Modeling Electricity Usage

A Case Study from Vartan-Gregorian Quad

[GitHub Repository]

Report by Ethan Drake
Data Science Institute at Brown University

Introduction

Accurately predicting electricity consumption is an important step in managing an electricity grid, particularly a grid that employs renewable energy technologies. Additionally, from the consumer side, predicting electricity usage can help better understand consumption patterns, which can be used to lower electricity consumption. This study looks at the electricity consumption patterns of a dormitory building, Vartan-Gregorian Quad (VGQ), and provides a case study for what it might look like to predict and monitor energy usage. By lowering electricity consumption, particularly during high-use times of the day, high-energy-consuming institutions like Brown can significantly reduce their carbon footprint and save money on energy bills.

For this project, I used two datasets: hourly electricity usage data since 2018 from the energy meter at VGQ and hourly weather data since 2018 from a weather station in Providence. The project aims to predict future electricity consumption based on weather patterns, the time of the year, and a categorical feature 'open to students,' which is based on Brown's academic calendar and indicates when students are occupying the dorm. Brown's Office of Sustainability, which provided me with the VGQ energy data, doesn't currently perform predictive analytics related to its campus energy consumption, so this project is a test case for the feasibility and usefulness of expanding the predictions to other buildings.

The target variable that I aim to predict is **hourly kWh** of electricity consumption. This is a non-iid, time-series dataset, and it is a regression problem. The original dataset spans from January 2018 to October 2023, but given the irregular patterns of energy consumption due to the COVID-19 pandemic, I decided to drop a year and a half worth of data points (March 2020 to September 2021) to not dilute the training set with outliers. Before feature engineering and pre-processing, the dataset has 36,807 data points and 21 features. Some features of the dataset include hourly humidity, dry bulb temperature, wet bulb temperature, visibility, and wind speed.

Exploratory Data Analysis

To analyze the features of the dataset, I compared the target variable (hourly kWh usage) to temperature (ºF) for the five years, and I looked at the effect that both the month and the feature

'open to students' has on kWh usage. Figure 1 shows electricity consumption compared to temperature from January 2018 to March 2020, and Figure 2 from September 2021 to October 2023. In these graphs, we tend to see spikes in kWh usage at the end of the summer months, around September, when the temperature is warm and AC usage must be high. Interestingly, though, not all of the summer months indicate high energy consumption. Indeed, in Figure 2, June to September 2022 shows *decreased* energy usage.

