MGMT 6560 Project: Introduction to Machine Learning Applications

Name: Yulan Yang RIN: 661958741

Executive Summary

Project background description:

Project address: https://www.kaggle.com/c/ashrae-energy-prediction

This project is held by an American professional association named ASHRAE. They are seeking to advance heating, ventilation, air conditioning and refrigeration systems design and construction. The topic here is the expense of cooling a skyscraper in the summer. A lot of dollar has been investing in improving building efficiencies to reduce costs and emissions in an environmental impact. The task we are exploring here is whether the investment of improvements work is working.

The building owner makes payments based on the difference between their real energy consumption and what they would have used without any retrofits. A current model is used to estimate the payment without any retrofits. But it is fragmented and do not scale well.

In this competition, some accurate models of metered building energy usage in the areas of chilled water, electric, hot water, and steam meters should be developed to estimate the energy-saving investments.

Data Description:

The data comes from over 1,000 buildings over a three-year timeframe. It is hourly meter readings data.

For the **train.csv**, there are 4 attributes including **building_id** – Foreign key for the building metadata, **meter** -- {0: electricity, 1: chilledwater, 2: steam, 3: hotwater}, **timestamp** – when the measurement was taken, and the target variable **meter_reading**.

For file **building_meta.csv,** there are 6 attributes captures the information of the buildings including the primary use of the building, the square feet to describe the fross floor area of the building, the building's age and the floor count of the building.

For file **weather_[train/test].csv,** it describe the weather data from a meteorological station with some attributes such as the air temperature, cloud coverage, precipitation, wind, seal level pressure, etc.

The **test.csv** has 4 attributes: **row_id** for the submission file, **building_id**, **meter,** and **timestamp.**

Evaluation Metric

The evaluation metric for this competition is Root Mean Squared Logarithmic Error.

The RMSLE is calculated as

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

Where:

ϵ is the RMSLE value (score)
n is the total number of observations in the (public/private) data set,
pi is your prediction of target, and
ai is the actual target for i.
log(x) is the natural logarithm of x.

My findings: It is impossible to find the real accurate consumption, since there is no way to measure that. What can be done here is to find a counterfactual model to approach the reality. There should be some time-based features construction since the weather can be changed in different hours in one day or different days in one year. More than that, some other factors could be related to the impact on the building energy using such as holidays. For meter, there are four different meter types, they should be separated or aggregated, since meter types could have a heavy impact to the final prediction. K-Fold cross validation should be used, since time have significant impact.

Benchmarking of Other Solutions

Identify 3 other Kaggle *solutions* completed by others. The solution should include a score on the Kaggle prediction task. You can find by selecting on the project and then clicking on the link to Kernels. Summarize the features, modeling approach, and performance in a table. Then do some further research to comment on the approach and try to characterize what make the kernel more or less successful than others.

	Features	approach	Performance
Solution 1	Time-based features; Aggregate features for buildings at various levels; Lag-based features, created in the weather data itself and then merged with the train and test data.	This solution explores features and optimize models for each site_id. The idea is to resolve some data discrepancies that are present by dividing the data rather than cleaning. KFold Cross Validation with LGBM	Public score: 1.15

Solution 2	Extra features created includes “is_holiday”,” hour”, “weekday”.	It is a half and half approach, which splits the data in half and use each half to build a model. Build model with first half and validating on second half.	Public score: 1.10
Solution 3	Four Models: LGBM by meter type, half and half, Keras NN with embeddings for cat.features, and CV method	Model ensembling	Public score: 1.09

Apparently, Solution 3 did a better job based on the public score performance. It is not surprised that it is the best method, since it is a combination of various models, which could captures more complete information.

As shown, the prominent difference between them is the feature selection and construction. Solution 1 did a lag-based feature construction, which incurs the relationship of time manners. Solution 2 added more features such as “is_holiday”, ”hour”, and “weekday”. Which gives more specific information about time.

There is one thing should be noted, which is all of the three solutions used LightGBM as its training model. LGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed. There are some advantages of this method:

- Faster training speed and higher efficiency.
- Lower memory usage
- Better accuracy.
- Support of parallel and GPU learning.
- Capable of handling large-scale data.

