# ASHRAE - Great Energy Predictor III

Rupali Saini, Andrew Reed
*Department of Computer Science*
*University of North Carolina at Chapel Hill*
Chapel Hill, NC, USA
{rupaliis@live, reed@cs}.unc.edu

***Abstract*—We developed a model to predict the electricity, chilled water, hot water, and steam consumption for different buildings at specific points of time in the future, given the physical features of the buildings, their primary usage, readings for their meters over one year, and the corresponding weather data for the time the readings were taken. Through our process, we tried several different techniques including a "naïve" model to form a benchmark, Linear Regression, Decision Tree Regressors, Random Forest Regressors, and Gradient Boosting Regressors, most of them used through the sklearn framework. We created two separate models, one for the specific buildings provided to us, and a universal model for any new building that may have similar features to those provided. Our final model is a combination of both a universal DTR and a building+meter specific DTR.**

***Index Terms*—applied machine learning, predictive regression models, energy consumption**

## I. Introduction

With the rise of climate change, there has been significant investment in worldwide efforts to reduce emissions [8]. Here, we have built a system that measures how successful these efforts have been. We measured the energy consumption of numerous buildings and created a predictive model for future consumption based on the continuance of current policies and conditions for energy consumption. These predicted values will be compared with the actual energy consumption after new policies or initiatives have been put in place to reduce emissions. The difference between these tells us how effective new initiatives to reduce energy consumption for that building have been.

Although the predictive capabilities of our models do not rank among the top on the Kaggle leaderboard, we nonetheless believe that we offer insight into the temporal nature of energy consumption. Specifically, we believe that our results demonstrate that buildings develop their own 'energy signature' that is highly dependent on when its occupants are in the building throughout the day/week/year.

We have made our code available at [2]. The rest of our paper is organized as follows. In Section II, we describe the Kaggle competition that provided the data for our models. In Section III, we detail the various models that we tested. Section IV presents our results. In Section V, we discuss some of the challenges that we faced in our testing. Section VI provides a brief overview of related work and alternate approaches. We conclude the paper with recommendations for future improvements to our work.

## II. Background

The training and testing data for this project were taken from the "ASHRAE - Great Energy Predictor III" competition [1] hosted on the website Kaggle. We were provided with over 20,000,000 instances of training data including 1400 buildings in 15 locations and the meter readings for their electricity, chilled water, hot water, and steam meters with daily and hourly timestamps throughout 2016. We were also provided with data for the weather in those locations at those times including air temperature, cloud coverage, dew temperature, precipitation, sea level pressure, wind direction, and wind speed. Before getting started on this project, we first created a data frame to combine these two files – connecting the `site_ids` and `timestamps` in the building data file to those in the weather data file. This gave us easy access to all the provided data for each particular `meter_reading` in one single row.

For testing, our models were required to produce over 40,000,000 predictions across 2017 and 2018. For each prediction, our models were given the building ID, meter, and timestamp for the prediction, as well as the associated weather data. Our models would then predict the associated meter reading. These predicted values were then scored against the actual values using the Root Mean Squared Logarithmic Error (RMSLE) [6].

## III. Models

We tested several models in order to compare their performance. Below, we detail the parameters and features chosen for each model.

### A. Naïve Models

Prior to discussing the machine learning algorithms that we tested, we begin with our 'naïve models'. These models not only served as benchmarks for performance, they also provided us with valuable insight into the temporal nature of energy consumption.

Both of our naïve models are implemented as Python dictionaries, with the only difference being the key used to delineate meter readings.

*a) Version 1:* The **key** for version 1 consists of the concatenation of `<building_id, meter, hour_of_day>`. The **value** is then the average meter reading for the given **key**. Thus, version 1 computes the year-long average for each (building_id, meter, hour_of_day)-tuple, and uses these averages for predictions.

