# Modeling and Predicting Energy Consumption

*Kevin Pham (kpham123),*
*Vikram Shanker (vshanker),*
*Vikas Yadav (vikasy)*

Stanford
CS221 P-Poster

## Problem

Identifying and reducing energy consumption is one of many necessary avenues in tackling climate change. A large source of energy consumption goes into the heating, cooling, and powering buildings. The goal of our project is to build a model that can be used to accurately **predict the pre-improvement energy consumption of buildings with energy improvements based on their use case, size, age, and local weather information**. The predicted usage can be compared with the actual usage to understand the value added by specific energy improvements, thereby creating a greater understanding of how these energy improvements impact energy usage.

## Dataset

- The data set consists of three types of data: **(1) building information, (2) local weather, and (3) energy usage**. The building data, for 1449 buildings, consists of its location, area, age, floor count, usage type. The weather and energy usage data have various time-series over one year sampled at every hour, total size ~1.5GB. The data source is a Kaggle competition scored by RMSLE [1].

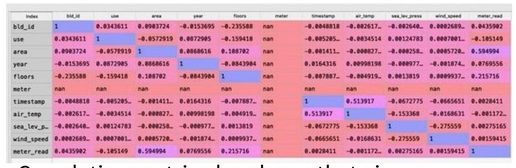
- The weather data has air and dew temperatures, wind speed and direction, cloud coverage and precipitation.

- The two data sets (building and weather) have **missing values**. We addressed this using standard methods of dropping and imputing as follows:
  - cloud coverage and precipitation columns are dropped as they are missing more than 50% of values
  - missing floor counts are replaced by 1 as that is the mode (more than 90%) and missing age was replaced by the median.

Stanford University

## Choosing Features

We performed EDA by computing **cross-correlation** of all of the features with each other. This showed that weather data has a much lower correlation with meter reading and our current models are deprioritizing and/or removing those features during training. The meter reading is highly correlated with building area and slightly correlated with floor counts.



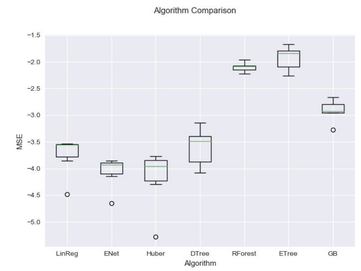
| Index | bld_id | use | area | year | floors | meter | timestamp | air_temp | sea_lev_press | wind_speed | meter_read |
|---|---|---|---|---|---|---|---|---|---|---|---|
| bld_id | 1 | 0.0343611 | 0.0903724 | -0.0153695 | -0.235588 | nan | -0.0048818 | -0.002617… | -0.002640… | 0.0002689… | 0.0435902 |
| use | 0.0343611 | 1 | -0.0572919 | 0.0072985 | -0.159418 | nan | -0.005205… | -0.0034514 | 0.00124783 | 0.0007001… | -0.185149 |
| area | 0.0903724 | -0.0570919 | 1 | 0.0868616 | 0.180702 | nan | -0.001411… | -0.000827… | -0.000258… | 0.0005720… | 0.594994 |
| year | -0.0153695 | 0.0072985 | 0.0868616 | 1 | -0.0043904 | nan | 0.0164316 | 0.00998198 | -0.000977… | -0.001874… | 0.0769556 |
| floors | -0.235588 | -0.159418 | 0.180702 | -0.0043904 | 1 | nan | -0.007887… | -0.004919… | 0.0013819 | 0.0009937… | 0.215716 |
| meter | nan | nan | nan | nan | nan | nan | nan | nan | nan | nan | nan |
| timestamp | -0.0048818 | -0.005205… | -0.001411… | 0.0164316 | -0.007887… | nan | 1 | 0.513917 | -0.0672775 | -0.0665651 | 0.0028411 |
| air_temp | -0.002617… | -0.0034514 | -0.000827… | 0.00998198 | -0.004919… | nan | 0.513917 | 1 | -0.153368 | -0.0168631 | -0.001172… |
| sea_lev_p… | -0.002640… | 0.00124783 | -0.000258… | -0.000977… | 0.0013819 | nan | -0.0672775 | -0.153368 | 1 | -0.275559 | 0.00275165 |
| wind_speed | 0.0002689… | 0.0007001… | 0.0005720… | -0.001874… | 0.0009937… | nan | -0.0665651 | -0.0168631 | -0.275559 | 1 | 0.00159415 |
| meter_read | 0.0435902 | -0.185149 | 0.594994 | 0.0769556 | 0.215716 | nan | 0.0028411 | -0.001172… | 0.00275165 | 0.00159415 | 1 |

Correlation matrix also shows that air temperature is highly correlated with dew temperature, and wind direction is correlated with wind speed, thus, the feature dimension is reduced by dropping dew temperature and wind direction.

We convert our single categorical var, " primary_use", with one-hot encoder. The timestamp field of hourly measurements across 2016 is decomposed into "month", "weekday", "hour" fields. However, the weather patterns in December are quite similar to January but are not captured. So we use **trig functions** to map time features to a unit circle and define 2 features per time feature, such as "month_sin", "month_cos". Also, we ran into general issues due to dataset size (basic manipulation and MemoryError) - after importing, we change 64-bit data types to the smallest possible size in numpy that can still represent the information.