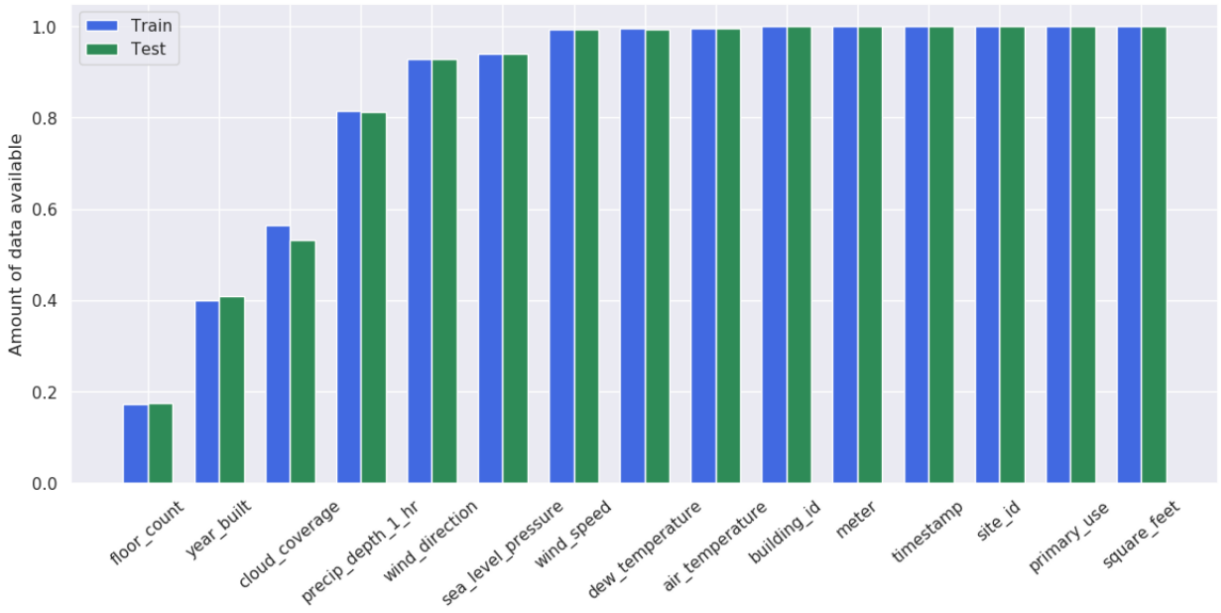
From all of them, the last one – capable of handling large-scale data is extremely important for machine learning, as machine learning always comes with large datasets.

K-Fold Validation is important as well. When data was split, how many folds should it be? Time factor should be considered when the decision was made. A more quantified way is to calculate the off-RMSE and Public LB for each K selected.

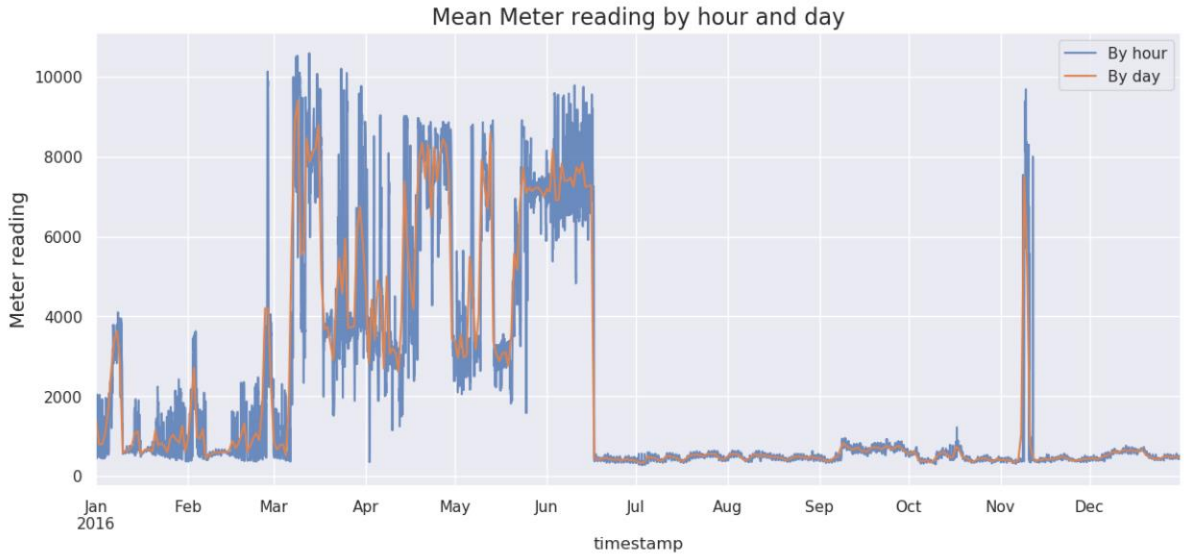
Data description and Initial Processing (3 pages)
First of all, load all the data which includes building data, training data, weather data, etc. Then merge them into one dataframe. The training dataset contains over 20M rows, while the test dataset contains over 40M rows, which makes it is hard to run on a personal computer. More

than that, some of the features are categorical features. As known, for categorical data, before putting all the data into models, some feature engineering work need to be done. However, this process could lead to a data explosion. Some simple exploratory has been done as below to have a better view of the dataset.