*b) Version 2:* The **key** for version 2 consists of the concatenation of <workday, building_id, meter, hour_of_day>. Here, 'workday' is a Boolean which is set to false for readings occurring on either weekends or U.S. federal holidays; otherwise, it is set to true. As in version 1, the **value** is the average meter reading for the given **key**. Thus, version 2 computes two year-long averages for each (building_id, meter, hour_of_day)-tuple: both a workday and non-workday average.

### B. Universal Models

From the given data, we attempted to create a "universal" model for predicting energy use. For this, we ignored the building_id feature and focused on designing a general model for any building with similar features.

*1) Using Linear Regression:* First, we attempted to use the Linear Regression model from sklearn [7]. Using the default setting for the model, we simply split the given data into the **X** array of feature vectors and set the meter_readings as the **y** values to be predicted.

We manipulated the data in the following ways:

- year_built became the age of the building by subtracting the given year from the present year.
- air_temperature and dew_temperature were regularized by taking the absolute value of the difference with the mean temperature. This balanced out the values of the negative and positive temperatures.
- All the **NaN** values in the data frame were replaced with the mean values of their respective columns. This helped ensure that they did not have a huge impact on the predictions.
- Further, we detected some outliers in the data that heavily skewed our linear regression coefficients. To address these, we simply removed the meter_reading values that were outside the upper and lower 0.99 quantile range.

*a) Version 1:* The first model we tried was a general linear regression model using weather and building data including square_feet, year_built, floor_count, air_temperature, cloud_coverage, dew_temperature, sea_level_pressure, wind_speed as the feature vector array **X** and all the meter_readings as the **y** values to be predicted.

However, this configuration did not produce a highly accurate model as we did not perform any feature selection to detect how important a feature may be to the prediction. Further, it did not account for any of the other features such as the primary use for the building, time of the reading, or the four different types of meter readings we were supposed to predict.

*b) Version 2:* We further created a Correlation Heat Map to select features most correlated with the meter_readings. From this, we determined that weather details beyond the air_temperature may not be strongly related to energy consumption.

So, in this next model, we only included the most correlated building features year_built and floor_count as the feature vector array **X** and all the meter_readings as the **y** values to be predicted.

This produced a more accurate model than version 1 since it only took into account the most important features. These two features' values were also in a small range which reduced the variance among the vectors. We, however, still did not account for categorical differences described by other features such as time, primary use, or meter type. We will look at these in the following section.

*2) Using Decision Tree Regressor:* The next and final model we used to implement the "universal" model with was a Decision Tree Regressor from sklearn.

We manipulated the data in the following ways:

- We first encoded all the primary_use values to numerical labels using the Label Encoder provided by sklearn.preprocessing.
- We added a feature for workday mapping to 1 if the timestamp indicated the reading was taken on a workday (Monday – Friday) and 0 otherwise (Saturday – Sunday).
- We added another feature for hour that indicated the hour of the day the reading was taken in the range of 0 - 23.
- We also used the same regularization techniques for air_temperature and year_built as earlier, converting them to the absolute distance from the mean air_temperature and the age of the building, respectively.
- Further, we replaced all the **NaN** values with the mean values of their respective columns to help ensure that they did not have a huge impact on the predictions.

Now, with all our values as integers or floating point numbers, we fit the Decision Tree Regressor with meter, site_id, primary_use, square_feet, year_built, floor_count, air_temperature, workday, and hour as the values in the **X** feature vectors, and meter_readings as the **y** values to be predicted.

This model provided a higher accuracy than the simple linear regression models since it took into account the categorical differences of the data, used only the selected features, and had a temporal component to it.

### C. "Building+Meter"-Specific Models

After attempting to design a "universal" model for energy prediction that is capable of making predictions for any building and meter, we then designed models that were "building+meter"-specific (i.e. train a model on each building+meter pair). This is possible for this dataset since no new building+meter pairs are presented in the test set that were not in the training set.

*1) Decision Tree Regressor Models:* The first of our "building+meter"-specific models is a Decision Tree Regressor (DTR) from sklearn. We used sklearn's default parameters for each DTR.

*a) Version 1:* The features used for version 1 were `hour_of_day`, `month_of_year`, and `day_of_week`. These features were one-hot-encoded, as they are each categorical in nature. Additionally, we dropped training data where the `meter_reading` was 0.0.

