

Feature	Gain	Cover	Frequency
Is_Working_Time	0.9103588	0.2229509	0.0529010
temp	0.0326631	0.2911052	0.3105802
dwpt	0.0254833	0.2173578	0.3054608
Day_of_Week	0.0234343	0.1385346	0.1587031
rhum	0.0068405	0.1166600	0.1587031
Is_Public_Holiday	0.0012200	0.0133915	0.0136519

In addition, the best model's parameters is extracted as following. This best tune hyperparameters will be used in the process.

Table 7: Best LightGBM Parameters

learning_rate	num_leaves	max_depth
0.1	20	5

2.8) Accuracy Improvement

According to the findings above that the working hour is the most importance feature reflects the accuracy of the model. Therefore, the working hour is adjusted from 8:30-16:30 to 8:00-16:00. Then retrain the best tuned LightGBM model with the same seed to confirm the consistence of the results. The RMSE and R2 of the best tuned model and the importance feature adjustment are compared as follow:

Table 8: Comparison of RMSE and R-squared

Model	RMSE	R_squared
Best Tuned Model	1087.875	0.683
Importance Feature Adjustment	900.200	0.783

It is clearly seen that adjustment on the most importance feature helps to increase the accuracy of the predictions.

3) Results

The final evaluation is on unseen data shown below. The RMSE and R2 is calculated based on holdout test data, the electrical metered data in 2018.

```
## RMSE on holdout test set: 824.851
## R-squared on holdout test set: 0.797
```

4) Conclusion

In summary, the project studied two tree-based model, LightGBM and Random Forest. The evaluation based on test set of simple train-test split data was found similar. However, the LightGBM was selected due to its ability to handling the missing values and reflecting the real world problem. The feature importance of the best tuned LightGBM model was explore. Later, the importance feature was adjusted and the accuracy increased. Finally, the adjusted importance feature LightGBM model was used to predict the unseen data. The R2 was found 0.797 which is satisfied, and shows that the model is well generalized to the unseen data.

5) Reference

1) https://www.kaggle.com/c/ashrae-energy-prediction: The ASHRAE - Great Energy Predictor III competition held on Kaggle platform.
2) https://github.com/buds-lab/ashrae-great-energy-predictor-3-solution-analysis: The repository contains the code and documentation of top-5 winning solutions from the ASHRAE - Great Energy Predictor III competition.

3) https://meteostat.net/en/: Weather and climate database providing detailed weather data for thousands of weather stations and places worldwide.
4) https://sgrudata.github.io/: Building-level Electricity Consumption and Environmental Sensor Data.
5) https://github.com/KornkamonTantiwanit/DataScience_Capstone: Dataset for edX course, Data Science: Capstone.
6) https://chatgpt.com/: Computer Coding Tutor using GPT-4o