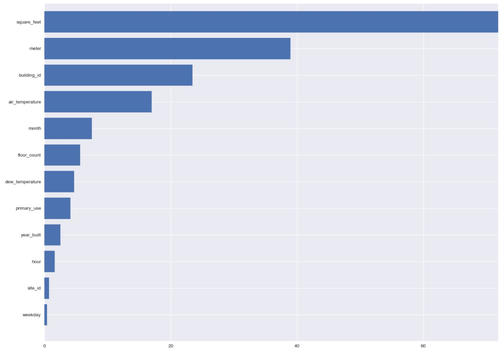
## Hybrid RF + Gradient Boosting

To explore models in SKLearn, a variety of linear and decision-based approaches were trained and fitted using default hyperparameters on a subset of the data and the different values of MSE after 5-fold cross-validation is shown in box-plots.

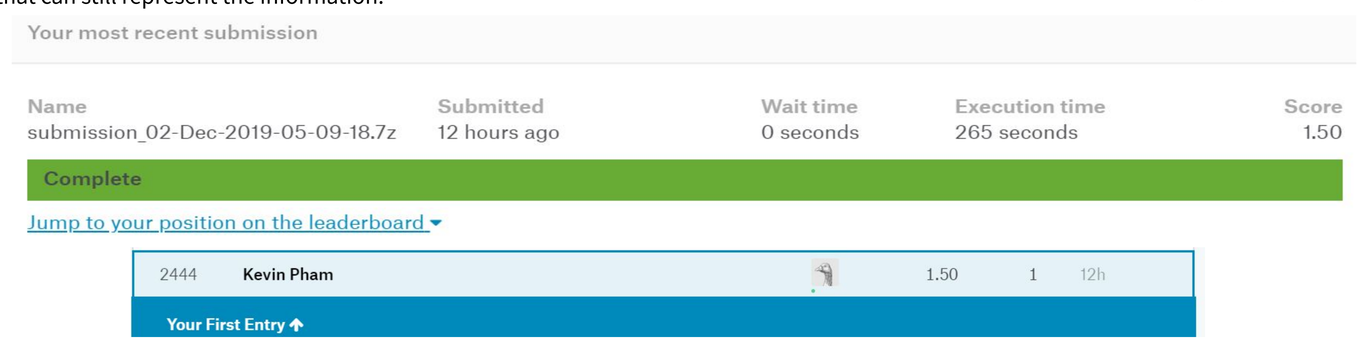



Support vector regressor was not considered due to its poor runtime performance on large number of data points. The basic linear regression, ElasticNet (combo of Ridge and Lasso by reducing model complexity and reducing number of features), and Huber (less susceptible to outliers) performs the lowest. A single decision tree is also mediocre but we see that the innovations of random forest, extra trees, and gradient boosting algorithms are promising. Random forests (and its cousin Extra Trees) are easier to train but are generally outperformed by modern boosting algos that are carefully tuned to avoid overfitting. To balance this tradeoff, the chosen approach is to train two concurrent models 1) Extra trees, 2) LightGBM and average their prediction.

## (continued)

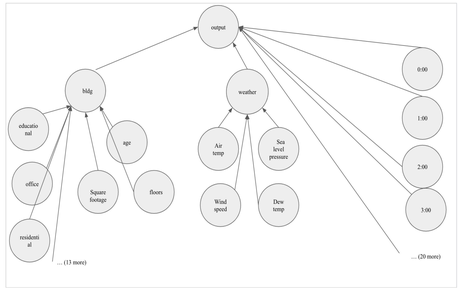



Gradient boost algos have seen recent progression and top proposed algorithms are XGBoost, LightGBM, and recently CatBoost. LightGBM is chosen due to fast training time as CatBoost seems to only outperform with very high numbers of categorical variable. Above, we graph the feature importances of the trained model, showing that meter type, building_id, air_temperature are prioritized. Of interest is square_feet and building_id are 1:1 uniquely mapped, so square_feet possibly has high importance due to its large comparative magnitude and is a candidate for normalization. By using default parameters of ExtraTrees and LightGBM, the model was able to achieve RMSLE of 1.5 and at the time of writing, 2444/2933 on leaderboard. Training on Surface Book 2 took 2.25 hrs with n_jobs=2, or using 2 cores that are hyperthreaded into 4 threads. Further work is necessary to explore parallelization and Google Cloud for training speedup.




## Neural Net

The neural network (still a work in progress) will use two topographies. The first is the following network topography. The fundamental idea behind it is that the features of the building that affect energy consumption are independent of the weather features. Finally, the time of day is also independent. Therefore, they are each grouped independently in their own hidden layer.




The second topography will be a fully connected topography. We will explore performance with different numbers of hidden nodes.

## Link to Video

shorturl.at/euxIZ

## References

[1] https://www.kaggle.com/c/ashrae-energy-prediction/data
[2] https://www.energy.gov/eere/buildings/about-building-energy-modeling
[3] https://arxiv.org/pdf/1607.06332.pdf
[4] https://www.researchgate.net/publication/316653386_Modeling_energy_consumption_in_residential_buildings_A_bottom-up_analysis_based_on_occupant_behavior_pattern_clustering_and_stochastic_simulation

# Next Steps

More Data Exploration

- Identify outliers and/or trends that could give more features

Gradient Boosting Model

- Explore sensitivity to: data scaling/normalization
- Hyperparameter search (built-in, RandomizedCV, ‘hyperopt’)
- Model training speedup (parallelization?)

Neural Net Model

- finish implementation
- explore topographies (number of hidden nodes, whether it is fully connected)