1. Both available train data set and test data set in one graph:



2. Take a look at the mean meter reading by hour and day:



As shown, at the beginning from Jan to Mar, the reading was at a mean about 2,000 level, then suddenly it jumped to a mean about 6,000 level with great volatility from April to June. Then it dropped immediately in the middle of June to around 500 level. However, at the beginning of November, there is a huge bump. It shows some abnormality in the data.

Modeling

For the modeling process, considering the large data site, we chose LightGBM, which is known by its distributed computation and RAM saving features. But before that, data cleaning need to be done to make the input data applicable to the model. As previous analysis, some of the features are numerical features. And it is going to use a lot of computation power and RAM space to do such as one-hot technic for these features basing on such large-size dataset. Fortunately, LightGBM has a easy function for categorical features which tells the model which ones are categorial features so that this data cleaning process can be done more easily. Moreover, the target value is the meter_reading value, which means regression rather than classification should be applied here. These are the reasons, LightGBM was selected here.

For understanding the feature of the large dataset, sampling was used here. Random shuffling technic was applied to randomly select some logs for the training dataset(1%). Noticing the timestap feature contains all the time information. For the modeling, the time should be breakdown to hour, month, weekday so that a clearer feature importance view will be seen. Generally, during the day, more energy would be used compared to the night. During different months, energy using could vary a lot as well. Same as weekday, during workdays, more energy should be used compared to weekends. The below picture shows the result.

Index	pred_y
0	5.00778
1	6.50953
2	5.84678
3	5.50973
4	4.44758
5	4.94788
6	5.46912
7	3.54134
8	4.01248
9	4.77858

Results

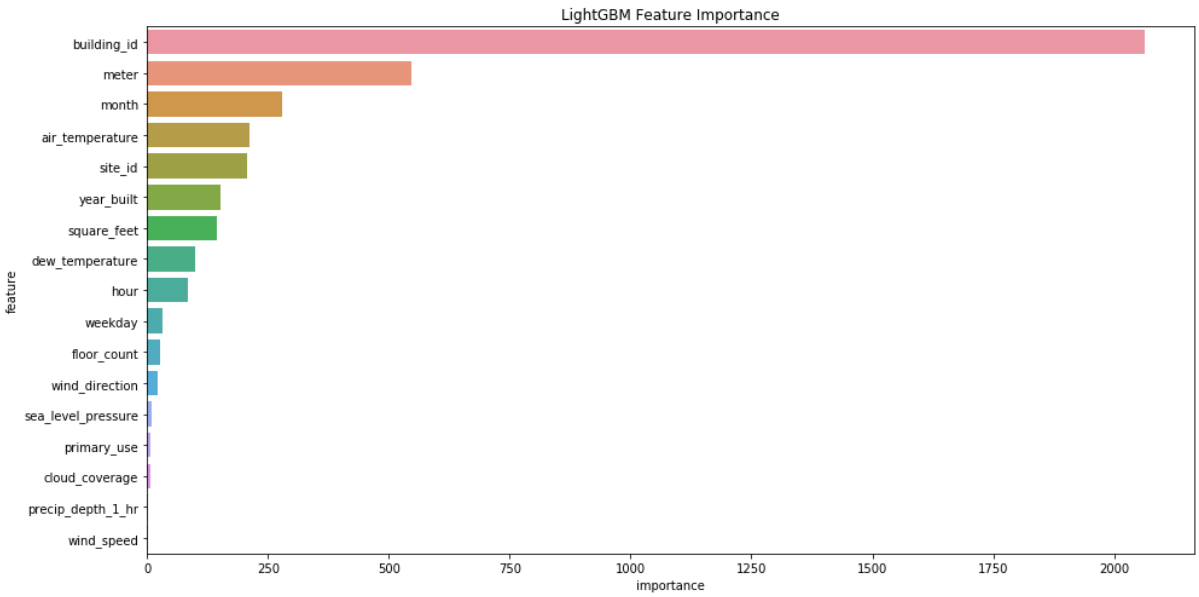
- Dataset: 1% training data from original training dataset by random shuffling
- RMSLE (Root Mean Squared Logarithmic Error.)

0.39

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

Improvement

After getting the sampling model result, we can do a feature selection based on the feature importance, so that we can reduce the dimensions. The graph below shows the rank of the feature importance. Based on this, some features such as sea_level_pressure, primary_use, cloud_coverage, precip_depth_1_hr and wind_speed can be dropped.



More than that, a model improvement can be done. Since it is a bosting model, multiple models can be compiled into one. This process may enhance the performance of the model.

Appendix

https://github.com/yangy29/MGMT6560-Final-Project.git