Figure 1

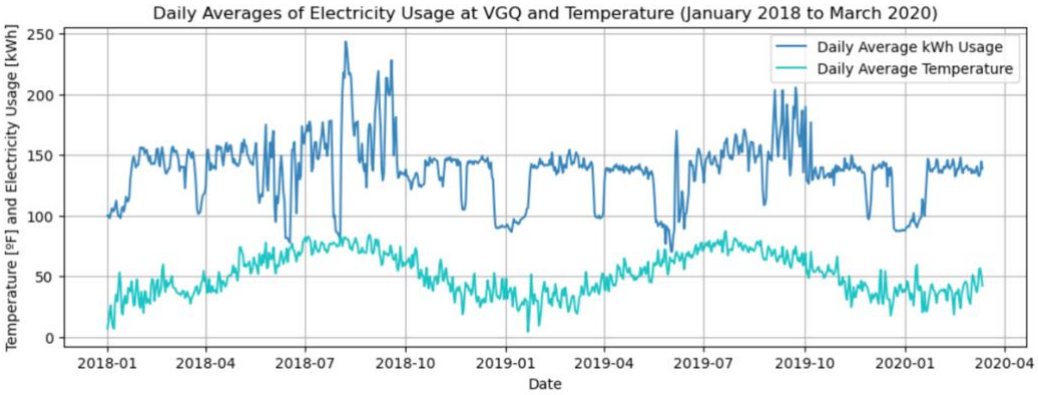


Figure 2

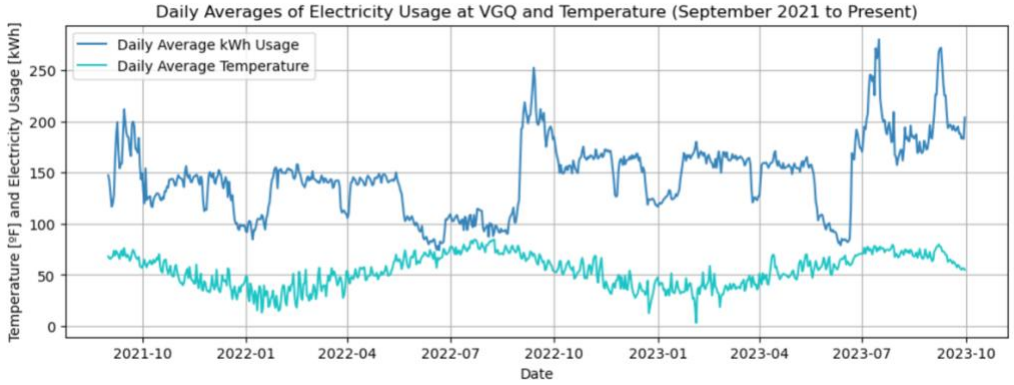
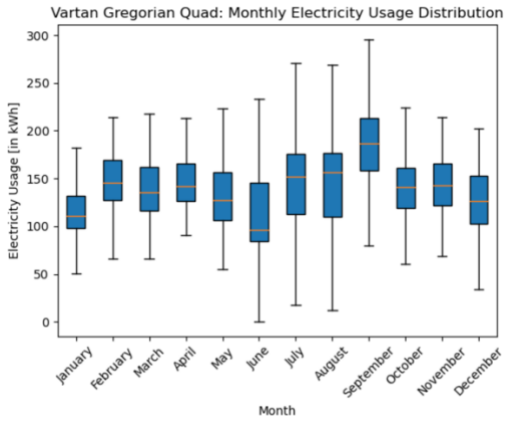
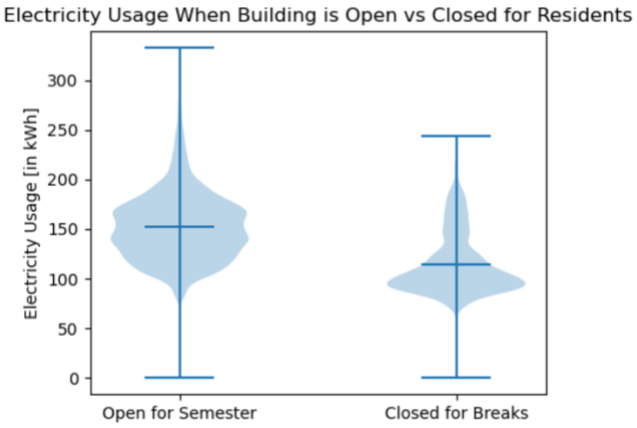


Figure 3 corroborates this finding, showing that, historically, the summer months have been both the highest energy-consuming months and the lowest energy-consuming months. Figure 4 explains this phenomenon. Given that VGQ is a student dorm, holidays from the academic calendar correlate with energy usage. In past years, such as 2023, VGQ was used to house students from Brown's summer program, which led to heightened energy use, while years like 2022 are lower because the dorm was empty.

Figures 3 and 4





Methods

Splitting: In the data preprocessing stage, I used Scikit-learn's train-test-split and time series split to split the dataset into training, validation, and test sets. Given that this is a time series dataset, I prioritized keeping the dataset in date order, so for the 90-10 train-test-split, I turned shuffling off, and for the time series split, I performed 5 splits of varying lengths. Notably, both of these splitting techniques are deterministic, meaning that the splits will produce the same training, validation, and test sets, regardless of setting a random state.

Feature Engineering: To increase the model's potential predictive power, I lagged the target variable, as well as two features, hourly temperature and humidity. For the model, I lagged the target variable by one hour, meaning that the model will predict energy usage one hour in advance. The lagged weather features include the temperature and humidity metrics for the prior 1, 2, 3, and 24 hours.

Preprocessing: I used one hot encoder and a standard scaler to preprocess the categorical and continuous features, respectively. The categorical features include the year, the month, and 'open to students,' while every other feature is a continuous feature.