*b) Version 2:* For version 2, we added `air_temperature` as a feature. We dropped training data that either lacked weather data or a value for `air_temperature`. DTR version 1 predictions were used for test data that lacked an `air_temperature` reading.

*2) Random Forest Regressor Models:* We then tested sklearn's Random Forest Regressor (RFR). We set `n_estimators = 10` (i.e. 10 decision trees in each forest).

*a) Version 1:* The features used for version 1 were `air_temperature`, `cloud_coverage`, `precip_depth_1_hr`, `wind_speed`, `hour_of_day`, `month_of_year`, and `day_of_week`. After dropping training data that lacked a reading for `air_temperature`, we then imputed 0's for any other missing weather data.

*b) Version 2:* For version 2, we dropped `cloud_coverage`, `precip_depth_1_hr`, and `wind_speed` from the model (i.e. we used the same features as DTR version 2).

*3) Gradient Boosting Regressor Model:* For our final "building+meter"-specific model, we used sklearn's Gradient Boosting Regressor (GBR). We set `n_estimators = 100` and we used the same features that we used for both DTR version 2 and RFR version 2.

### *D. Ensemble Model*

Upon completion of our various models, we created a final set of predictions by averaging the predictions from our two best-performing models: (i) the "universal" Decision Tree Regressor and (ii) version 2 of the "building+meter"-specific Decision Tree Regressor.

## IV. Results

Table I depicts the RMSLE scores for each of our models against the actual values for the test set. Table I also shows the approximate Kaggle leaderboard position (labeled as "Kaggle Rank") for each score (as of Dec. 8, 2019).

As we can see, the Naïve Models establish a baseline at an RMSLE of 1.45 (Kaggle Rank of 82%). We interpret this to mean that any model with an RMSLE lower than 1.45 (i.e. in the bottom fifth of the leaderboard) is ill-matched for this predictive task.

Given our baseline, we see that the Linear Regression models clearly under-performed at this task. Thus, we believe that it is reasonable to assume that predicting a building's energy usage cannot be reduced to a linear combination of its features or weather data. Instead, as implied by our Naïve Models, there is a temporal and categorical component to energy usage that cannot be modelled by Linear Regression.

We then see that all of our tree-based models, both "universal" and "building+meter"-specific, range from 1.34 - 1.38. As mentioned in Section III, our two best models were both Decision Trees: (i) the "universal" Decision Tree Regressor and (ii) version 2 of the "building+meter"-specific Decision Tree Regressor. We hypothesize that our Random Forest and Gradient Boosting models performed worse than the Decision Tree models because we did not tune their hyperparameters.

Interestingly, our best score is achieved by simply averaging the predictions from our two best models. We believe that this is an indication that predicting energy usage is well-suited for ensemble (blending) models. Clearly, a building's energy usage is influenced by a variety of complex factors, and it is likely that no single model can adequately capture their interactions.

## V. Challenges

The size of the dataset presented a significant challenge throughout our attempts to explore the data and run our models. For example, when we were training the "building+meter"-specific models, we could run each model's `.fit()` function on all of the training features across all of the training instances at once, since we were only training on one (building_id, meter)-tuple at a time. However, since we were training a model per (building_id, meter)-tuple, we would have had to apply a lambda function across the test set in order to use the correct model's `.predict()` function per test row. Unfortunately, this was not feasible given our memory constraints. Instead, we were relegated to iterating through each test row via `.itertuples()`, a process that is much slower.

## VI. Related Work

The first two ASHRAE Great Energy Predictor Competitions were held in 1993 and 1995. While the nature and concept of the competition has remained the same over more than two decades, the techniques and ideas used to make predictions have certainly transformed. It was interesting to read about models created today and in the past by fellow participants. Here, we mention some we found the most clever and compelling.