ML pipeline: I used four regression models: Linear, Ridge, SVR, and KNeighbors. For each model other than Linear, I used GridSearchCV to iterate through several parameters, which can be seen in Figure 5. Notably, like the splitting strategies, each of these models is deterministic, meaning that I won't expect to see variability in model predictions. No matter how many times I run each model, it will return the same predictions, eliminating the need to run each model more than once. This means that there will be no standard deviation in the train and test scores of the models.

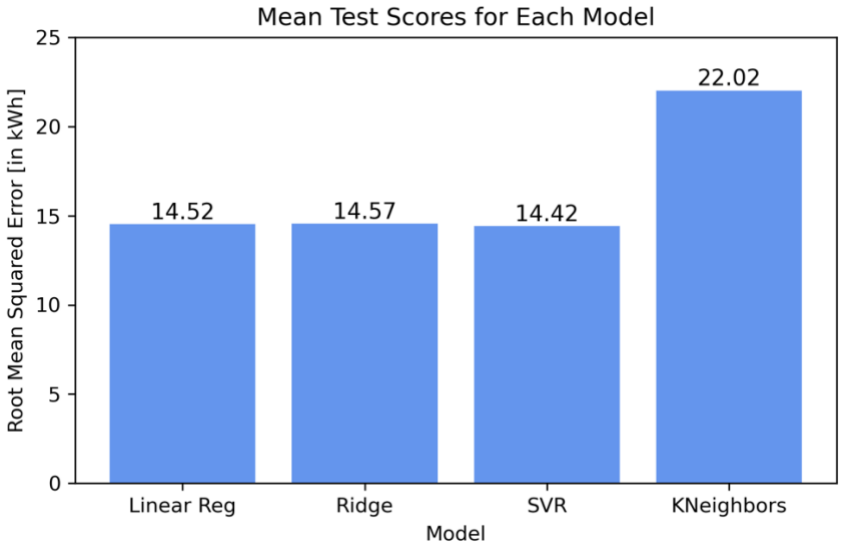
Figure 5

Model	Parameter Grid
Linear Regression	n/a
Ridge (L2 regularization)	alpha: [10^{-7}, 10^{-6}, 10^{-5}, 10^{-4} … 10^{0}, 10^{1}, 10^{2}, 10^{3}] max_iter: [100000]
Support Vector Regressor (SVR)	gamma: [10^{-4}, 10^{-3}, 10^{-2}, 10^{-1} … 10^{2}, 10^{3}, 10^{4}, 10^{5}] C: [0.1, 0, 1]
KNeighbors Regressor	n_neighbors: [1, 3, 10, 30] weights: [‘uniform’, distance’]

Results

Each of the models performs significantly better than the baseline root mean squared error (RMSE) of 51.94. As seen in Figure 6, the model with the lowest RMSE is SVR, but the linear and ridge regressions perform comparably.

Figure 6



Given how similarly the SVR and linear models performed, I decided to further analyze the linear model because this model will be much easier to deploy in a real-world context. Figures 7 and 8 show the linear model’s predictions for the test set, which stretches from April to October 2023, with two different target variable lags. Figure 7 shows the predictions for a one-hour target

variable lag, with an RMSE of 14.52, while Figure 8 shows the predictions for a 120-hour lag (5 days in the future).

Figure 7

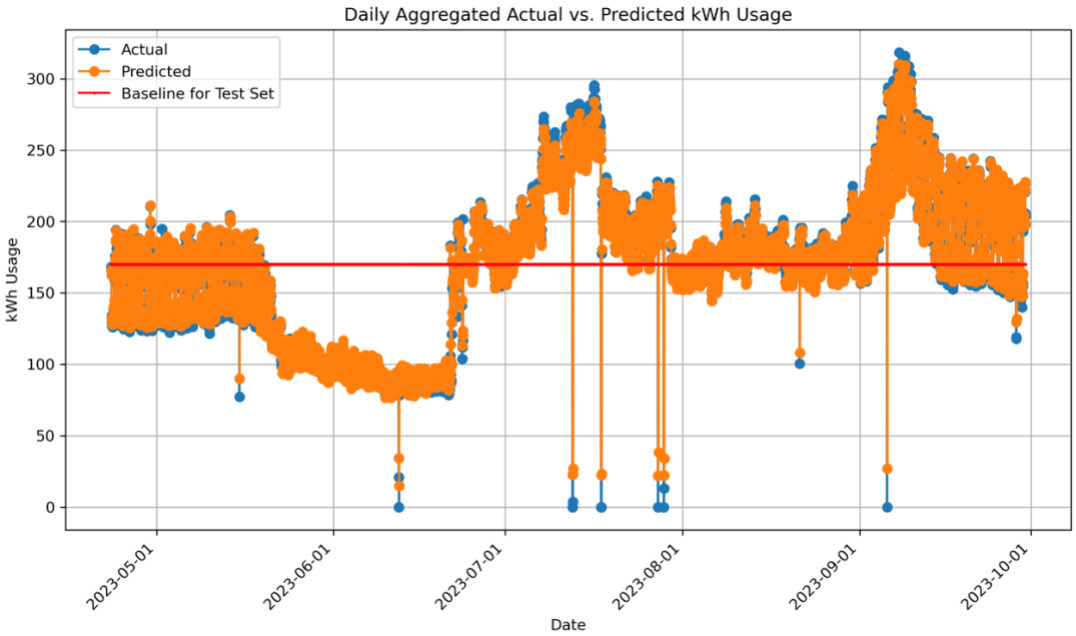
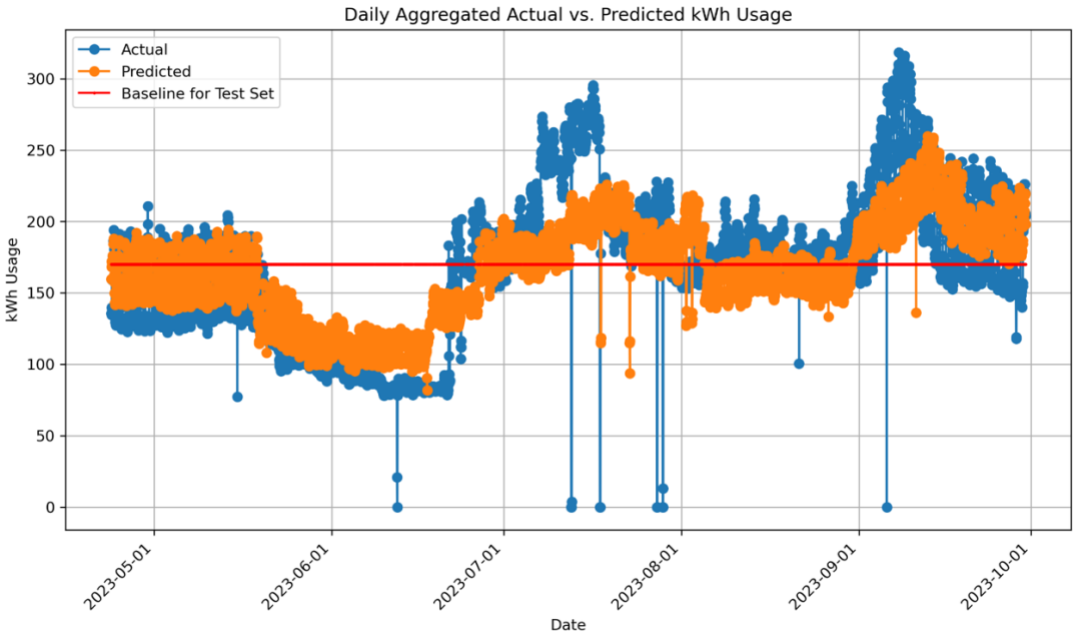


Figure 8



As for global feature importance, we see in Figure 8 that the lagged target variable ‘Net kWh lag 1’ is far and away the most influential feature. Other relevant features include the hour, humidity level, and wet bulb temperature.