*a) Bayesian Nonlinear Model:* (Winner #1, 1993) This model, developed by John MacKay [5], uses Automatic Relevance Determination using Bayesian methods. This automatically reduces the impact of junk inputs on the predictions by putting a "prior probability distribution over the regression parameters that embodies the concept of relevance" and prevents overfitting, taking into account only the most relevant parameters.

TABLE I
ROOT MEAN SQUARED LOGARITHMIC ERRORS

| **Model Class** | **Model** | **RMSLE** | **Kaggle Rank**[a] |
|---|---|---|---|
| Naïve | Python Dictionary - ver. 1 | 1.46 | 82.3% |
| Naïve | Python Dictionary - ver. 2 | 1.45 | 82.0% |
| Universal | Linear Regression - ver. 1 | 4.11 | 98.3% |
| Universal | Linear Regression - ver. 2 | 2.61 | 94.6% |
| Universal | Decision Tree Regressor | 1.35 | 75.6% |
| Building+Meter | Decision Tree Regressor - ver. 1 | 1.38 | 78.4% |
| Building+Meter | Decision Tree Regressor - ver. 2 | 1.34 | 75.0% |
| Building+Meter | Random Forest Regressor - ver. 1 | 1.35 | 75.8% |
| Building+Meter | Random Forest Regressor - ver. 2 | 1.35 | 75.8% |
| Building+Meter | Gradient Boosting Regressor - ver. 1 | 1.38 | 78.4% |
| Ensemble | Avg of Universal DTR & B+M DTR v2 | 1.29 | 71.7% |

[a]Kaggle ranks are approximate as of Dec. 8, 2019.

*b) Piecewise-Linear Regression:* (Winner #5, 1993) This Linear Regression model [3] took the hour and time of the year into account as regression parameters. Similar to our building+meter specific models, it addressed holiday seasons, summer vacations, the hour of the day, and weather variables.

*c) Other Models:* Apart from these, there have been numerous other successful models – neural networks, conjugate gradient neural network model, proprietary similarity based model, linear approximations using K-nearest neighbors, cascade correlation feedforward neural set, etc. [4] - that are distinct and similar to our model in many small and big ways.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we have demonstrated that predicting a building's energy usage cannot be reduced to Linear Regression. Instead, a successful algorithm must incorporate 'time' as part of its model, along with categorical differences attributed to primary usage, type of energy, and location. Moreover, our results point to the strength of ensemble methods, especially for tasks that must deal with multiple complex interactions.

In our future work, we plan to revisit Random Forests and Gradient Boosting. We believe that our training and predicting strategies can be improved to overcome the memory limitations of our hardware. Furthermore, we intend to research other algorithms that may also be well-suited to predicting energy usage.

## REFERENCES


[1] American Society of Heating, Refrigerating and Air-Conditioning Engineers - Great Energy Predictor III, https://www.kaggle.com/c/ashrae-energy-prediction.

[2] GitHub Repository, https://github.com/rupalisaini/energypredictor.

[3] M. Iijima, *A Piecewise Linear Regression on the ASHRAE Time Series Data*, ASHRAE May 17. 1994, Accessed on: Dec. 8, 2019. [Online].

[4] J. Kreider, *Predicting Hourly Building Energy Use: The Great Energy Predictor Shootout – Overview and Discussion of Results*, ASHRAE Transactions June 1. 1994, Accessed on: Dec. 8, 2019. [Online].

[5] D. MacKay, *Bayesion Nonlinear Modeling for the Prediction Competition*, ASHRAE July 2. 1994, Accessed on: Dec. 8, 2019. [Online].

[6] Metrics - MSE, R2, RMSLE, https://hrngok.github.io/posts/metrics/.

[7] scikit-learn: Machine Learning in Python, scikit-learn 0.22 documentation, https://scikit-learn.org/stable/index.html.

[8] S. Sengupta, *'Bleak' U.N. Report on a Planet in Peril Looms Over New Climate Talks*, The New York Times, Nov. 26, 2019, Accessed on: Dec. 7, 2019. [Online]. Available: https://www.nytimes.com/2019/11/26/climate/greenhouse-gas-emissions-carbon.html