Figure 8

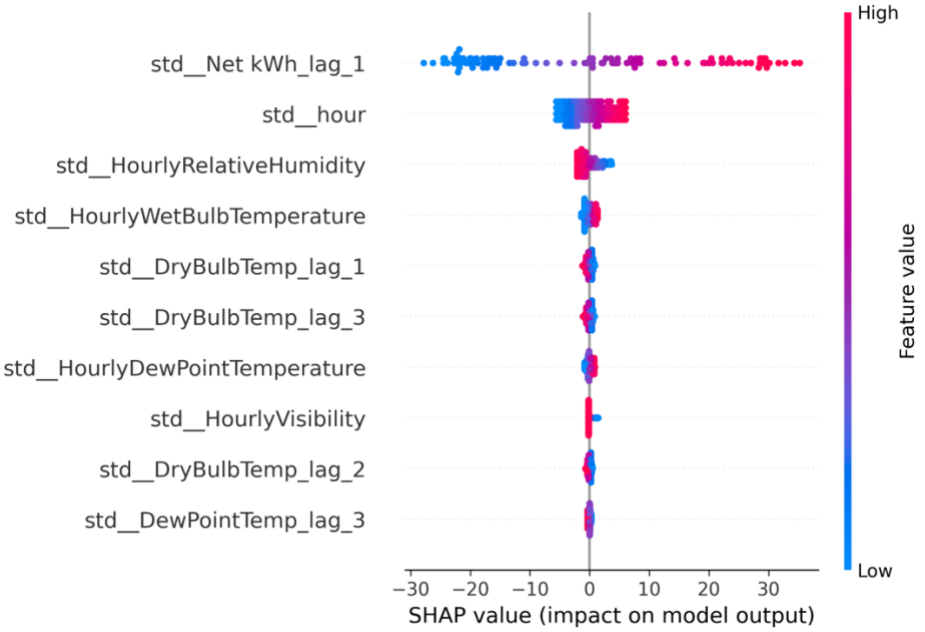


Figure 9

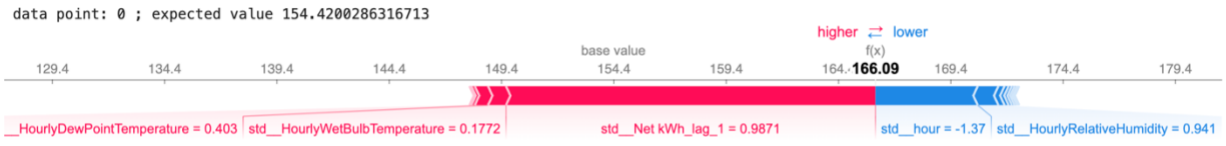
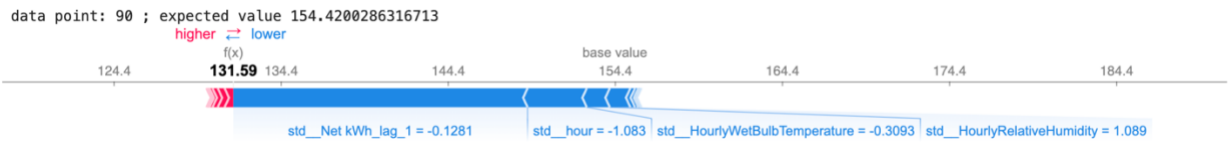


Figure 10



As for local feature importance, in Figures 9 and 10, we further find how influential the lagged target variable is in predicting the individual point.

Outlook

Going forward, I would hope to improve the model's predictive power by getting more recent data from the Office of Sustainability, which would allow me to further test the reliability of the model's predictions in a real-world context. I could also improve the model by obtaining more granular features. One thought would be accessing Brown ID swiping data, which would give us the number of individuals that swiped to enter the dorm on an hourly basis.

In considering deployment readiness, I believe that given the similarity in the RMSE of the results, it makes more sense to employ a more flexible and interpretable model, like Linear Regression. The biggest question becomes determining the best target variable lag, which influences how far in the future the model predicts energy consumption but changes the accuracy of the predictions.

References

I. Obtained electricity data from Vartan-Gregorian Quad from the Brown Office of Sustainability. https://sustainability.brown.edu/about/office-sustainability-and-resiliency

II. National Centers for Environmental Information. "Data Access." Providence, Rhode Island, TF Green Airport. January 2018 to October 2023. https://www.ncei.noaa.gov/access.